This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2830661, IEEE Access

A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends

William G. Hatcher* and Wei Yu*
*Department of Computer and Information Sciences
Towson University, Maryland 21252 USA
Emails: whatch2@students.towson.edu, wyu@towson.edu

***Abstract*—Deep learning has exploded in the public consciousness, primarily as predictive and analytical products suffuse our world, in the form of numerous human-centered smart-world systems, including targeted advertisements, natural language assistants and interpreters, prototype self-driving vehicle systems, etc. Yet to most, the underlying mechanisms that enable such human-centered smart products remain obscure. In contrast, researchers across disciplines have been incorporating deep learning into their research to solve problems that could not have been approached before. In this paper, we seek to provide a thorough investigation of deep learning in its applications and mechanisms. Specifically, as a categorical collection of state-of-the-art in deep learning research, we hope to provide a broad reference for those seeking a primer on deep learning and its various implementations, platforms, algorithms, and uses in a variety of smart-world systems. Furthermore, we hope to outline recent key advancements in the technology, and provide insight into areas, in which deep learning can improve investigation, as well as highlight new areas of research that have yet to see the application of deep learning, but could nonetheless benefit immensely. We hope this survey provides a valuable reference for new deep learning practitioners, as well as those seeking to innovate in the application of deep learning.**

***Index Terms*—Human-centered Smart Systems, Deep Learning, Platform, Neural Networks, Emergent Applications, Internet of Things, Cyber-Physical Systems, Survey, Networking and Security.**

I. Introduction

Along with Big Data and Analytics [149], [79], Cloud/Edge Computing-based Big Computing [155], [120], and the Internet of Things (IoT)/Cyber-Physical Systems (CPS) [125], [143], [148], [82], [80], [161], [127], [147], the topic of Deep Learning has come to dominate industry and research spheres for the development of a variety of smart-world systems, and for good reason. Deep learning has shown significant potential in approximating and reducing large, complex datasets into highly accurate predictive and transformational output, greatly facilitating human-centered smart systems [25], [98]. In contrast to complex hard-coded programs developed for a sole inflexible task, deep learning architectures can be applied to all types of data, be they visual, audio, numerical, text, or some combination. In addition, advanced deep learning platforms are becoming ever more sophisticated, often open source and available for widespread use. Furthermore, major companies, including Google, Microsoft, Amazon, Apple, etc., are heavily investing in deep learning technologies to supply hardware and software innovations that can further improve deep learning performance, which can be used for next generation smart-world products [4].

Though regression analysis and auto-encoding are not new topics in the field of machine learning, deep learning implementations can provide higher accuracy and better predictive performance, and are more flexible and configurable. As one of the largest areas of deep learning applications, supervised learning tasks for classification have far outstripped even human abilities in areas like handwriting and image recognition [99], [73]. In addition, unsupervised learning on datasets without any particular labels has shown the potential for the extraction of unforeseen analytical and commercial value in the form of clustering and statistical analysis. Potentially the most interesting yet, reinforcement learning provides the potential for deep learning without human supervision, through feedback from a connected environment. This type of deep learning has been heavily applied to the field of robotics and computer vision [19].

With the unceasing growth of IoT and smart-world systems driven by the advance of CPS, in which all devices are network connected and able to communicate sensed data and monitor physical objects, larger and larger datasets are becoming available for the application of deep learning, poised to materially impact our daily lives [161], [82], [143], [91], [144], [81], [36], [89]. For example, smart transportation systems will interconnect self-driving vehicles and infrastructure networks to revolutionize daily mass transit, virtually eliminating collisions and enabling secondary electrical grid storage. Smart cities shall enable the optimization of resource management via command and control in nearly all domains, from electricity, communications, and other utilities, to construction, transportation, and emergency response. Smart wearables and tele-health devices collecting diagnostic data may reveal trends that could prolong human life through disease and pattern discovery, creating a research population of unimaginable scale. Smartphones have afforded the massive creation of rich textual, audio, and visual data from various social media applications and embedded sensors, and likewise massive location and population movement data via embedded GPS modules. It is clear that all of these applications, alone or in combination, generate unprecedented Big Data. As a solution to the processing, dimensionality reduction, compression, and extraction of such Big Data, deep learning provides the most

Corresponding Author: Prof. Wei Yu (Email: wyu@towson.edu).

immediately relevant and appropriate tools, enabling the rapid analysis of complex data that spans a variety of modalities.

The primary contributions of this paper are summarized below:

- We provide an overview of deep learning technologies, commenting briefly on the history of machine learning and distinguishing deep learning from constituent shallow learning techniques. We introduce a definition of deep learning, describe the basic operations of neural networks, and further describe some basic types of neural networks applied generally.
- We categorize deep learning by learning mechanism (e.g., supervised, unsupervised, and reinforcement learning). Each learning mechanism is presented, along with the subcategories of output tasks, and common algorithmic examples.
- We provide brief descriptions of prominent deep learning platforms, commenting on their intended applications, utility, and some implementation specifics. We note useful properties of extensibility and interoperability, and highlight benchmarked platform comparisons.
- We provide a thorough and detailed investigation into the numerous areas where deep learning has been broadly applied. In particular, we demonstrate that deep learning has advanced the state-of-the-art in image and video processing, audio processing, text analysis and natural language processing, autonomous systems and robotics, medical diagnostics, computational biology, physical sciences, finance and economics, and cyber security, among others. Furthermore, we note advances in algorithmic and architectural mechanisms in deep learning research.
- Having reviewed areas of deep learning advancement, we provide insights into areas where deep learning has not been applied, or has been applied minimally. Of prime importance, deep learning acceleration and optimization will hasten the realization of in-device IoT and mobile deep learning. In addition, distributed deep learning for IoT and CPS must implement schemes to operate under an edge computing paradigm to better serve constrained devices. Applications of deep learning for network operation, management, design, and control remain relatively unexplored, yet technologies are advancing to allow inference on continuous high-throughput streams. Finally, the security of implemented deep learning networks and models, and specifically their resilience to attacks, is a critical issue given the rapid adoption of deep learning technologies coupled with prominent examples of their subversion.

In addition, compared to other survey works on the topic of deep learning [109], [67], [98], [68], [19], [38], our work takes a broad view of all fields/applications to which deep learning has been applied, and their contributions to the study and improvement of deep learning. Particularly, other works focus primarily on the advances and needs of a single learning mechanism or modality [109], [19], or towards improvements in a single application [38], [67], [68], [98]. Instead, in this paper, we primarily focus on a sweeping evaluation of deep learning applications and mechanisms to illuminate areas, in which deep learning has yet to make significant contributions.

The remainder of this paper is as follows. In Section II, we provide a brief overview of deep learning. In Section III, we categorize deep learning objectives, mechanisms, and algorithmic approaches. In Section IV, we outline various common platforms for implementing deep learning architectures. In Section V, we present a broad review of the applications of deep learning. In Section VI, we highlight areas of future deep learning application and research. Finally, in Section VII, we provide concluding remarks.

II. Overview of Deep Learning

Machine learning incorporates a vast array of algorithmic implementations, not all of which can be classified as *deep learning*. For example, singular algorithms, including statistical mechanisms like Bayesian algorithms, function approximation such as linear and logistic regression, or decision trees, while powerful, are limited in their application and ability to learn massively complex data representations. Deep learning has developed from cognitive and information theories, seeking to imitate the learning process of human neurons and create complex interconnected neuronal structures. As one of the key concepts of computing neurons and the neural model, the ability for a generic neuron to be applied to any type of data and learn indiscriminately is a powerful concept [99]. In essence, there is no singular structure for each application, but instead a generally applicable model for all applications.

Inherent to the process of machine learning are the concepts of training (iterative improvement in learning) and inference (the extraction of output of a *trained model* from some practical input). In training a model, a volume of data is split into training and testing sets, and likely a validation set as well. A machine learning algorithm is given the training data to learn some representation of, which could be in the form of a function approximation of the given feature distributions or as a set of decisions based on the contributions of each feature, among others. The validation set would also be used during training, but as a method to validate the effectiveness of the training process, the result of which is applied to tune learning parameters of the algorithm and improve the final accuracy. The test set would then provide a previously unobserved set of data to determine the final accuracy of the trained model, and is generally the source of the reported accuracy scores and other effectiveness metrics. The term inference, then, can be considered as the process of inputting a data item into a *trained* and implemented machine learning model and getting an inferred output.

With the advancement of computing technologies, the implementation of large collections of neurons was possible, giving rise to neural networks. Indeed, though neural networks are becoming commonplace, they are actually an old technology [44] that fell out of favor because of complexity and computing deficiencies. Nonetheless, this has clearly changed, thanks in no small part to the applications at which neural networks have excelled. Examples include winning the ImageNet object recognition competition [73], in which neural networks can

exceed even human accuracy, or beating humans in the game of Go without having received any direct input or game sessions against human players [123].

By definition, deep learning is the application of multi-neuron, multi-layer neural networks to perform learning tasks, including regression, classification, clustering, auto-encoding, and others. Conceptually, the most basic computational neuron, the sigmoid neuron, can be considered as a single logistic node (though there are many other algorithms that can be implemented as activation functions). Each neuron is connected to the input ahead of it, and a loss function is used to update the weights of the neuron and optimize the logistic fit to the incoming data. As part of a neural network layer, multiple parallel neurons initiated with different weights learn on the same input data simultaneously. In the application of multiple layers of multiple nodes, each node learns from all the outputs of the previous layers, stepwise reducing the approximation of the original input data to provide an output representation set. Thus, the complexity of multiple interconnected neurons is evident.

The general structure of deep neural networks is shown in Fig. 1 and Fig. 2, in which an input layer representing the raw input data (blue) feeds into multiple hidden layers of varying types (yellow), finally exiting as some output (white), which can represent regressive values, classification values, etc. Many types of neurons can be implemented, and likewise there are many types of layers depending on the necessary or desired function thereof. The most basic layer is a fully-connected layer, in which all neurons are fully connected to all input, as demonstrated in Fig. 1a. In contrast, to reduce over-fitting, some connections are removed, usually in a random manner by some percentage. This type of layer is called a dropout layer, as demonstrated in the two hidden layers of Fig. 1b.

Fig. 2 represents a convolutional neural network (CNN), as opposed to a more general recursive neural network (RNN). In RNNs, some form of optimization is used to recursively update the weights of the neurons based on the loss function results after each learning step. In CNNs, alternating convolution and pooling layers are added prior to fully-connected or dropout layers. Convolutional layers are used to filter large multidimensional matrices, such as the Red, Green, and Blue channels of an 2-dimensional image, into feature map. The pooling layer then spatially reduces the size of the feature map into a smaller and more manageable matrix. In essence, the convolution layers reduce the complexity of the image by some filter (identity, edge detection, sharpening, etc.), and the pooling layers reduce the size of each filtered result. Notice that multiple filters are typically applied to extract parallel and complementary features [131]. In addition, not all networks have this progressively reductive layer shape. Stacked autoencoders (SAEs) [38], for example, typically have an hourglass shape, first reducing dimensionality, and then expanding beck to a larger feature set. Similarly, generative adversarial networks (GANs) [45] are composed of generator and discriminator networks, where the output of the generator network is typically the same feature set as the input, and the final layers may be deconvolutional, complementing the convolutional layers of the input.

III. Categorization of Deep Learning

We now provide a categorical review of deep learning architectures by learning mechanism and learning output task, and provide brief descriptions of the many algorithmic implementations of each. The primary learning mechanisms are supervised learning, unsupervised learning, and reinforcement learning. In general, learning mechanisms are classified by the type of input data that they operate upon. Output tasks include classification, regression, dimensionality reduction, clustering, and density estimation [38].

A. Supervised Learning

Supervised learning is so named because of the requirement that the data investigated be clearly labeled, and thus the result of the output can be supervised, or classified as correct or incorrect. In particular, supervised learning is used as a predictive mechanism, in which a portion of the data is learned upon (otherwise known as the training set), another portion is used to validate the trained model (cross-validation), and the remainder is used to determine the accuracy and effectiveness in prediction. Though accuracy is an important metric, other statistical mechanisms, such as precision, recall, and F1 score, are used to assess the ability of a trained model to generalize to new data. The two primary learning tasks in supervised learning are classification and regression.

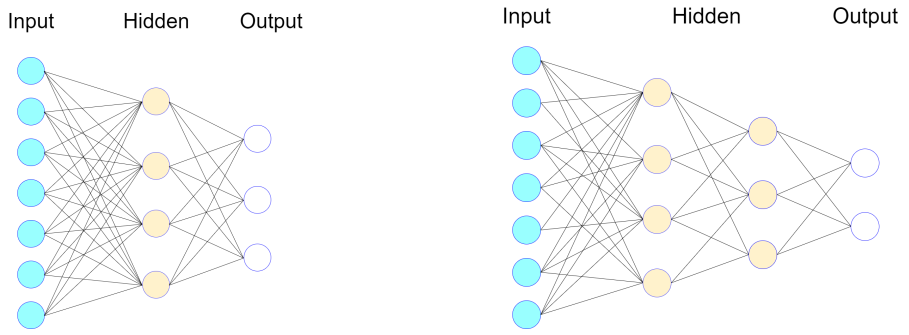
Classification: In classification, the output of the learning task will be of a finite set of classes. This can take the form of binary classification of only two classes (0 or 1), multi-class classification resulting in one class out of a set of three or more total classes (red, green, blue, etc.), as multi-label classification, where objects can belong to multiple binary classes (red or not red, and car or not car), and even as *all pairs* classification, in which every class in a finite set is directly compared to every other class in a binary way [103]. In *all pairs* classification, comparing red, green and blue, the resulting output would be test: red vs. green, red vs. blue, and green vs. blue. Examples for deep learning applications of classification include binary output in malware detection (Malicious and Benign) [48], as well as non-binary classification of handwritten numbers, as in the MNIST dataset [99].

