Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

Deep Learning in Mining Biological Data

Mufti Mahmud* · **M. Shamim Kaiser*** ·
T. Martin McGinnity · **Amir Hussain**

Received: 10/08/2020/ Accepted: date

Abstract

Background: Recent technological advancements in data acquisition tools allowed life scientists to acquire multimodal data from different biological application domains. Categorised in three broad types (i.e., images, signals, and sequences), these data are huge in amount and complex in nature. Mining such enormous amount of data for pattern recognition is a big challenge and requires sophisticated data intensive machine learning techniques. Artificial neural network based learning systems are well known for their pattern recognition capabilities and lately their deep architectures - known as deep learning (DL) - have been successfully applied to solve many complex pattern recognition problems.
Methods: To investigate how DL - especially its different architectures - has contributed and utilised in the mining of biological data pertaining to those three types, a meta analysis has been performed and the resulting resources have been critically analysed.

Mufti Mahmud (✉)
Department of Computing & Technology, Nottingham Trent University, Clifton, Nottingham, NG11 8NS, UK E-mail: mufti.mahmud@ntu.ac.uk, muftimahmud@gmail.com

M. Shamim Kaiser (✉)
Institute of Information Technology, Jahangirnagar University, Savar, Dhaka, 1342, Bangladesh E-mail: mskaiser@juniv.edu

T. Martin McGinnity
Intelligent Systems Research Centre, Ulster University, Northern Ireland, Derry, BT48 7JL, UK
Department of Computing & Technology, Nottingham Trent University, Clifton, Nottingham, NG11 8NS, UK

Amir Hussain
School of Computing, Edinburgh Napier University, Edinburgh, EH11 4BN, UK

* M. Mahmud and M.S. Kaiser are joint first and corresponding authors.

Results and Conclusion: Focusing on the use of DL to analyse patterns in data from diverse biological domains, this work investigates different DL architectures' applications to these data. This is followed by an exploration of available open access data sources pertaining to the three data types along with popular open source DL tools applicable to these data. Also, comparative investigations of these tools from qualitative, quantitative, and benchmarking perspectives are provided. Finally, some open research challenges in using DL to mine biological data are outlined and a number of possible future perspectives are put forward.

Keywords Brain machine interfaces · bioimaging · deep learning performance comparison · medical imaging · omics · open access data sources · open source tools.

1 Introduction

The pursuit of understanding human behaviours, along with the various pathologies, their early diagnosis and finding cures have driven the life sciences research in the last two centuries [1]. This accelerated the development of cutting edge tools and technologies that allow scientists to study holistically the biological systems as well as dig down, in an unprecedented manner, to the molecular details of the living organisms [2,3]. Increasing technological sophistication has presented scientists with novel tools for DNA sequencing [4], gene expression [5], bioimaging [6], neuroimaging [7], and body/brain-machine interfaces [8].

These innovative approaches to study the living organisms produce huge amount of data [9] and create a situation often referred as 'Data Deluge' [10]. Depending on the target application and experimentation, this