Regression: In contrast to classification, the output of regression learning is one or more continuous-valued numbers. Regression analysis is a convenient mechanism to provide scored labels equivalent to multi-label classification, where each item of a set has a probability of belonging (*i.e.*, 0.997 red, 0.320 green, 0.008 blue). Regression has been applied in various areas, including monocular image object recognition for outdoor localization [97], among others.

B. Unsupervised Learning

In unsupervised learning, datasets provided as input for machine learning are not labeled in any way that determines a correct or incorrect result. Instead, the result may achieve some broader desired goal, be judged on the ability to find something that is easily human-discernible, or provide a complex application of a statistical function to extract an intended

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2830661, IEEE Access



(a) (b)

Fig. 1. Representative neural networks, where (a) is fully connected, and (b) includes dropout

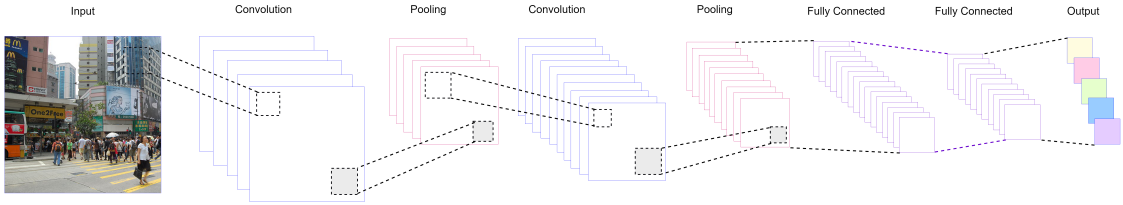


Fig. 2. Convolutional neural network for image recognition

value. For instance, clustering algorithms may cluster data into hard or fuzzy groups as desired, but, without an appropriate visual representation, it may be difficult to tell whether the clustering was indeed appropriate. Similarly, density estimation provides just that, an estimation, which may or may not be appropriate to the dataset, and auto-encoders can reduce and encode data efficiently, to be used for compression or dimensionality reduction. Nonetheless, the ability to extract the compressed representation accurately may still need to be tested to determine the appropriateness of the implementation [38].

Dimensionality Reduction: Dimensionality reduction can be carried out in various ways, including different forms of component and discriminant analysis. As an example, auto-encoders can transform input data into a reduced or encoded output for the purposes of data compression or storage space reduction. Examples of dimensionality reduction include the reduction of sequential data, such as video frames, to reduce noisy or redundant data while maintaining important features of the original data [126], or the use of deep belief networks to reduce dimensionality of hyperspectral (400-2500 nm) images of landscapes to determine plant life content [18].

Clustering: Clustering algorithms are used to statistically group data. Generally speaking, this occurs through the alternating selection of cluster centroids, and cluster membership. For example, k-means and fuzzy c-means clustering utilize the least mean square error of the distances between clusters and centroids [28], [26]. In the latter, *fuzzing* allows data membership in multiple cluster centroids, making the edges of the clusters “fuzzy". Other clustering algorithms utilize the Gaussian Mixture Model (GMM), or other statistical and probabilistic mechanisms, instead of *Euclidean Distance*, as a means to make cluster selection [111], [150]. In addition, deep neural network architectures can provide deep learning implementations for cluster analysis [38]. Examples include the use of Self-Orienting Feature Maps (SOFMS) to satisfy real-time image registration [42], and the TSK_DBN fuzzy learning network that combines the Takagi-Sugeno-Kang (TSK) fuzzy system with a Deep Belief Network (DBN) [163], among others.

Density Estimation: Density estimation, in general, is the statistical extraction or approximation of features of a data distribution, such as the extraction of densities of subgroups of data to evaluation correlations, or the approximation of the data distribution as a whole. Examples of density estimation include the estimation of power spectral density for noise reduction in binaural assisted listening devices [92], and intersection vehicle traffic density estimation utilizing CNNs on heterogeneous distributed video [152].

C. Reinforcement Learning

Reinforcement learning can be considered as an intermediate between supervised and unsupervised learning, because, though data is not explicitly labeled, a reward is supplied upon the execution of an action. More specifically, the learning architecture in reinforcement learning interacts with the environment directly, such that a change in the environment returns a specific reward. The goal of the reinforcement learning system is to maximize the reward of every state transition by learning the best actions to take at each given state. This is embodied by the perception-action-learning loop, as demonstrated in Fig. 3. This loop can occur for infinite time, or can be applied in sessions, to learn to maximize the outcome

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2830661, IEEE Access

of each session. In addition, the feedback can come directly from the environment, such as in the form of a numerical counter in a visual environment, or can be supplied as the result of some calculation or function. The two primary means of reinforcement can be divided between policy search and value function approximation, though hybrids of both have been introduced [19].

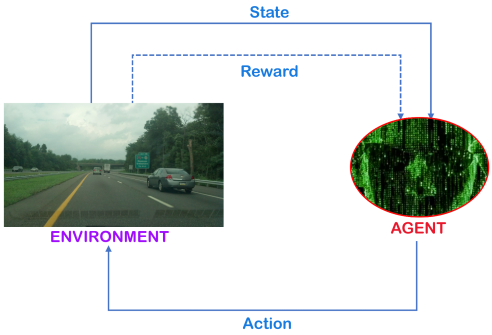


Fig. 3. Reinforcement Learning Model

Policy Search: Policy search can be carried out by gradient-based (via backpropagation) or gradient-free (evolutionary) methods, to directly search for an optimal policy. These typically output parameters for a probability distribution, either for continuous or discreet actions, resulting in a stochastic policy [19]. Though prior implementations of Google's AphaGo program, which were the first to beat a professional human player without handicap [122], were a hybrid of policy search and value function approaches, the most recent implementation, AlphaGo Zero, is entirely policy search-based, learned without any human input, and significantly outperforms the prior implementations.

Value Function: Value function methods operate by estimating the expected return of being in a given state, attempting to select an optimal policy, which chooses the action that maximizes the expected value given all actions for a given state. The policy can be improved by iterative evaluation and update of the value function estimate. The state-action value function, otherwise known as the quality function, is the source of *Q-learning* [19], [21]. An alternative to the quality function, the advantage function represents relative state-action values, as opposed to absolute state-action values [19]. As a seminal work on the application of Q-learning and Deep Q-Networks (DQN), Mnih *et al.* [93] implemented a DQN to play 49 different Atari 2600 videogames, observing four frames as environment data, extracting the game score as reward, with controller and button combinations encoded as actions. Their DQN implementation outperformed human users in the majority of games, as well as outperforming the best linear learners handily.

IV. Deep Learning Platforms

In this section, we provide an overview of popular open-source deep learning platforms. This list is not exhaustive, but is meant to provide a reference for deep learning practitioners.

A. TensorFlow

A relatively new offering in the sphere of deep learning technologies, TensorFlow was initially released by Google in late 2015, though version 1.0.0 was released in early 2017. It includes Java, C++, Go, and Python APIs, and is designed for computation on data flow graphs, in which graph nodes represent operations and edges are multidimensional data arrays (tensors). TensorFlow supports computation on multiple CPUs and GPUs, with optional CUDA and SYCL extensions [9]. In addition, TensorFlow Lite is designed for mobile and embedded machine learning, and provides an Android Neural Networks API. Recent work by Lane *et al.* [119] showed that TensorFlow performs best in server-grade multi-thread (more than 8) implementations.

B. DeepLearning4J

Deep Learning for Java (DL4J) is a robust, open-source distributed deep learning framework for the JVM created by Skymind [5], which has been contributed to the Eclipse Foundation and their Java ecosystem. DL4J is designed to be commercial-grade as well as open source, supporting Java and Scala APIs, operating in distributed environments, such as integrating with Apache Hadoop and Spark, and can import models from other deep learning frameworks (TensorFlow, Caffe, Theano) [6]. It also includes implementations of restricted Boltzmann machines, deep belief networks, deep stacked autoencoders, recursive neural networks, and more, which would need to be built from the ground up or through example code in many other platforms.

C. Theano

Theano is a highly popular deep learning platform designed primarily by academics which, unfortunately, is no longer supported after release 1.0.0 (November, 2017). Initiated in 2007, Theano is a Python library designed for performing mathematical operations on multi-dimensional arrays and to optimize code compilation [10], primarily for scientific research applications. More specifically, Theano was designed to surpass other Python libraries, like NumPy, in execution speed and stability optimizations, and computing symbolic graphs. Theano supports tensor operations, GPU computation, runs on Python 2 and 3, and supports parallelism via BLAS and SIMD support.

D. Torch

Torch is also a scientific computing framework, however its focus is primarily on GPU accelerated computation. It is implemented in C and provides its own scripting language, LuaJIT, based on Lua. In addition, Torch is mainly supported on Mac OS X and Ubuntu 12+, while Windows implementations are not officially supported [11]. Nonetheless, implementations have been developed for iOS and Android mobile platforms. Much of the Torch documentation and implementations of various algorithms are community driven and hosted on GitHub. Despite the GPU-centric implementation, a recent benchmarking study [119] demonstrated that Torch

does not surpass the competition (CNTK, MXNet, Caffe) in single- or multi-GPU computation in any meaningful way, but is still ideal for certain types of networks.

E. Microsoft Cognitive Toolkit (CNTK)

The Microsoft Cognitive Toolkit, otherwise known as CNTK, began development in Mid-2015. It can be included as a library in Python, C#, and C++ programs, or be used as a standalone with its own scripting language, BrainScript. It can also run evaluation functions of models from Java code, and utilizes ONNX, an open-source neural network model format that allows transfer between other deep learning frameworks (Caffe2, PyTorch, MXNet) [8]. Conceptually, CNTK is designed to be easy-to-use and production-ready for use on large production scale data, and is supported on Linux and Windows. In CNTK, neural networks are considered as a series of computational steps via directed graphs, and both neural network building blocks and deeper libraries are provided. CNTK has emerged as a computationally powerful tool for machine learning with performance similar to other platforms that have seen longer development and more widespread use [119].

F. Caffe and Caffe2

Caffe was designed by Berkeley AI Research (BAIR) and the Berkeley Vision and Learning Center (BVLC) at UC Berkeley to provide expressive architecture and GPU support for deep learning and primarily image classification, originating in 2014 [2], [62]. Caffe is a pure C++ and CUDA library, which can also be operated in command line, Python, and MatLab interfaces. It runs on bare CUDA devices and mobile platforms, and has additionally been extended for use in the Apache Hadoop ecosystem with Spark, among others. Caffe2, as part of Facebook Research and Facebook Open Source, builds upon the original Caffe project, implementing an additional Python API, supports Mac OS X, Windows, Linux, iOS, Android, and other build platforms [3].

G. MXNet

Apache MXNet supports Python, R, Scala, Julia, C++, and Perl APIs, as well as the new Gluon API, and supports both imperative and symbolic programming. The project began around Mid-2015, with version 1.0.0 released in December of 2017. MXNet was intended to be scalable, and was designed from a systems perspective to reduce data loading and I/O complexity [1]. It has proven to be highly efficient primarily in single- and multi-GPU implementations, while CPU implementations are typically lacking [119].

H. Keras

Though not a deep learning framework on its own, Keras provides a high-level API that integrates with TensorFlow, Theano, and CNTK. The strength of Keras is the ability to rapidly prototype a deep learning design with a user-friendly, modular, and extensible interface. Keras operates on CPUs and GPUs, supports CNNs and RNNs, is developer-friendly, and can integrate other common machine learning packages, such as scikit-learn for Python [7]. In addition, it has been widely adopted by researchers and industry groups over the last year.

V. Applications of Deep Learning

In this section, we review the primary applications of deep learning. A significant body of work toward the application of deep learning has grown steadily in the last few years. Particularly, the primary advances have been in the application of deep learning toward multimedia analysis, including image, audio, and natural language processing, which has afforded significant leaps in the state of the art for autonomous systems. Indeed, machine learning is fundamentally concerned with data fitting, the primary uses of which are optimization, discrimination, and prediction. In addition, advances in big data and cloud computing have created the potential for machine learning to flourish, enabling the requisite data collection and dissemination, as well as the computational capacity to execute deep models [26], [155], [79], [156], [120]. The existence of the data, and the nature of its potential have directly necessitated more accurate, generalized, and efficient learning mechanisms.

As shown in Fig. 4, we categorize the deep learning applications into two groups: mature applications and emerging applications. In this paper, we consider mature applications to be those with significant breadth and depth of research, such that many works exist not only across the particular field, but also within narrow subfields as well. For example, in the realm of image and video recognition, we can consider object detection, image classification, and image generation to be representative subfields which each have significant work. In contrast, we consider emerging applications as those with only breadth or depth, but not both, as well as applications that have neither. Notice that in our classification, clear subcategories exist, yet many works can be considered multi-disciplinary, belonging to multiple classes or subclasses. In the following, we review some typical examples of mature applications, reserving emerging applications for Section VI.

A. Image and Video Recognition/Classification

As generally the largest area of deep learning investigation, image and video processing, recognition, and detection have seen explosive growth in recent years. This is in no small part due to the many machine learning competitions, such as ImageNet [73]. In image and video processing, typical deep structures are convolutional neural networks, which first convolve multiple channels of images and then pool the image layers, layer by layer reducing the size of the image or frame field, before passing the results to fully-connected layers. Image and video processing has been applied to many fields of study, including autonomous systems, medical imaging, astrophysics, biometric analysis, etc.

For example, in the area of bioinformatics, Thammasorn *et al.* [129] created a three-layered extractor or *triplet* network of CNNs, fed into a comparator network, to extract features from gamma images, in which no known suitable features exist. These images, and the resulting feature, can be utilized to

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2830661, IEEE Access

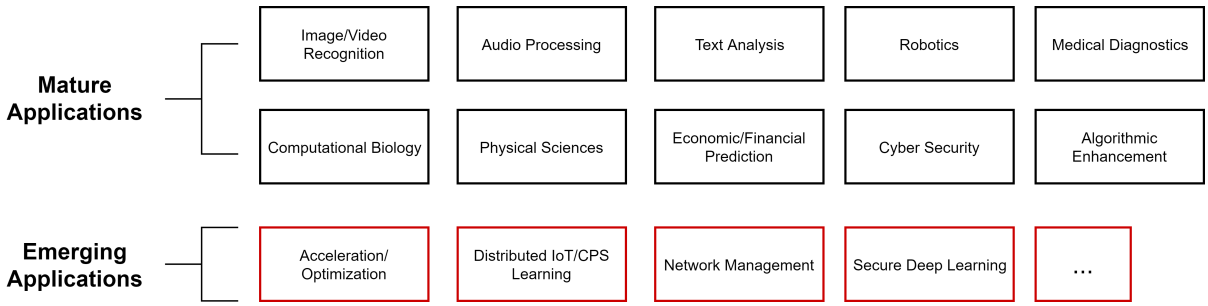


Fig. 4. Deep Learning Applications

detect potential errors in radiation therapy delivery. To carry out human action recognition, Ijjina *et al.* [59] utilized RGB-D camera data from ConvNet, NATOPS, and other datasets to develop learned temporal templates for pattern recognition via CNN.

Other examples in category of human action recognition include [124] and [54]. Particularly, Srinivasa *et al.* [124] utilized Long Short Term Memory (LSTM) for regression analysis of facial expression responses of individuals watching advertisements. Using the Affectiva-MIT Facial Expression Dataset, the authors train a model to extract expressiveness over a group of frames to better understand the advertisement's impact on the viewer. The authors additionally cluster the results to extract meaningful user states during viewing. Hong *et al.* [54] extracted 3-dimensional hand poses from 2-dimensional images, known as Hand Pose Recovery (HPR). The authors implement semi-supervised learning (combining labeled and unlabeled data), using low rank representation to map unlabeled data into labeled data space achieved via autoencoders, and utilize a ReLU activated neural network to perform classification.

As applied for general object recognition tasks, Guo *et al.* [47] developed an approach to improve 3D object recognition based on multi-view 2D images. Specifically, the authors increase intra-object variation and reduce inter-object variation through the application of a Deep Embedding Network supervised by triplet and classification loss. This framework converts the problem to a set-to-set matching problem, and the resulting DeepEm(M) implementation outperforms thirteen other methods on the MV-RED-721 dataset, and significantly improves upon precision and recall. In addition, Hickson *et al.* [52] investigated semantic classification of objects in images via weak supervision, and proposed a fully differentiable unsupervised deep clustering model. In their study, K-means clustering was used to learn parameters of the network, building features while simultaneously learning to cluster them, and storing cluster means as weights. Data was provided as objects vs. background using segmentation masks, and clustering was performed only on foreground objects. Another object recognition example is the image-based search engine developed by retraining a pretrained GoogleNet Inception-v3 CNN model using transfer learning [61]. In this study, the network was applied as a feature extractor for nearest neighbor comparison, using Euclidean distance as the similarity metric of the last-but-one fully connected layer, which was taken as the feature vector.

In considering the reconstruction of compressed data, Iliadis *et al.* [60] demonstrated the use of deep neural network architecture for compressed video sensing. Their proposed schemes for trained fully-connected networks outperform competing schemes in reconstructing compressed high-definition video. Similarly, Adler *et al.* [12] applied deep learning to block-based compressed sensing (BCS) for simultaneous learning of a linear sensing matrix and non-linear reconstruction operator. As a means to reconstruct sparse signals that have linear transforms from high-dimensional images and video, their proposed method outperforms comparable BCS mechanisms in terms of peak signal to noise ratio vs. sensing rate, as well as in execution time.

Various works have also explored the improvement of deep image learning networks through variations on architectures and constrains. For example, Higgins *et al.* [53] developed a deep unsupervised generative framework for disentangled factor learning on raw image data. The authors applied constraints inspired by neuroscience (i.e., data continuity, redundancy reduction, and statistical independence), and demonstrated that disentangled representations enable zero-shot inference and can individually encode factors of variation. Likewise, He *et al.* [50] explored significantly increasing DNN depth, which causes accuracy degradation and high training error. In response, the authors developed a deep residual learning architecture, in which layers fit a residual of the previous mapping via shortcut connections. The layers learn residual functions, which are referenced to layer inputs, are easy to optimize, and gain accuracy and reduce error. Borkar and Karam [23] investigated the impact of image distortion on pre-trained convolutional filters used in deep learning neural network and designed an approach, called DeepCorrect to improve the robustness of deep learning neural network against image distortion. In addition, Dodge and Karam [33] investigated issue of performance impact of several deep learning neural network on image classification when quality distortion is in place.

B. Audio Processing

Audio signal processing is typically concerned with noise reduction and data compression that maintains the maximum value for human listeners, highly relevant to the field of audiology. Deep auto-encoders have shown great promise in this area, and the ability to discriminate voices, languages, and background noise from multiple and singular microphone input is a significant gain that deep learning has the potential to realize. In addition, natural language processing is concerned with the detection, comprehension, and also the translation of spoken language. Widely used smart assistants have only increased in complexity, and translation applications are becoming ever more sophisticated.

As mentioned previously, Marquardt and Doclo [92] conducted noise power spectral density estimation for binaural noise reduction. They exploited the direction of arrival to improve noise reduction, as demonstrated in their simulation. Also relevant to noise reduction and enhancement, Zhang [164] proposed a cost-sensitive deep ensemble learning mechanism based upon a cost-sensitive objective function, cost-sensitive oversampling, and cost-sensitive undersampling, to improve multi-condition (i.e., multi-environment) training. Specifically, as applied to speech separation in varyingly noisy environments, the influence of low signal-to-noise ratio (SNR) on training error can be improved via varying the learning objective and sampling with SNR. In addition, Sainath *et al.* [112] applied a deep neural network framework to jointly perform multichannel enhancement and acoustic modeling for automatic speech recognition (ASR). The acoustic model was trained by applying a convolutional filter network to reduce multiple microphone signals, and then the acoustic model was learned in a convolutional LSTM network. A neural adaptive beamforming model was additionally developed to allow adaptation to changing conditions during decoding.

Toward the classification of audio signals, Sharan and Moir [118] compared SVM and DNN performance in classifying environmental noise and sound. Despite increased training time in DNNs, which more than doubled the testing time of the SVM implementation, the accuracy of the DNN was greater in all scenarios. In a relevant but distinct task, Luo *et al.* [84] applied DNN with dropout and SAE to detect and classify audio recordings as original versus captured. A means to determine whether some audio recording might have been illegally re-recorded, samples are normalized and segmented into 20 or 40 non-overlapping frames for evaluation. Both methods are able to reduce error to approximately 7.5 %, and after applying majority voting all frames in a 2-second clip, the detection rate can reach over 99 %.

Finally, regarding the synthesis of audio signals, and in particular, human speech, Gonzalez *et al.* [43] presented a technique for synthesizing audible speech from sensed articulator movement. Based on permanent magnet articulography (PMA), the authors synthesize speech from learned biosignals via Gaussian Mixture Model and RNNs, achieving 92 % intelligibility. In addition, Saito *et al.* [113] utilized GANs for statistical parametric speech synthesis to alleviate common over-smoothing effects. Through a series of subjective AB and XAB evaluations, the proposed GAN model outperforms the conventional minimum generation error DNN training method.

C. Text Analysis and Natural Language Processing

Massively adopted mobile devices have enabled continuous on-demand computing from anywhere. The social applications that comprise up the bulk of common user interactions create continuous massive data, which can be harvested and analyzed for sentiment and social understanding. Text and natural language processing affords the potential for on-the-fly language translation and the communication of humans and computer systems via natural speech.

Related to sentiment classification through text analysis, there have been a number of research efforts. For instance, Araque *et al.* [17] developed a deep learning sentiment classifier and proposed two ensemble techniques to aggregate the baseline classifier with other widely used surface classifiers. Combining both surface and deep features, the authors merge information from several common sources and conduct a performance evaluation, which confirms that the performance of the proposed models surpass that the baseline. In addition, Severyn *et al.* [116] utilized deep learning to re-rank short text pairs for optimal representation and similarity approximation without curated feature engineering. Convolutional networks were used to learn query and sentence documents separately, and then joined with additional similarity matching score into the fully connected network. Experiments demonstrated exceptional performance on the TREC: QA dataset, and comparable performance in tweet re-ranking.

Other types of classification of textual data have been accomplished, such as by Majumder *et al.* [90], who implemented deep CNNs to extract personality traits from stream of consciousness essays, combining n-gram and document-level features. Each of five major traits were trained in an individual CNN with binary output, and experiments on minor variations of their model showed increased accuracy on individual traits, with only one model outperforming on a majority of traits versus the state of the art. Similarly, Kowsari *et al.* [71] proposed a hierarchical deep learning architecture for text-based document classification, in which each deep learning model is constructed of fully connected DNN, RNN with gated recurrent units (GRUs) and LSTM, and CNN. The framework employs a parent model trained on a set of global classes, which output to child models that each learn on a distinct set of subclasses belonging to a single global class.

Deep learning has also been applied toward the generation of convincing conversational and labeling texts. For example, Li *et al.* [75] applied deep reinforcement learning for natural dialog simulation. In their study, an LSTM encoder-decoder architecture is applied to simulate two virtual agents and optimize long term reward via policy gradient. The reward combines the constraint of subsequent responses based on the prior responses (forward-looking), penalization of repetition (informativity), and mutual information between prior responses and the current response (grammatical coherency). Additionally, Zhang *et al.* [162] utilized CNNs for image detection and classification from aerial landscapes, and then

constructed natural language descriptions, utilizing the class labels via a recurrent neural network language model. Their results showed 67 % correct and 30 % partly correct descriptions.

D. Autonomous Systems and Robotics

The field of robotics has seen incredible strides in the ability to create free-standing, un-tethered autonomous walking robots, as well as in autonomous flight, driving, and navigation, among others. Deep learning has been widely applied to enable diverse sensory input to assist in the workings of autonomous machines in manufacturing and commercial spaces, and these cannot be wholly removed from video recognition an most cases.

In the area of robotic manipulation, deep learning has produced significant strides toward the rapid training of robotic arms for repetitive manufacturing tasks in a variety of ways. For instance, Gu *et al.* [46] utilized deep reinforcement learning and off-policy updates to train robotic arms on manual tasks without human intervention or demonstration. They also utilized multiple robotic arms as simultaneous workers to increase learning efficiency. In contrast, Senft *et al.* [115] implemented an interactive machine learning architecture called SPARC (Supervised Progressively Autonomous Robot Competencies), in which a human user interactively guides the robot as it learns. This is in the form of reinforcement learning, in which the human teacher has full control over the actions of the learning machine, and positive feedback is supplied to the robot for every action that it completes.

Unlike direct user manipulation training, or human-free training, Yang *et al.* [146] developed a deep learning pipeline for generalized non-backdrivable humanoid robots through teleoperation training. The model uses a deep convolutional autoencoder for image feature extraction and encoding and applies a time-delay neural network for temporal sequence evaluation with image and motor angle data to generate a continuous operational task. Furthermore, Polydoros *et al.* [106] developed a deep learning framework for real-time learning of robotic controls through modeling the inverse dynamics of robotic manipulator joint torques from sensory data. Their model employed a self-organized layer to decorrelate inputs and a recursive reservoir to provide fading memory, which require no hyperparameter optimization or kernel selection. They additionally recorded and tested new datasets for inverse dynamics model learning, and demonstrated the adaptability of their model to changes in the inverse dynamics model.

A subfield of robotics, robotic vision affords autonomous systems with situation awareness via object detection. As applied to robotic manipulators, Mahler *et al.* [88] developed a Grasp Quality Convolutional Neural Network (GQ-CNN) model, called Dex-Net 2.0, to predict the probability of success of grasps from depth images. The authors developed a synthetic dataset of 6.7 million point clouds of 1,500 3D object models and grasp quality metrics. The model achieves a high success rate on both known and unknown objects, and is three times faster than the competing method. In addition, targeting more generalized detection for robotic vision, Poirson *et al.* [105] developed a model for simultaneous 3D object detection and pose estimation in a single deep CNN, greatly increasing the efficiency of detection and pose estimation over other state-of-the-art systems. They implemented a variant of the single shot detection (SSD) network with additional pose outputs, and conducted tests on the Pascal 3D+ dataset to verify design choices, such as shared pose output across objects, and combined Pascal and ImageNet annotated data with coarse-grained training for increased accuracy.