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

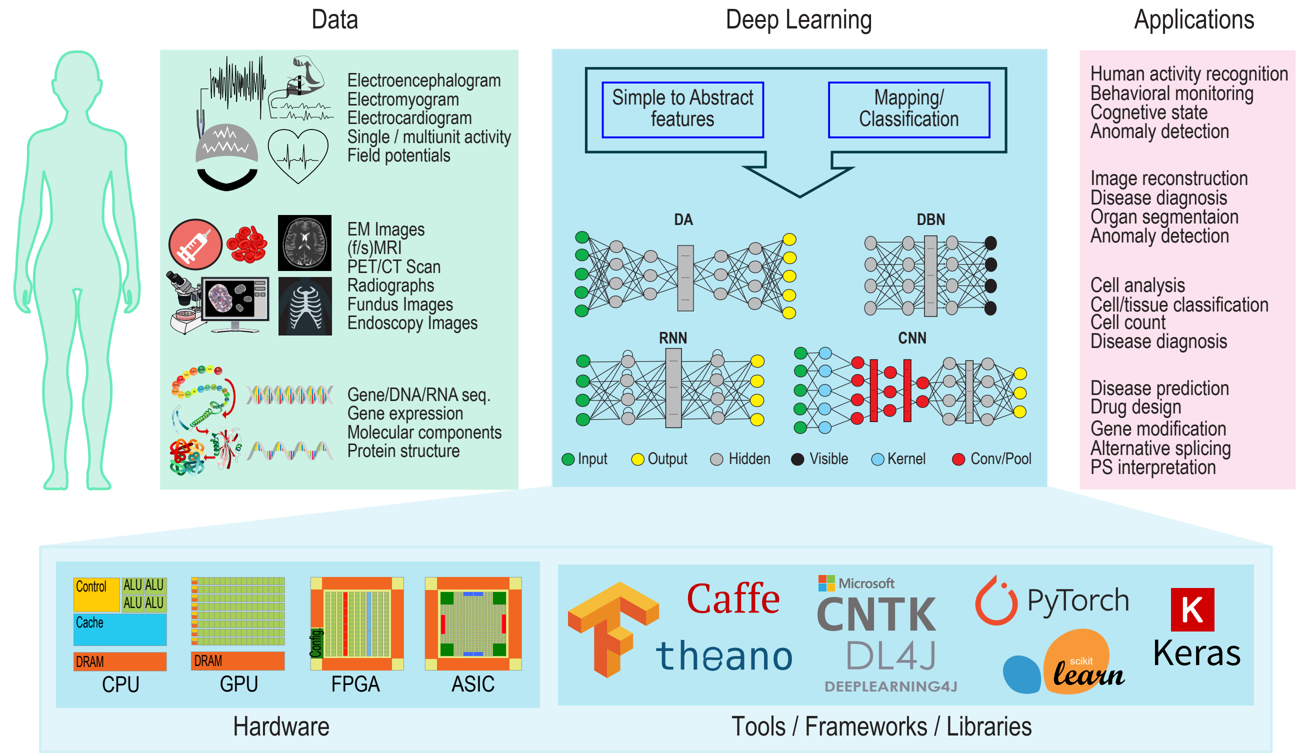


Fig. 1 The ecosystem of modern data analytics using advanced machine learning methods with specific focus on application of DL to biological data mining. The biological data coming from various sources (e.g., sequence data from the *Omics*, various images from the *[Medical/Bio]-Imaging*, and signals from the *[Brain/Body]-Machine Interfaces*) are mined using DL with suitable architectures tailored for specific applications.

biological big data can be characterized by their inherent characteristics of being *hierarchical* (i.e., data coming from different levels of a biological system – from molecules to cells to tissues to systems), *heterogeneous* (i.e., data acquired by different acquisition methods – from genetics to physiology to pathology to imaging), *dynamic* (i.e., data changes as a function of time), and *complex* (i.e., data describing nonlinear biological processes) [11]. These intrinsic characteristics of biological big data posed an enormous challenge to data scientists to identify patterns and analyze them to infer meaningful conclusions from these data [12]. The challenges have triggered the development of rational, reliable, reusable, rigorous, and robust software tools [11] using machine learning (ML) based methods to facilitate recognition, classification, and prediction of patterns in the biological big data [13].

Based on how a method learns from the data, the ML techniques can be broadly categorized into *supervised* and *unsupervised* approaches. In *supervised* learning, objects in a pool are classified using a set of known annotations or attributes or features, i.e., a *supervised* algorithm learns the pattern(s) from a limited number of annotated training data and then classifies the remaining testing data using the acquired knowledge. Instead, in the *unsupervised* learning, pattern(s) are first defined from a subset of the unknown data and then the remaining data are classified based on the defined patterns, i.e., an *unsupervised* algorithm first defines pattern(s) among the objects in a pool of data with unknown annotations or attributes or features, and then uses the acquired knowledge to classify the remaining data. In addition, there is another category called *reinforcement* learning which, is out of the scope of this work, but allows an agent to improve its experience and knowledge by learning iteratively through interacting with its environment.