As applied in particular to autonomous vehicle systems, Dairi *et al.* [30] developed an unsupervised object detection system based on a hybrid implementation of deep Boltzman machines (DBMs), auto-encoders (AEs), and support vector machines (SVMs), utilizing stereovision as input. In their system, the DBMs and AEs are combined for feature extraction and encoding, and the SVMs are used for anomaly detection. In addition, Kahn *et al.* [64] presented a generalized computational graph-based deep reinforcement learning framework that combines value-based and model-based mechanisms, and applied this model to autonomous navigation using monocular images. The model was tested both in simulated and in real environments, and experiments were conducted to evaluate appropriate model horizons and bootstrapping efficacy, sample efficiency of classification and regression for value versus collision probability predictions, and performance against various double Q-learning approaches. Furthermore, in a rather unique application, Rajesh and Mantur [108] applied a deep CNN to eye movement and blink tracking for the control of an electric wheelchair. Trained on the Eye-Chimera dataset and implemented via head-mounted camera, the achieved accuracy is upwards of 99 %.

E. Medical Diagnostics

Highly influenced by advances in image analysis, medical diagnostics have benefited significantly from the rapid improvements in deep learning. Considerable work has been done toward improving the detection of diseases, tumors, and other abnormalities from MRI images, CT scans, etc. In addition, IoT devices for medical applications can provide autonomous monitoring of patients and extract useful data on medical populations.

In utilizing data from widely deployed smart IoT devices, advances have been made in increasing the accuracy of remote sensor metrics. For example, Jindal [63] utilized deep learning to increase the accuracy of heart rate estimation via photoplethysmography (PPG) by smartphones and wearables during exercise. The authors fuse PPG and accelerometer data, and utilize deep belief networks composed of Restricted Boltzman Machines (RBMs) implemented in the cloud to classify the PPG signals into subgroups. The PPG signal then undergoes particle filtering to predict the heart rate over time, achieving an average error of 4.88 %. Similarly, Ravì *et al.* [110] applied deep learning to human activity recognition using inertial sensor data from wearable devices. In their learning model, the authors combine deep convolutional learning in parallel with shallow feature extraction, converging in a fully connected network for more accurate classification. Their model

is implemented in Torch for Android and as an embedded algorithm for the Intel Edison Development Platform. Likewise, Schirrmeister *et al.* [130] investigated deep learning for the classification of electroencephalography (EEG) pathologies by signal analysis. Specifically, CNNs were developed and automatically optimized using Sequential Model-based Algorithm Configuration (SMAC), resulting in higher accuracy and specificity.

In addition to diagnostic for continuous sensed data, the application of deep learning to diagnostic scans and exam artifacts has produced significant results. For instance, Wang *et al.* [136] implemented deep learning for automated detection of metastatic breast cancer from lymph node biopsy images as part of the Camelyon Grand Challenge 2016. Their framework assessed patch level predictions via CNN, and aggregated patches to produce tumor probability heatmaps for localization prediction. Additionally, Gulshan *et al.* [133] applied deep learning for carrying out the detection of diabetic retinopathy and macular edema from retinal fundus images automatically. Their implemented framework, consisting of an ensemble of 10 CNNs with pre-initialized weights first trained on the ImageNet dataset. Also, Wang *et al.* [138] proposed a multi-scale rotation-invariant CNN architecture for classifying lung tissues from high-resolution computed tomography (HRCT) scans. The authors applied Gabor filtering and local binary pattern (LBP) feature extraction prior to CNN learning of interstitial lung disease (ILD) classes, and further implemented a mechanism to handle unbalanced data effectively.

F. Computational Biology

Though similar to applications for medical diagnostics, nonetheless we consider deep learning as applied to biological sciences an altogether different category. Specifically, these applications intersect the domain of chemical and molecular interactions, and investigate processes inherent to various micro- and macro-organisms. These can be considered approaches toward fine-grain understanding of the continuous mechanisms that produce a result.

There have been a number of research efforts in this area, as outlined by Angermueller *et al.* [16]. In their work, they presented a comprehensive review of deep learning mechanisms as applied to computational biology, and detail important areas in which deep learning has been applied. These include the prediction of DNA mutation affects and multi-trait prediction, and cellular and tissue image analysis. Considering predictive examples, Angermueller *et al.* [15] applied deep learning methods to predict methylation states in individual cells. The designed schemes are used to recognize known and *de novo* predictive sequence motifs. They applied a multi-network framework consisting of a DNA CNN module and a bidirectional gated RNN CpG module, which learns relationship between DNA sequence patterns and methylation states, as well as between neighboring CpG sites within and across cells.

Additional predictive uses of deep learning include [86] and [107]. Particularly, Ma *et al.* [86] presented DeepRT, a deep learning model for peptide retention time prediction for liquid chromatography-tandem mass spectrometry experiments. In their model, features are learned directly from peptide sequences in parallel CNN and LSTM networks, and further reduced via an ensemble of support vector regression, random forest, and gradient boosting. In addition, Qu *et al.* [107] presented a deep learning approach to predict DNA-binding proteins solely from their primary sequences using CNNs to detect function domains and LSTM to discover long-term dependencies. Their model demonstrated superior performance on both equalized and asymmetric datasets.

In use for image processing, deep learning has been applied to enhance various analytical tools. For instance, Kraus *et al.* [72] applied CNNs for the localization of subcellular components of yeast cells in high-content microscopy, and implemented activation maximization to visualize the learned feature morphology by applying and incrementally updating randomized green pixel channels to maximize the feature activation. They also tested their model on unseen yeast cell morphologies, and implemented a transfer learning process to incorporate additional features and apply the model to distinctly different microscopy techniques, which performed significantly better than from-scratch training. Also, Eulenberg *et al.* [37] applied a deep CNN with nonlinear dimensionality reduction for reconstructing continuous biological processes and t-distributed stochastic neighbor embedding (tSNE) visualization of flow cytometry images. Their method outperformed a comparable boosting-based approach, and the authors further applied their model to the progression of diabetic retinopathy from fundus images

G. Physical Sciences

Deep learning has significant potential for enhancing physical sciences via dimensionality reduction and the ability to achieve fine-grained analyses with expansive data in a generalized way. Indeed, astrophysical, geological, environmental, and quantum-mechanical sciences have benefited greatly, to name a few.

Regarding particle physics modeling and detection applications, Barberio *et al.* [20] expanded upon prior work to test the CP state of the Higgs boson via $H \rightarrow \tau\tau$ decays using deep neural network. In particular, they investigated the effects of detector resolution and the systematics of τ decay modeling on the sensitivity of the deep learning result, which remains largely stable. Likewise, Komiske *et al.* [69] tested CNNs for discriminating quark and gluon jet deposits in comparison with traditional observables designed by physicists. The resulting experiments showed that the CNN configuration is generally insensitive to the event generator (Pythia or Herwig). Deep learning has also been used in modeling atomic and molecular interactions in chemical and material sciences as well. For example, Schutt *et al.* [114] developed SchNet, a deep tensor neural network architecture for modeling atomic interactions for the prediction of potential-energy surfaces and exploration of chemical space. SchNet was trained on molecular properties, formation energies, chemical potentials, and was applied to path-integral molecular dynamics (PIMD) simulations with the MD17 benchmark set.

In the realm of astrophysics, deep learning has been applied to model and monitor various cosmological phenomena, as

well as for image analysis of dense telescope data. For instance, Gabbard *et al.* [40] investigated the use of deep convolutional learning to detect gravitational waves produced by binary black hole (BBH) mergers. Simulated data was generated of Gaussian noise negative cases, and Gaussian noise plus BBH wave signal positive cases, for training and testing, and tests demonstrated that the CNN framework is closely aligned with the more computationally expensive matched-filtering technique. In addition, Ma *et al.* [87] applied deep multimodal learning for solar radio burst classification, specifically treating different frequency channel spectrums as differing modes. The model is composed of autoencoders (AEs) for each channel mode, structured regularization for connecting the AEs to the fully connected network, concluding with softmax for classification into coarse burst, non-burst, and calibration classes.

Given the vast body of work validating human-caused climate change, it should be no surprise that the climate, geographic, geological, and meteorological fields have also employed deep learning. For example, Shao *et al.* [117] utilized stacked sparse autoencoders (SSAE) for wall-to-wall forest above-ground biomass (AGB) prediction. Their model combined discrete and simulated LiDAR-derived data as an AGB reference map, replacing traditional forest inventory, with optical and synthetic aperture radar (SAR) data from Landsat 8 OLI and Sentinel-1A satellite images. In addition, Ducournau and Fablet [35] applied a super-resolution CNN (SRCNN) to reconstruct high resolution images from low resolution ocean remote sensing data for satellite-derived sea surface temperature (SST) mapping. Evaluation was conducted to compare the mean peak signal-to-noise ratio (PSNR) gain of different CNN models, and further evaluation against comparable reconstruction methods (e.g., bicubic interpolation, EOF-based) showed improved performance.

H. Finance, Economics, Market Analysis and Others

As mechanisms for prediction and analysis, deep learning tools have the capacity to learn from stochastic data and recognize trends, such that machine learning-based systems have been widely developed for market prediction. In addition, verification and validation of monetary transactions greatly benefits from the potential data generated by users, and can be used to detect anomalous behavior.

The ability to accurately predict market fluctuations provides a material advantage in stock trading and investment. As a powerful predictive tool, deep learning for market and economic analysis has been highly investigated. For example, Korczak and Hernes [70] presented financial time-series forecasting utilizing deep learning architectures. Compared with multilayer perceptron (MLP), their CNN implementation in the H2O framework significantly decreases the forecasting error rate when trading on the FOREX market, and increases the average rate of return per transaction. In addition, Heaton *et al.* [51] considered deep learning for financial prediction and classification, particularly in the application of deep models over shallow models for high-dimensionality data reduction, high-dimensionality feature extraction for risk analysis, and event analysis classification. They also provide and example, namely smart indexing, to approximate a stock index through a subset of stocks.

Additionally, the wide variety of deep network architectures and data sources afford strikingly varied implementations. For instance, Fischer and Krauss [39] deployed long short-term memory (LSTM) networks, which are used to carry out the prediction of out-of-sample directional movements for S&P 500 stocks from 1992 to 2015. Yielding daily returns of 0.46 percent, their LSTM networks outperformed memory-free classification (random forest, DNN, etc.). Nonetheless, analysis of the results also reveals returns fluctuating around zero after about 2009, due to low exposure of the LSTM to systemic risks. Also, Hu *et al.* [55] deployed a Hybrid Attention Network (HAN) for the prediction of stock trends based on news reports. Their framework embedded attention values into news vectors, input multiple news vectors for temporal analysis in RNN, and further applied attention values to and RNNs for sequential modeling, and further applied temporal attention encoding and trend prediction. In addition, a self-paced learning training mechanism increased accuracy over the basic HAN, and both mechanisms outperformed competing methods in trading simulations.

Furthermore, Ding *et al.* [31] investigated event-driven stock market prediction through the implementation of a neural tensor network to learn event embeddings, and deep CNNs for short-, medium- and long-term event analysis. Their model significantly improved accuracy and profit in both individual stock prediction and market simulation experiments when compared to baseline neural network methods, especially for low fortune ranking companies for which news is less available. Also, Zhao *et al.* [166] introduced a deep learning ensemble approach for crude oil price forecasting using stacked deep autoencoders trained on bootstrap aggregation or bagging. Training and testing were conducted on the West Texas Intermediate (WTI) crude oil spot price series, and experiments demonstrated the improvements of bagging/ensemble architectures for improving neural network and SAE accuracy and error, with the designed SDAE-B performing the best.

Other aspects of market forecasting, include prediction of specific market segments and cycles, such as in the work by Zhao *et al.* [167], who utilized DBNs to predict customer mobile device or terminal replacement for use in marketing strategies and targeted sales. The authors utilized a combination of customer business data and collected customer device data, and compared their DBN results with several more shallow learning techniques, demonstrating improved performance. Similar predictive analytics mechanisms have been applied to various smart and autonomous systems, such as energy consumption forecasting, traffic prediction, user geolocation, etc. For example, Yu *et al.* [153] designed several machine learning-based schemes (e.g., neural networks and support vector machines) to carry out the forecasting of energy usage in the smart grid and conducted performance comparison using real-world smart meter dataset. Also, Wang *et al.* [137] proposed a deep learning scheme using an Error-feedback Recurrent Convolutional Neural Network structure (eRCNN) to carry out the prediction of traffic speed and

congestion.

I. Cyber Security

Given the unprecedented utilization of network connected devices, and the significant dependence on information communication technology (e.g., software applications, networked servers and end-user devices) throughout the world, it is no surprise that nefarious users attempt and occasionally succeed in subverting credentials, bypassing security systems, or simply attacking hosts with network traffic. Similar to existing research applying machine learning to conduct accurate cyber situation awareness [24], [56], [41], [49], [154], [158], the use of deep learning technologies for cyber security analysis and intrusion detection is highly relevant, as the majority of attacks use families of intrusive software that can be observed and classified.

Considering the prevalence of malicious software (malware), propagated by increasingly sophisticated obfuscation and techniques, the use of deep learning has been widely applied for highly accurate malware analysis and detection of previously unforeseen threats. For example, Zhu *et al.* [168] presented an Android malware detection tool called DeepFlow. The designed tool utilizes FlowDroid to obtain data flows from sensitive sources to sensitive sinks. the SUSI technique is also leveraged to transform the data flows feature granularity. They then classify the applications via beep belief network with the transformed dataflows as input. In addition, Ding *et al.* [32] extracted opcode sequences from Windows PE files for malware classification via DBN. Converting the opcode features to n-gram representations, the features were downselected by maximum information gain and document frequency. The authors demonstrate both the capacity of DBNs to perform classification, as well as to perform autoencoding for unsupervised feature selection to enhance the performance of shallow learning models (K-Nearest Neighbors, Decision Tree, etc.).

The detection of ongoing attacks in real time is paramount to enable timely response and mitigation techniques. Work to secure systems to attacks include that of Uwagbole *et al.* [132], who designed a system to detect and prevent SQL injection attacks via hybrid static and dynamic analysis utilizing deep learning techniques. Their proxy-based system combines pattern matching with numerical feature encoding for neural network and logistic regression classification. Similarly, Zolotukhin *et al.* [170] utilized stacked autoencoders (SAEs) for the detection of application layer distributed denial-of-service (DDoS) attacks in encrypted traffic. Without decrypting the traffic packets, the system extracts and clusters features into normal traffic patterns, conducting traditional anomaly detection for trivial DoS attacks. In addition, the SAE to detects attacks designed to mimic typical browser activity based on the reconstruction error of vectorized conversation traffic groups in time intervals. Additionally, Kim *et al.* [66] developed a network Intrusion Detection System (IDS) based on an LSTM recursive neural network. The model is trained on the KDD Cup 1999 dataset, which includes 22 attacks in 4 categories. In comparison with other neural networks using the same training data, the authors' model has the highest detection rate and accuracy, particularly on DoS attacks.

Deep learning is also applicable to the security of real-world interactions, primarily in situation awareness and analysis. For instance, Wang *et al.* [139] applied a deep convolutional architecture for person re-identification as they are viewed by different non-overlapping cameras. The network was first pre-trained on the ImageNet dataset, and then further tuned by training on the CUHK03 re-identification dataset. With only minor changes to the fully connected layers of the model in retraining on the second dataset, the authors are able to significantly improve the matching rate over the existing schemes. Also, in the realms of validation and authentication, the security of transactions, identity, and the physical world can be significantly enhanced through deep learning. For instance, Niimi [100] investigated the use of deep learning for credit card approval determination and transaction validation. The implemented framework is written in R and implemented in Amazon's EC2 cloud platform. Their experimental evaluation demonstrated similar accuracy to various shallow learning methods, but with higher precision.

J. Architectural and Algorithmic Enhancement

Necessary for the continued enhancement and progress of deep learning as a generalized framework for diverse applications, the development of state-of-the-art architectural and algorithmic implementations of deep networks is paramount. Particularly, the reduction in training and inference time through clever design, as well as the improvement of accuracy through multi-network ensemble approaches and automatic hyperparameter optimization, are necessary for the realization of deep learning in mobile, commodity, and IoT hardware.

A variety of efforts have been applied to analyzing the architectures, activations, optimizers, hyperparameters, etc. of deep learning models for particular tasks, as well as when applied more generally. For example, Keskar *et al.* [121] investigated the effects of batch size in DNN training via Stochastic Gradient Descent (SGD), namely that large batches converge to sharp minima and result in poor generalization. Further investigating possible remedies, attempts using data augmentation and conservative training fail to correct problem, while the most promising solution is dynamic sampling to gradually increase the batch size. Moreover, Francois Chollet [27] presented an analysis of depthwise separable convolutions and their relationship to convolutional inception architecture. Specifically, the designed extreme inception (Xception) architecture decouples the mapping of cross-channel correlations and spatial correlations in convolutional feature maps, and outperforms the Inception V3 architecture.

In addition, adaptive techniques have been developed to down-select or fine-tune deep learning models. These can be helpful in allowing both experts and non-experts to optimize parameters and architectures more quickly. For example, Cortes *et al.* [29] developed AdaNet, a set of algorithms to adaptiveley learn ANN network structure and weights utilizing explicit Rademacher complexity measures to define data-dependent learning and generalisation bounds. The algorithms are strongly convex, indicating a global solution,

and iteratively growing the structure of a neural network, and balancing complexity with risk minimization. Patel *et al.* [102] developed a deep learning framework based on the Deep Rendering Mixture Model (DRMM), a theoretical framework that explicitly models nuisance variation via rendering function. The hierarchical generative model can reproduce the CNN architecture through the relaxation of parameter constraints, and can improve upon CNN performance without hyperparameter tuning.

VI. Emerging Research Trends

Deep learning has thoroughly saturated commercial industries and suffused business applications. Yet, several primary needs have arisen and persist without any comprehensive solution. These emerging research trends, as outlined in Fig. 4 primarily encompass: acceleration and optimization, distributed deep learning in IoT and CPS, network management and control, and security in deep learning. These needs can be seen as highly interdependent, and arise directly from the emerging IoT/CPS paradigm and trustworthy computing, as well as due to growing needs for edge computing. In the following, we present these needs one at a time, elaborating upon the background and characteristics of each, and provide a sampling of future research.

A. Acceleration and Optimization

Due to the massive adoption of smart mobile and embedded devices that comprise IoT and support smart-world applications such as smart grid, smart transportation, smart cities, etc., network congestion and latency are only going to increase without a diverse array of complementary solutions [125], [143], [148], [82], [80], [161], [127], [147]. Advances in edge computing and in-device computing provide avenues to reduce network congestion by providing computing near to users and reducing communication needed to reach resource-rich services. In addition, advances in network infrastructure and technologies (5G, Software-Defined Networks, etc.) are also forthcoming [13], [155], [157]. Regarding the former, the acceleration and optimization of deep learning architectures through thoughtful design of software, hardware, and algorithms is driven by needs for low energy, low resource, cheap, and efficient computation. Notwithstanding neural network architecture design and algorithm implementation, which are continuously evaluated and improved upon, examples of emerging areas of deep learning acceleration include the design of programmable computational arrays, bare-hardware implementation, and stochastic computation mechanisms.

For instance, Lacey *et al.* [74] investigated the application of Field Programmable Gate Arrays (FPGAs) as alternative hardware to GPUs or CPUs for implementing deep learning networks due to their better performance per watt and flexibility in configuration. The authors consider the strengths of FPGAs, including the customizable hardware circuits for multithreading and parallelism, and architectures that can be tailored for the intended application. Morcel *et al.* [96] utilized signal flow graph reduction, fixed-point arithmetic, and modularity to design a deep learning accelerator to minimize Field-programmable gate array (FPGA) resource use. Using AlexNet as the case study, they compared an optimized CPU implementation with the developed FPGA-based implementation, realizing increased throughput and reduced energy usage. Kim *et al.* [65] explored the design of FPGA systems for fully-connected neural network hardware, analyzing synthesis constraints and using the MNIST dataset on Caffe as an example. Further, Li *et al.* [78] explored the application of Stochastic Computing (SC) as applied to the implementation of deep neural networks. In particular, they investigated hardware-oriented optimization of feature extraction blocks in CNNs through SC implementations of convolution, pooling, and activation, and the arrangement of pooling and activation with respect to hardware implementation. In addition, they developed equations to optimize Stochastic tanh calculations, and demonstrate reduction in absolute error by placing the activation ahead of pooling, structurally reverse that of traditional software-based feature extraction implementations.

In addition, Wang *et al.* [140] proposed Memsqueezer, an active on-chip memory design for low-overhead deep learning acceleration. It utilizes active buffers, network weight and intermediate data compression, and in-memory redundancy detection to boost performance and reduce memory size of CNN inference. Simulation results demonstrate the ability to significantly reduce energy consumption. Tang *et al.* [128] explored executing convolutional neural networks in IoT hardware to overcome the limitations of latency in offloading computation to the cloud. Utilizing a bare-metal ARM v7 system on a chip (SoC), the authors compare TensorFlow against the recent ARM Compute Library (ACL). Implementing a SqueezeNet architecture, and ARM NEON vector computation optimization, the results show that the ACL outperforms TensorFlow by nearly 150 ms in execution time, though memory usage and power consumption are higher in the ACL. The authors additionally describe their ongoing work of developing an integrated deep learning IoT ecosystem consisting of lightweight OS comprised of sensor interfacing, a compiler based on NNVM to optimized deep learning models, and a message passing framework based on Nanomsg. Also, Du *et al.* [34] proposed a CNN acceleration architecture for IoT devices using streaming data flows to achieve high efficiency. In their architecture, convolution can be carried out in parallel with maximum pooling, and filter decomposition allows large kernel size with only a small computation unit. In comparison with other acceleration designs, their architecture achieves higher peak throughput and greater energy efficiency on a smaller core area.

Despite these advances, challenges remain in the development of accelerators for deep learning, and further investigation is necessary in several particular areas. First, many of these studies on acceleration remain largely developmental, and more work is necessary to refine them for hardware implementation. While simulation studies can demonstrate the potential for various improvements, these concepts must be transferred to hardware to realize the potential benefits in actuality. Second, the combination and comparison of the many diverse acceleration mechanisms should be compared whenever possible, and combined where applicable. To this

end, various methods of acceleration should be quantitatively compared with each other, and not simply with CPU or GPU counterparts. There remains great promise in these areas, as deep learning is being driven by industry groups to provide more efficiency and reduce costs, and new experimental hardware devices are coming to market. In addition, how to automatically select features and parameters in deep learning and the development of science of deep learning remain challenging issues.

B. Distributed Deep Learning in IoT and CPS

We now discuss distributed deep learning applied to IoT and CPS.

1) Internet of Things: In considering the applications of deep learning for IoT, significant work has been carried out toward broadly applying those typical categories mentioned above (image/video/audio processing, text analysis, etc.) across centralized and distributed cloud computing frameworks, utilizing IoT devices and some novel mechanisms [85], [67], [95], [142], [134], [160], [101].

For instance, Kim [67] proposed a deep learning system for use in identifying and tracking motion of individuals via Channel State Information (CSI) of IoT devices and widely deployed MIMO enabled access points. Mohammadi *et al.* [95] developed a semi-supervised deep reinforcement learning system to support smart city applications based on both structured and unstructured data. Utilizing Variational Autoencoders (VAEs), the authors studied indoor user localization using Bluetooth Low Energy (BLE), collecting the received signal strength indicator (RSSI) from a grid of iBeacon devices. In addition, Wu *et al.* [142] developed an efficient road scene segmentation deep learning model for embedded devices, termed ApesNet. Via time profiling and analysis, the authors developed an asymmetric encoder-decoder network, and limited the size of large feature maps in convolutional layers. In comparison with a complementary encoder-decoder network, ApesNet improves accuracy and reduces model size and runtime when tested on CamVid and Cityscapes datasets. Valipour *et al.* [134] developed a deep convolutional network for parking stall vacancy detection. Designed with existing parking lot cameras and infrastructure in mind, the system implemented and provides web and mobile interfaces for users. Additionally, the inference time of their model running on embedded Raspberry Pi architecture was only 0.22 seconds.

Despite these and many other works, there remain several critical issues which have yet to be resolved. Particularly, while a number of early efforts have shown the potential to run inference operations in IoT devices, the training of deep models in IoT hardware remains a practical impossibility. Nonetheless, local training of distributed and partial neural network input in IoT devices provides an opportunity to reduce network overhead and latency in training by offloading pre-trained feature output for additional training at higher layers. This would be particularly practical for image and object recognition processing offloaded to edge computing nodes, where the dimensionality of transmitted data can be reduced. Some relevant examples include [160] and [101]. For example, Yuan and Jia [160] proposed and demonstrated the use of sparse autoencoder networks on distributed servers to perform anomaly detection on smart electricity meter data. The distributed slave nodes perform the anomaly detection individually, alerting the master node and reducing computing overhead for the centralized master node, and outperform complementary learning algorithms. Park *et al.* [101] designed a Situation Reasoning framework that extracts multiple low-level contexts in DNN modules, and combines them in a higher level Situation Reasoning module based on the Feature Comparison Model of cognitive psychology. Utilizing the spatio-temporal contexts of IoT data, the author's framework demonstrates good performance in comparison with other Situation Reasoning methods.

While the examples provided show some promise, significant work must still be done. Given that IoT systems facilitate near-infinite potential for integrating deep learning networks for innumerable applications, the development of appropriate paradigms to analyze said data in a timely manner is imperative. While not all applications will require real-time analysis and inference, those that converge with critical infrastructure and safety applications surely will. The requirements of such real-time functionality can be considered from two domains, distributed deep learning at the network edge, and in-device deep learning.

In the first case, distributed deep learning is a solution to the inability to resolve deep learning in-device because of complexity and processing power. This almost certainly necessitates the intervention of edge computing to offset network latency that will critically reduce the effectiveness of the target learning system. To this end, though the edge computing paradigm has recently seen significant study, the intersection of edge computing infrastructures with deep learning remains to be thoroughly investigated. Specifically, parallel simultaneous learning network implementations for edge architectures should be developed and optimized for self-organization and runtime.

In the second case, we consider that deep learning inference has only recently been realized in IoT hardware, with scalability at cost still on the horizon. In general, then, the implementation of deep networks in IoT devices is a preeminent concern that requires continued investigation. This development is significantly affected by advances in hardware and computational capabilities. In addition, in-device deep learning provides the potential for reductions in network overhead in terms of data transfer and signaling, the impact of which has yet to be considered.

2) Cyber-Physical Systems: In addition, Cyber-Physical Systems (CPS), more than just network connected devices like IoT, include the vertical layering of IoT devices, networking, service, applications, and command and control (C&C) platforms. Examples of CPS systems include smart transportation system with self-driving vehicles, smart cities, smart electrical grids, etc. [94], [58], [125], [143], [148], [82], [80], [161], [127], [147], [145], [165]. More specifically, as applied to power generation, monitoring and control, Mocanu *et al.* [94] utilized Factored Four-Way Conditional Restricted Boltzmann Machines (FFW-CRBMs) and Disjunctive Factored Four-Way

Conditional Restricted Boltzmann Machines (DFFW-CRBMs) to carry out energy disaggregation, and flexibility classification and prediction, on smart appliance data. Likewise, Liangzhi *et al.* [58] investigated electrical load forecasting in the smart grid via deep learning. Utilizing seven years of smart meter and IoT device data, the designed system first forecasts daily total consumption via DNN with complex input features, and then predicts intra-day load variation by applying the daily consumption prediction, along with a more limited set of features, to a second DNN. In addition, Zhao *et al.* leverage convolutional neural networks to develop a new deep heartbeat classification system, which can accurately analyzing the raw electrocardiogram (ECG) signal in healthcare smart-world system.

As applied to CPS, distributed deep learning takes on a new dimension, as we consider the distinct layered structure, and the heterogeneity of the data within. In this case, we shall integrate the aforementioned distributed IoT context with multi-modal data fusion. Relevant examples include [141] and [76], the former of which also seeks to adapt new features to a pre-trained model to improve the overall system. Particularly, Wang *et al.* [141] explored image classification in CPS, and proposed a fast feature fusion algorithm. The authors extracted features from various deep and shallow learning mechanisms in parallel, utilized a Genetic Algorithm (GA) to convert those features into weights for a fusion feature vector, and introduced partial selection to choose a classifier. Using this mechanism, pre-trained neural network models could be combined with original models, which add new features and classes to outperform any single constituent model.

Further, Li *et al.* [76] proposed a deep convolutional computation model, which is used for conducting hierarchical feature learning on IoT big data. Utilizing tensor representation, which preserves raw data structures and thus mutuality and complementarity, they can better represent hierarchical multi-modal data. Designing tensor-based convolution, pooling, and fully-connected layers, as well as high-order back propagation, the authors demonstrate the effectiveness of their approach against multi-modal deep learning, as well as deep computation models on three datasets (CUAVE, SNAE2, and STL-10). Another relevant work is to adapt the ST-ResNet structure to predict the hourly distribution of crime in parceled areas in the city of Los Angeles [135]. In this work, the necessary spatial and temporal resolutions for optimal prediction were investigated, and a ternarization of the model was additionally developed to reduce model size and execution time, with a minor increase in error.

Finally, in considering CPS, autonomous command and control can be distributed to the lowest levels necessary via deep learning for in-time analysis, and can be configured uniquely for each layer. In this way, resource use can be reduced throughout the system. Indeed, if IoT enables the convergence of many technologies (networking, distributed computing, deep learning, big data, etc.), then CPS compounds this through the imperative of infrastructure security and communication. This, then, presents a challenging issue of how to integrate the various CPS layers that include deep learning mechanisms. For instance, in considering smart vehicle technologies, autonomous transportation must enable inter-vehicle communication, but must also communicate both the smart transportation infrastructure in transit, and smart city and smart grid infrastructures in locating parking and acting as secondary electricity storage to enhance grid function. We can envision many such communication exchanges across different domains, including in user identification and tracking, autonomous services such as delivery or manufacturing, and even in localized network and electrical load prediction via massive fine-grained IoT device transit data. Further investigation is necessary to understand hierarchically combined deep learning models and the policies to optimize and secure their use in critical infrastructure systems, as well as best practices for managing and updating individual aspects of such a system using deep learning.

C. Network Management and Control

In future 5G broadband systems [13], [157] and other networking systems, as well as evolving traditional network infrastructures, complex heterogeneous protocols, interfaces, and hardware will be massively implemented to realize throughput and bandwidth gains, supporting a massive number of users with diverse quality of service requirements. Solutions to the growing complexity and need for agile service include software-defined networking (SDN), network function visualization, and edge computing paradigms. While these technologies are indeed poised to provide solutions to address future network challenges, the architecture, management, and security of these future networks will be highly dependent on the effective optimization of services and hardware. In this regard, deep learning offers a viable technique that can effectively learn the characteristics of the network and the behavior of users, leading to better network management and control decisions and outcomes. Furthermore, the massive increase in users, including humans and autonomous machine-to-machine equipment, will necessitate analysis, density estimation, and complexity reduction to handle such massive data. With the continuous developments of deep learning, these challenges can be resolved, yet thorough research is necessary to achieve these goals.

In this regard, very little research has been conducted. For example, Zhu *et al.* [169] implemented stacked auto-encoders (SAEs) to realize Q-learning for transmission scheduling in cognitive IoT relays. Modeling the system as a Markov decision process, and seeking to maximize system utility, a simulation evaluation shows improved performance over W-learning, but not strategy iteration. Nonetheless, strategy iteration considers all states of the system at a given time, instead of the current state, and is not scalable. Likewise, Lopez-Martin *et al.* [83] demonstrated flow statistics-based network traffic classification via deep neural networks. Using only packet headers, the authors investigated the use of RNN, CNN, and combined CNN/RNN models, convolving over the time series of the incoming data. Their designed models demonstrate good performance, especially on labels with a frequency higher than 1 %. It is worth noting that not all CNN/RNN models outperform the basic RNN. Aminanto *et al.* [14] developed

a three-layer Wi-Fi impersonation attack detection system. In their tiered model, Stacked Autoencoders first performed feature extraction on the original dataset. Feature selection was then performed via ANN, SVM, or Decision Tree, on the original data plus the newly extracted features, and an ANN was used for final classification. Their system achieves a 99.918 % detection rate and 0.012 % false positive rate, and significantly outperforms the comparable systems.

Despite the works outlined here, the majority of applications for deep learning in network management and control remain unexplored. Indeed, deep learning has the potential to fundamentally transform network design, management, and service through integration with advanced architecture such as cognitive radio and network function virtualization, as well as in optimization and analysis to enable adaptability and autonomy. For instance, deep learning models can be applied to learn the characteristics of the network and the behaviors of connected users. In this way, optimal decisions (e.g., routing optimization and node placement) can be made. In addition, network traffic analysis could be implemented in routing devices and utilized for traffic offloading, or implemented to provide hierarchical prioritized relay. While we have yet to see significant research in this regard, this area is garnering increased attention. Additionally, this remains a challenging area, due to the limitations of hardware for deep learning and the latency that deep learning may introduce into networking systems, as the smaller time scale necessary for inference (compared to training) cannot be considered trivial. Furthermore, traditional network transmission considerations are aimed at minimizing data size and transmission frequency to reduce network load. Nonetheless, in the context of deep learning, additional data generally increases the accuracy of the learning model. Thus, a balance must be struck that satisfies the needs of any implemented deep learning system with those of congestion reduction, quality of service, energy efficiency, and latency. Furthermore, the automation of network management via intelligent networked systems must be scalable, secure, and fault-tolerant.

D. Secure Deep Learning

Given the increasing number of devices, operating systems, and communication protocols that abound in IoT, security is an ever-ballooning problem [151], [82]. Securing the data, operation, and mechanisms of deep learning are all the more relevant in considering edge computing, which can be a viable computing infrastructure to provision deep learning schemes [155], supporting a variety of smart-world systems (smart cities, smart manufacturing, smart grid, smart transportation, and many others). As computing nodes will be more dispersed and local to the user, they will also have fewer resources and be more available to would-be adversaries. The investigation and application of increasingly sophisticated security mechanisms, such as homomorphic encryption, are thus significant. For example, Li *et al.* [77] proposed multiple schemes for machine learning on multi-key homomorphic encrypted data in the cloud. In the first scheme, deep learning is conducted on multiple users' data who share the same public key. In the second scheme, using double decryption, training is performed on ciphertexts of users with different public keys. While these are novel methods that leverage the state-of-the-art in security research, encrypting not only data, but computation as well, they can still be considered as expansions of traditional security techniques.

In exception to traditional mechanisms, attacks that seek to undermine the output of deep learning systems have recently received deeper consideration. Indeed, the widespread adoption of machine learning is cause for concern, as attacks that are solely intended to thwart the normal operation of the learning network can lead to catastrophic harm. For instance, a recent work by Yuan *et al.* [159] specifically investigated the space of these attacks that target only the inference mechanism through adversarial input. The authors classified no less than sixteen different attack methods, which have been shown to be effective against various targets, including subverting segmentation (removal of objects from detection) and facial recognition. This is similar to an investigation by Huang *et al.* [57] from 2011, which investigated security in machine learning and provided a taxonomy for causative and exploratory attacks, and formulated game-theory-based formalisms to understand each attack. Nonetheless, the latter focused on the shallow learning methods of the time.

In the interim, Goodfellow *et al.* [45] proposed the generative adversarial network, pitting generator and discriminator networks against one another in a minimax game. Here, a discriminator is used to discern the data distribution from the generated model distribution via a learning process, while a generator learns to better undermine the discriminator, improving both in the process. A relevant insight observed from the development of GANs is that they do not necessarily resolve adversarial examples: those which fail in ways that are imperceptible to humans, or succeed while not retaining any of the human-perceptible attributes. Though GANs have been used to great effect in increasing accuracy in generative and discriminative networks, they fail to address problems posed by these corner cases.

While Yuan *et al.* [159] did point out various defensive mechanisms against adversarial input, such as network distillation, adversarial retraining, adversarial detection, and input reconstruction, significantly more work is needed. In addition, Pei *et al.* [104] developed DeepXplore, the first whitebox testing framework for evaluating deep learning systems. Their work developed the concept of neuron coverage, which is characterized as the amount of deep network logic or neurons activated by a given input. They also leveraged multiple complementary deep networks as cross-referencing oracles, and formalized the maximization of neuron coverage and differential behaviors as a joint optimization problem with gradient ascent. Beyond demonstrating the effectiveness of their framework in terms of runtime and neuron coverage, they also leveraged their framework to augment network training to demonstrably improve accuracy. In addition, Booz *et al.* [22] investigated how to fine-tune parameters of deep learning to improve the accuracy of detecting Android malware.

Though considerations for the limitations of deep learning go back a few years, and adversarial learning has helped

increase the accuracy achieved in training models, as well as the generation of unique data, significant work is still needed to secure deep learning systems. In particular, further study is necessary to fully develop standardized testing practices for deep learning to reveal hidden or unforeseen vulnerabilities. This should include two primary directions: securing deep learning models, and securing deep learning systems. While the latter can be considered within the traditional realm of security analysis and prevention, nonetheless, deep learning should be further applied to enhance security detection systems at all levels. Examples include deep learning for static and dynamic analysis in intrusion detection to secure deep learning systems and underlying architectures. In addition, the verification of appropriateness and resiliency of the trained deep learning models, as well as their further improvement, should be the primary goals of future adversarial investigations.

Of particular interest are attacks against deep models, as inappropriately or insufficiently tested models may be easily subverted by and attacker, causing damage to digital or physical property, and potentially endangering human lives. Deep learning, like many technologies, is a double-edged sword that can be used by both adversaries and defenders in the cybersecurity field. In fact, advancements in deep learning are likely to have a profound impact on future cyber attacks, as attackers leverage the technology to enact more encompassing, effective, autonomous, and potentially novel attacks. Therefore, systematically investigating threats in the full lifecycle (training and inference) of deep learning in its use for cybersecurity, in addition to adversarial input, becomes critical. Further, understanding the capabilities of deep learning in detecting cyber threats and investigating how to optimize deep learning networks to achieve the highest detection accuracy, dealing with both known and unknown threats remain challenging issues.

VII. Final Remarks

Deep learning is a technology that continues to mature, and has clearly been applied to a multitude of applications and domains to great effect. While the full-scale adoption of deep learning technologies in industry is ongoing, measured steps should be taken to ensure appropriate application of deep learning, as the subversion of deep learning models may result in significant loss of monetary value, trust, or even life in extreme cases. In this survey, we have provided an overview of deep learning operation, distinguishing deep learning from traditional shallow learning methods, and outlining prominent structural implementations. We have reviewed deep learning architectures in detail based on learning mechanisms (supervised, unsupervised, and reinforcement) and the target output structures, and provided typical examples in each case. We have also introduced many common and widely adopted deep learning frameworks, and considered them from the perspectives of design, extensibility and comparative efficacy. It is worth mentioning that each of the frameworks implements the basic elements of deep learning in different ways using different libraries, are optimized for different hardware systems, and provide varying degrees of control over model design.

Additionally, we have thoroughly investigated the state-of-the-art in deep learning research. These categories include multimedia (audio, visual, and text) processing, autonomous systems, medical diagnostics, biological and physical sciences, financial applications, security analysis, and algorithmic enhancement. Finally, having surveyed the landscape of completed works, we have highlighted areas in which deep learning research has yet to make significant strides, or where significant advances are immediately forthcoming. These include acceleration and optimization of deep learning via fundamental hardware and encoding methods, distributed deep learning for IoT and CPS, network management and control applications of deep learning, and, perhaps most importantly, securing deep learning models and systems. Given the widespread adoption of deep learning, especially in multimedia fields, and the inevitability of increasingly sophisticated cyber threats, the development of mechanisms to harden systems against adversarial data input is imperative. We hope this work provides a valuable reference for researchers and computer science practitioners alike in considering the techniques, tools, and applications of deep learning, and provokes interest into areas that desperately need further consideration.

Acknowledgement

This work was supported in part by the US National Science Foundation (NSF) under grants: CNS 1350145 (Faculty CAREER Award), and the University System of Maryland (USM) Endowed Wilson H. Elkins Professorship Award Fund. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

References

[1] Apache mxnet: A flexible and efficient library for deep learning. 2017. https://mxnet.apache.org/.
[2] Caffe. 2017. http://caffe.berkeleyvision.org/.
[3] Caffe2: A new lightweight, modular, and scalable deep learning framework. 2017. https://caffe2.ai/.
[4] Cloud tpu alpha: Train and run machine learning models faster than ever before. 2017. https://cloud.google.com/tpu/.
[5] Deep learning: For data scientists who need to deliver. 2017. https://skymind.ai/.
[6] Deep learning for java: Open-source, distributed, deep learning library for the jvm. 2017. https://deeplearning4j.org/.
[7] Keras: The python deep learning library. 2017. https://keras.io/.
[8] The microsoft cognitive toolkit. 2017. https://docs.microsoft.com/en-us/cognitive-toolkit/.
[9] An open-source software library for machine intelligence. 2017. https://www.tensorflow.org/.
[10] Theano. 2017. http://deeplearning.net/software/theano/.
[11] Torch: A scientific computing framework for luajit. 2017. http://torch.ch/.
[12] A. Adler, D. Boublil, M. Elad, and M. Zibulevsky. A Deep Learning Approach to Block-based Compressed Sensing of Images. *ArXiv e-prints*, June 2016.
[13] M. Agiwal, A. Roy, and N. Saxena. Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 18(3):1617–1655, thirdquarter 2016.
[14] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim. Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Transactions on Information Forensics and Security*, 13(3):621–636, March 2018.
[15] C. Angermueller, H. J. Lee, W. Reik, and O. Stegle. Deepcpg: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biology*, 18(1):67, Apr 2017.

[16] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle. Deep learning for computational biology. *Molecular Systems Biology*, 12(7), 2016.

[17] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77(Supplement C):236 – 246, 2017.

[18] D. M. S. Arsa, G. Jati, A. J. Mantau, and I. Wasito. Dimensionality reduction using deep belief network in big data case study: Hyperspectral image classification. In *2016 International Workshop on Big Data and Information Security (IWBIS)*, pages 71–76, Oct 2016.

[19] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, Nov 2017.

[20] E. Barberio, B. Le, E. Richter-Was, Z. Was, D. Zanzi, and J. Zaremba. Deep learning approach to the Higgs boson CP measurement in $H \rightarrow \tau\tau$ decay and associated systematics. *Phys. Rev.*, D96(7):073002, 2017.

[21] A. Bonarini, A. Lazaric, F. Montrone, and M. Restelli. Reinforcement distribution in fuzzy Q-learning. *Fuzzy Sets and Systems*, 160(10):1420 – 1443, 2009. Special Issue: Fuzzy Sets in Interdisciplinary Perception and Intelligence.

[22] J. Booz, J. McGiff, W. G. Hatcher, W. Yu, and C. Lu. Tuning deep learning performance for android malware detection. In *Technical Report, Towson University, 2018*, Feb. 2018.

[23] T. S. Borkar and L. J. Karam. Deepcorrect: Correcting DNN models against image distortions. *CoRR*, abs/1705.02406, 2017.

[24] A. L. Buczak and E. Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, Secondquarter 2016.

[25] X. W. Chen and X. Lin. Big data deep learning: Challenges and perspectives. *IEEE Access*, 2:514–525, 2014.

[26] Z. Chen, H. Zhang, W. G. Hatcher, J. Nguyen, and W. Yu. A streaming-based network monitoring and threat detection system. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 31–37, June 2016.

[27] F. Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *ArXiv e-prints*, Oct. 2016.

[28] O. Cominetti, A. Matzavinos, S. Samarasinghe, D. Kulasiri, S. Liu, and P. K. Maini. DifFUZZY: a fuzzy clustering algorithm for complex datasets. 1, 01 2010.

[29] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang. AdaNet: Adaptive Structural Learning of Artificial Neural Networks. *ArXiv e-prints*, July 2016.

[30] A. Dairi, F. Harrou, M. Senouci, and Y. Sun. Unsupervised obstacle detection in driving environments using deep-learning-based stereovision. *Robotics and Autonomous Systems*, pages –, 2017.

[31] X. Ding, Y. Zhang, T. Liu, and J. Duan. Deep learning for event-driven stock prediction. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 2327–2333. AAAI Press, 2015.

[32] Y. Ding, S. Chen, and J. Xu. Application of deep belief networks for opcode based malware detection. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3901–3908, July 2016.

[33] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. In *2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–6, June 2016.

[34] L. Du, Y. Du, Y. Li, J. Su, Y. C. Kuan, C. C. Liu, and M. C. F. Chang. A reconfigurable streaming deep convolutional neural network accelerator for Internet of Things. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 65(1):198–208, Jan 2018.

[35] A. Ducournau and R. Fablet. Deep learning for ocean remote sensing: an application of convolutional neural networks for super-resolution on satellite-derived sst data. In *2016 9th IAPR Workshop on Pattern Recogniton in Remote Sensing (PRRS)*, pages 1–6, Dec 2016.

[36] N. Ekedebe, C. Lu, and W. Yu. Towards experimental evaluation of intelligent transportation system safety and traffic efficiency. In *2015 IEEE International Conference on Communications (ICC)*, pages 3757–3762, June 2015.

[37] P. Eulenberg, N. Köhler, T. Blasi, A. Filby, A. E. Carpenter, P. Rees, F. J. Theis, and F. A. Wolf. Reconstructing cell cycle and disease progression using deep learning. *Nature Communications*, 8(1):463, 2017.

[38] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE Communications Surveys and Tutorials*, 19(4):2432–2455, Fourthquarter 2017.

[39] T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, pages –, 2017.

[40] H. Gabbard, M. Williams, F. Hayes, and C. Messenger. Matching matched filtering with deep networks in gravitational-wave astronomy. *ArXiv e-prints*, Dec. 2017.

[41] L. Ge, H. Zhang, G. Xu, W. Yu, C. Chen, and E. P. Blasch. Towards mapreduce based machine learning techniques for processing massive network threat monitoring data. *Networking for Big Data, published by CRC Press & Francis Group, USA, Yu, S. (Ed.), Lin, X. (Ed.), Misic, J. (Ed.), Shen, X.*, 2015.

[42] L. L. Ge, Y. H. Wu, B. Hua, Z. M. Chen, and L. Chen. Image registration based on SOFM neural network clustering. In *2017 36th Chinese Control Conference (CCC)*, pages 6016–6020, July 2017.

[43] J. A. Gonzalez, L. A. Cheah, A. M. Gomez, P. D. Green, J. M. Gilbert, S. R. Ell, R. K. Moore, and E. Holdsworth. Direct speech reconstruction from articulatory sensor data by machine learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2362–2374, Dec 2017.

[44] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[45] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[46] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3389–3396, May 2017.

[47] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu. Multi-view 3D object retrieval with deep embedding network. *IEEE Transactions on Image Processing*, 25(12):5526–5537, Dec 2016.

[48] W. G. Hatcher, J. Booz, J. McGiff, C. Lu, and W. Yu. Edge computing based machine learning mobile malware detection. In *National Cyber Summit*, 2017.

[49] W. G. Hatcher, D. Maloney, and W. Yu. Machine learning-based mobile threat monitoring and detection. In *2016 IEEE 14th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 67–73, June 2016.

[50] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *ArXiv e-prints*, Dec. 2015.

[51] J. B. Heaton, N. G. Polson, and J. H. Witte. Deep Learning in Finance. *ArXiv e-prints*, Feb. 2016.

[52] S. Hickson, A. Angelova, I. Essa, and R. Sukthankar. Object category learning and retrieval with weak supervision. *ArXiv e-prints*, Jan. 2018.

[53] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner. Early Visual Concept Learning with Unsupervised Deep Learning. *ArXiv e-prints*, June 2016.

[54] C. Hong, J. Yu, R. Xie, and D. Tao. Weakly supervised hand pose recovery with domain adaptation by low-rank alignment. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 446–453, Dec 2016.

[55] Z. Hu, W. Liu, J. Bian, X. Liu, and T.-Y. Liu. Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. *ArXiv e-prints*, Dec. 2017.

[56] H. H. Huang and H. Liu. Big data machine learning and graph analytics: Current state and future challenges. In *2014 IEEE International Conference on Big Data (Big Data)*, pages 16–17, Oct 2014.

[57] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence*, AISec '11, pages 43–58, New York, NY, USA, 2011. ACM.

[58] Y. Huang, X. Ma, X. Fan, J. Liu, and W. Gong. When deep learning meets edge computing. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*, volume 00, pages 1–2, Oct. 2017.

[59] E. P. Ijjina and K. M. Chalavadi. Human action recognition in rgb-d videos using motion sequence information and deep learning. *Pattern Recognition*, 72(Supplement C):504 – 516, 2017.

[60] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, 72(Supplement C):9 – 18, 2018.

[61] S. Jain and J. Dhar. Image based search engine using deep learning. In *2017 Tenth International Conference on Contemporary Computing (IC3)*, pages 1–7, Aug 2017.

[62] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[63] V. Jindal. Integrating mobile and cloud for PPG signal selection to monitor heart rate during intensive physical exercise. In *2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, pages 36–37, May 2016.

[64] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine. Self-supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation. *ArXiv e-prints*, Sept. 2017.

[65] J. Kim, J. Kim, B. Kim, M. Lee, and J. Lee. Hardware design exploration of fully-connected deep neural network with binary parameters. In *2016 International SoC Design Conference (ISOCC)*, pages 305–306, Oct 2016.

[66] J. Kim, J. Kim, H. L. T. Thu, and H. Kim. Long short term memory recurrent neural network classifier for intrusion detection. In *2016 International Conference on Platform Technology and Service (PlatCon)*, pages 1–5, Feb 2016.

[67] S. C. Kim. Device-free activity recognition using CSI big data analysis: A survey. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 539–541, July 2017.

[68] 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*, 19(4):2392–2431, Fourthquarter 2017.

[69] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz. Deep learning in color: towards automated quark/gluon jet discrimination. *Journal of High Energy Physics*, 2017(1):110, Jan 2017.

[70] J. Korczak and M. Hemes. Deep learning for financial time series forecasting in A-Trader system. In *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 905–912, Sept 2017.

[71] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, M. S. Gerber, and L. E. Barnes. HDLTex: Hierarchical Deep Learning for Text Classification. *ArXiv e-prints*, Sept. 2017.

[72] O. Z. Kraus, B. T. Grys, J. Ba, Y. Chong, B. J. Frey, C. Boone, and B. J. Andrews. Automated analysis of high-content microscopy data with deep learning. *Molecular Systems Biology*, 13(4), 2017.

[73] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[74] G. Lacey, G. W. Taylor, and S. Areibi. Deep Learning on FPGAs: Past, Present, and Future. *ArXiv e-prints*, Feb. 2016.

[75] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep Reinforcement Learning for Dialogue Generation. *ArXiv e-prints*, June 2016.

[76] P. Li, Z. Chen, L. T. Yang, Q. Zhang, and M. J. Deen. Deep convolutional computation model for feature learning on big data in Internet of Things. *IEEE Transactions on Industrial Informatics*, 14(2):790–798, Feb 2018.

[77] P. Li, J. Li, Z. Huang, T. Li, C.-Z. Gao, S.-M. Yiu, and K. Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74(Supplement C):76 – 85, 2017.

[78] Z. Li, A. Ren, J. Li, Q. Qiu, Y. Wang, and B. Yuan. DSCNN: hardware-oriented optimization for stochastic computing based deep convolutional neural networks. In *2016 IEEE 34th International Conference on Computer Design (ICCD)*, pages 678–681, Oct 2016.

[79] F. Liang, W. Yu, D. An, Q. Yang, X. Fu, and W. Zhao. A survey on big data market: Pricing, trading and protection. *IEEE Access*, 2018.

[80] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao. A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology*, 66(3):2551–2566, March 2017.

[81] J. Lin, W. Yu, N. Zhang, X. Yang, and L. Ge. On data integrity attacks against route guidance in transportation-based cyber-physical systems. In *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pages 313–318, Jan 2017.

[82] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, Oct 2017.

[83] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret. Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. *IEEE Access*, 5:18042–18050, 2017.

[84] D. Luo, H. Wu, and J. Huang. Audio recapture detection using deep learning. In *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pages 478–482, July 2015.

[85] X. Luo, Y. Lv, M. Zhou, W. Wang, and W. Zhao. A laguerre neural network-based ADP learning scheme with its application to tracking control in the Internet of Things. 20, 04 2016.

[86] C. Ma, Z. Zhu, J. Ye, J. Yang, J. Pei, S. Xu, R. Zhou, C. Yu, F. Mo, B. Wen, and S. Liu. DeepRT: deep learning for peptide retention time prediction in proteomics. *ArXiv e-prints*, May 2017.

[87] L. Ma, Z. Chen, L. Xu, and Y. Yan. Multimodal deep learning for solar radio burst classification. *Pattern Recognition*, 61:573 – 582, 2017.

[88] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio Ojea, and K. Goldberg. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. *ArXiv e-prints*, Mar. 2017.

[89] M. S. Mahmud, H. Wang, E. E. Alam, and H. Fang. A real time and non-contact multiparameter wearable device for health monitoring. In *2016 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, Dec 2016.

[90] N. Majumder, S. Poria, A. Gelbukh, and E. Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, Mar 2017.

[91] S. Mallapuram, N. Ngwum, F. Yuan, C. Lu, and W. Yu. Smart city: The state of the art, datasets, and evaluation platforms. In *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pages 447–452, May 2017.

[92] D. Marquardt and S. Doclo. Noise power spectral density estimation for binaural noise reduction exploiting direction of arrival estimates. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 234–238, Oct 2017.

[93] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529 EP –, Feb 2015.

[94] D. C. Mocanu, E. Mocanu, P. H. Nguyen, M. Gibescu, and A. Liotta. Big IoT data mining for real-time energy disaggregation in buildings. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 003765–003769, Oct 2016.

[95] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J. S. Oh. Semi-supervised deep reinforcement learning in support of iot and smart city services. *IEEE Internet of Things Journal*, PP(99):1–1, 2017.

[96] R. Morcel, H. Akkary, H. Hajj, M. Saghir, A. Keshavamurthy, R. Khanna, and H. Artail. Minimalist design for accelerating convolutional neural networks for low-end FPGA platforms. In *2017 IEEE 25th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pages 196–196, April 2017.

[97] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-DoF global localization in outdoor environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1525–1530, Sept 2017.

[98] N. D. Nguyen, T. Nguyen, and S. Nahavandi. System design perspective for human-level agents using deep reinforcement learning: A survey. *IEEE Access*, 5:27091–27102, 2017.

[99] M. Nielsen. Neural networks and deep learning. 2017. http://neuralnetworksanddeeplearning.com/.

[100] A. Niimi. Deep learning for credit card data analysis. In *2015 World Congress on Internet Security (WorldCIS)*, pages 73–77, Oct 2015.

[101] S. Park, M. Sohn, H. Jin, and H. Lee. Situation reasoning framework for the internet of things environments using deep learning results. In *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, pages 133–138, Sept 2016.

[102] A. B. Patel, M. T. Nguyen, and R. Baraniuk. A probabilistic framework for deep learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2558–2566. Curran Associates, Inc., 2016.

[103] M. Paul. Multiclass and multi-label classification. 2017. http://cmci.colorado.edu/classes/INFO-4604/files/slides-7_multi.pdf.

[104] K. Pei, Y. Cao, J. Yang, and S. Jana. DeepXplore: Automated Whitebox Testing of Deep Learning Systems. *ArXiv e-prints*, May 2017.

[105] P. Poirson, P. Ammirato, C. Y. Fu, W. Liu, J. Kosecka, and A. C. Berg. Fast single shot detection and pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 676–684, Oct 2016.

[106] A. S. Polydoros, L. Nalpantidis, and V. Krüger. Real-time deep learning of robotic manipulator inverse dynamics. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3442–3448, Sept 2015.

[107] Y.-H. Qu, H. Yu, X.-J. Gong, J.-H. Xu, and H.-S. Lee. On the prediction of DNA-binding proteins only from primary sequences: A deep learning approach. *PLOS ONE*, 12(12):1–18, 12 2017.

[108] A. Rajesh and M. Mantur. Eyeball gesture controlled automatic wheelchair using deep learning. In *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, pages 387–391, Dec 2017.

[109] D. Ramachandram and G. W. Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, Nov 2017.

[110] D. Ravì, C. Wong, B. Lo, and G. Z. Yang. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. *IEEE Journal of Biomedical and Health Informatics*, 21(1):56–64, Jan 2017.

[111] Y. P. Raykov, A. Boukouvalas, F. Baig, and M. A. Little. What to do when K-Means clustering fails: A simple yet principled alternative algorithm. *PLOS ONE*, 11(9):1–28, 09 2016.

[112] T. N. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim. Multichannel signal processing with deep neural networks for automatic speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(5):965–979, May 2017.

[113] Y. Saito, S. Takamichi, and H. Saruwatari. Statistical parametric speech synthesis incorporating generative adversarial networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(1):84–96, Jan 2018.

[114] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller. SchNet - a deep learning architecture for molecules and materials. *ArXiv e-prints*, Dec. 2017.

[115] E. Senft, P. Baxter, J. Kennedy, S. Lemaignan, and T. Belpaeme. Supervised autonomy for online learning in human-robot interaction. *Pattern Recognition Letters*, 99(Supplement C):77 – 86, 2017. User Profiling and Behavior Adaptation for Human-Robot Interaction.

[116] A. Severyn and A. Moschitti. Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 373–382, New York, NY, USA, 2015. ACM.

[117] Z. Shao, L. Zhang, and L. Wang. Stacked sparse autoencoder modeling using the synergy of airborne LiDAR and satellite optical and sar data to map forest above-ground biomass. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12):5569–5582, Dec 2017.

[118] R. V. Sharan and T. J. Moir. Robust acoustic event classification using deep neural networks. *Information Sciences*, 396(Supplement C):24 – 32, 2017.

[119] S. Shi, Q. Wang, P. Xu, and X. Chu. Benchmarking State-of-the-Art Deep Learning Software Tools. *ArXiv e-prints*, Aug. 2016.

[120] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.

[121] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *ArXiv e-prints*, Sept. 2016.

[122] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529:484 EP –, Jan 2016. Article.

[123] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354 EP–, Oct 2017. Article.

[124] K. G. Srinivasa, S. Anupindi, R. Sharath, and S. K. Chaitanya. Analysis of facial expressiveness captured in reaction to videos. In *2017 IEEE 7th International Advance Computing Conference (IACC)*, pages 664–670, Jan 2017.

[125] J. A. Stankovic. Research directions for the Internet of Things. *IEEE Internet of Things Journal*, 1(1):3–9, Feb 2014.

[126] B. Su, X. Ding, H. Wang, and Y. Wu. Discriminative dimensionality reduction for multi-dimensional sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(1):77–91, Jan 2018.

[127] Y. Sun, H. Song, A. J. Jara, and R. Bie. Internet of Things and big data analytics for smart and connected communities. *IEEE Access*, 4:766–773, 2016.

[128] J. Tang, D. Sun, S. Liu, and J. L. Gaudiot. Enabling deep learning on iot devices. *Computer*, 50(10):92–96, 2017.

[129] P. Thammasorn, L. Wootton, E. Ford, and M. Nyflot. Deep convolutional triplet network for quantitative medical image analysis with comparative case study of gamma image classification. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1119–1122, Nov 2017.

[130] R. Tibor Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball. Deep learning with convolutional neural networks for decoding and visualization of EEG pathology. *ArXiv e-prints*, Aug. 2017.

[131] ujjwalkarn. An intuitive explanation of convolutional neural networks. 2016. https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[132] S. O. Uwagbole, W. J. Buchanan, and L. Fan. Numerical encoding to tame SQL injection attacks. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pages 1253–1256, April 2016.

[133] G. V, P. L, C. M, and et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2016.

[134] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand. Parking-stall vacancy indicator system, based on deep convolutional neural networks. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 655–660, Dec 2016.

[135] B. Wang, P. Yin, A. L. Bertozzi, P. J. Brantingham, S. J. Osher, and J. Xin. Deep Learning for Real-Time Crime Forecasting and its Ternarization. *ArXiv e-prints*, Nov. 2017.

[136] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck. Deep Learning for Identifying Metastatic Breast Cancer. *ArXiv e-prints*, June 2016.

[137] J. Wang, Q. Gu, J. Wu, G. Liu, and Z. Xiong. Traffic speed prediction and congestion source exploration: A deep learning method. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 499–508, Dec 2016.

[138] Q. Wang, Y. Zheng, G. Yang, W. Jin, X. Chen, and Y. Yin. Multi-scale rotation-invariant convolutional neural networks for lung texture classification. *IEEE Journal of Biomedical and Health Informatics*, 22(1):184–195, Jan 2018.

[139] S. Wang, Y. Shang, J. Wang, L. Mei, and C. Hu. Deep features for person re-identification. In *2015 11th International Conference on Semantics, Knowledge and Grids (SKG)*, pages 244–247, Aug 2015.

[140] Y. Wang, H. Li, and X. Li. Re-architecting the on-chip memory sub-system of machine-learning accelerator for embedded devices. In *Proceedings of the 35th International Conference on Computer-Aided Design*, ICCAD '16, pages 13:1–13:6, New York, NY, USA, 2016. ACM.

[141] Y. Wang, B. Song, P. Zhang, N. Xin, and G. Cao. A fast feature fusion algorithm in image classification for cyber physical systems. *IEEE Access*, 5:9089–9098, 2017.

[142] C. Wu, H. P. Cheng, S. Li, H. Li, and Y. Chen. ApesNet: a pixel-wise efficient segmentation network for embedded devices. *IET Cyber-Physical Systems: Theory Applications*, 1(1):78–85, 2016.

[143] J. Wu and W. Zhao. Design and realization of WInternet: From Net of Things to Internet of Things. *ACM Trans. Cyber-Phys. Syst.*, 1(1):2:1–2:12, Nov. 2016.

[144] H. Xu, J. Lin, and W. Yu. *Smart Transportation Systems: Architecture, Enabling Technologies, and Open Issues*, pages 23–49. Springer Singapore, Singapore, 2017.

[145] Y. Xu, X. Luo, W. Wang, and W. Zhao. Efficient DV-HOP localization for wireless cyber-physical social sensing system: A correntropy-based neural network learning scheme. *Sensors*, 17(135), 2017.

[146] P. C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata. Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robotics and Automation Letters*, 2(2):397–403, April 2017.

[147] Q. Yang, D. An, R. Min, W. Yu, X. Yang, and W. Zhao. Optimal PMU placement based defense against data integrity attacks in smart grid. *IEEE Transactions on Forensics and Information Security (T-IFS)*, 12(7):1735–1750, 2017.

[148] X. Yang, X. Ren, J. Lin, and W. Yu. On binary decomposition based privacy-preserving aggregation schemes in real-time monitoring systems. *IEEE Transactions on Parallel and Distributed Systems*, 27(10):2967–2983, Oct 2016.

[149] X. Yang, T. Wang, X. Ren, and W. Yu. Survey on improving data utility in differentially private sequential data publishing. *IEEE Transactions on Big Data*, PP(99):1–1, 2017.

[150] X. Yang, P. Zhao, X. Zhang, J. Lin, and W. Yu. Toward a gaussian-mixture model-based detection scheme against data integrity attacks in the smart grid. *IEEE Internet of Things Journal*, 4(1):147–161, Feb 2017.

[151] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet of Things Journal*, 4(5):1250–1258, Oct 2017.

[152] C. Yeshwanth, P. S. A. Sooraj, V. Sudhakaran, and V. Raveendran. Estimation of intersection traffic density on decentralized architectures with deep networks. In *2017 International Smart Cities Conference (ISC2)*, pages 1–6, Sept 2017.

[153] W. Yu, D. An, D. Griffith, Q. Yang, and G. Xu. Towards statistical modeling and machine learning based energy usage forecasting in smart grid. *SIGAPP Appl. Comput. Rev.*, 15(1):6–16, Mar. 2015.

[154] W. Yu, L. Ge, G. Xu, and X. Fu. Towards neural network based malware detection on android mobile devices. *Pino R., Kott A., Shevenell M. (eds) Cybersecurity Systems for Human Cognition Augmentation. Advances in Information Security*, 61, 2014.

[155] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang. A survey on the edge computing for the Internet of Things. *IEEE Access*, PP(99):1–1, 2017.

[156] W. Yu, G. Xu, Z. Chen, and P. Moulema. A cloud computing based architecture for cyber security situation awareness. In *2013 IEEE Conference on Communications and Network Security (CNS)*, pages 488–492, Oct 2013.

[157] W. Yu, H. Xu, H. Zhang, D. Griffith, and N. Golmie. Ultra-dense networks: Survey of state of the art and future directions. In *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–10, Aug 2016.

[158] W. Yu, H. Zhang, L. Ge, and R. Hardy. On behavior-based detection of malware on android platform. In *2013 IEEE Global Communications Conference (GLOBECOM)*, pages 814–819, Dec 2013.

[159] X. Yuan, P. He, Q. Zhu, R. Rana Bhat, and X. Li. Adversarial Examples: Attacks and Defenses for Deep Learning. *ArXiv e-prints*, Dec. 2017.

[160] Y. Yuan and K. Jia. A distributed anomaly detection method of operation energy consumption using smart meter data. In *2015 International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP)*, pages 310–313, Sept 2015.

[161] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi. Internet of Things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[162] X. Zhang, X. Li, J. An, L. Gao, B. Hou, and C. Li. Natural language description of remote sensing images based on deep learning. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 4798–4801, July 2017.

[163] X. Zhang, X. Pan, and S. Wang. Fuzzy DBN with rule-based knowledge representation and high interpretability. In *2017 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, pages 1–7, Nov 2017.

[164] X. L. Zhang. Speech separation by cost-sensitive deep learning. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 159–162, Dec 2017.

[165] P. Zhao, W. Yu, D. Quan, and X. Yang. Deep learning-based detection scheme with raw ECG signal for wearable telehealth systems. In *Technical Report 2018-CS-021, Dept. of Computer and Information Science, Towson University*, March 2018.

[166] Y. Zhao, J. Li, and L. Yu. A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, 66:9 – 16, 2017.

[167] Z. Zhao, J. Guo, E. Ding, Z. Zhu, and D. Zhao. Terminal replacement prediction based on deep belief networks. In *2015 International Conference on Network and Information Systems for Computers*, pages 255–258, Jan 2015.

[168] D. Zhu, H. Jin, Y. Yang, D. Wu, and W. Chen. DeepFlow: deep learning-based malware detection by mining Android application for abnormal usage of sensitive data. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 438–443, July 2017.

[169] J. Zhu, Y. Song, D. Jiang, and H. Song. A new deep-q-learning-based transmission scheduling mechanism for the cognitive Internet of Things. *IEEE Internet of Things Journal*, PP(99):1–1, 2017.

[170] M. Zolotukhin, T. Hämäläinen, T. Kokkonen, and J. Siltanen. Increasing web service availability by detecting application-layer DDoS attacks in encrypted traffic. In *2016 23rd International Conference on Telecommunications (ICT)*, pages 1–6, May 2016.