Since the 1950s' many methods pertaining to both the learning paradigms (i.e., *supervised* and *unsupervised*) have been proposed. The popular methods in the *supervised* domain include: ANN [14] and its variants (e.g., Backpropagation [15], Hopfield Networks [16], Boltzmann Machines [17], Restricted Boltzmann Machines [18], Spiking Neural Networks [19], etc.), Bayesian Statistics [20], Support Vector Machines [21] and other linear classifiers [22] (e.g., Fisher's Linear Discriminant [23], Regressors [24], Naive Bayes Classifier [25], etc.), k-Nearest Neighbors [26], Hidden Markov Model [27], and Decision Trees [28]. Popular *unsupervised* methods include: Autoencoders [29], Expectation-Maximization

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

[30], Information Bottleneck [31], Self-Organizing Maps [32], Association Rules [33], Hierarchical Clustering [34], k-Means [35], Fuzzy Clustering [36], and Density-based Clustering [37,38] (e.g., Ordering Points To Identify the Clustering Structure [39]). Many of these methods have been successfully applied to data coming from various biological sources.

For the sake of simplicity, the vast amount of biological data coming from the diverse application domains have been categorized to a few broad data types. These data types include, *Sequences* (data generated by Omics technologies, e.g., [gen/ transcript/ epigen/ prote/ metabol]omics [40]), *Images* (data generated by [bio/ medical/ clinical/ health]-imaging techniques containing [sub-]cellular and diagnostic images), and *Signals* (electrical signals generated by the brain and the muscles and acquired using appropriate sensors at the [Brain/Body]-Machine Interfaces or BMI). Each of these data types originating at diverse biological application domains have witnessed major contributions from the specified ML methods and their variants (see for *Sequences* [41], *images* [42,43,44], and *signals* [45,46,47]).

In recent years DL methods are potentially reshaping the future of ML and AI [48]. Worthy to mention here that, from a broader perspective, ML has been applied to a range of tasks including anomaly detection [49,50], biological data mining [51,52], detection of Corona virus [53,54], brain disease detection [55,56, 57], education [58], natural language processing [59], and price prediction [60]. Despite notable popularity and applicability to diverse disciplines [61], there exists no comprehensive review which focuses on pattern recognition in biological data, provides pointers to the various biological data sources and DL tools, and the performances of those tools [51].

Also, considering the ecosystem of modern data analysis using advanced ML techniques (such as DL), providing information about methods' application only partially covers the components of this ecosystem (see the various components of the ecosystem in Fig. 1). The remaining components of the ecosystem include open access data sources and open source toolboxes and libraries which are used in developing the individual methods. It is therefore of paramount importance to have a complete understanding of the availability of datasets and their characteristics, the capabilities and options offered by the libraries and how they compare with each other in different execution environments such as central processing unit (CPU) and graphical processing unit (GPU). The current paper's novelty lies in being first of its kind to cover comprehensively the complete ecosystem of modern data analysis using advanced ML technique, i.e., DL.

Therefore, with the above aim, this review provides– a brief overview on DL concepts and their applications to various biological data types; a list of available open access data repositories offering data for method development; and a list of existing open source libraries and frameworks which can be utilized to harness the power of these techniques along with their relative and performance comparison. Towards the end, some open issues are identified and some speculative future perspectives are outlined.

The remainder of the article is organized as follows: section 2 provides the conceptual overview and introduces the reader to the underlying theory of DL; section 3 describes the applications; section 4 lists the open source data repositories; section 5 presents the popular open source DL tools; sections 6 and 7 compares the most popular tools from relative and performance perspectives. Section 8 presents the reader with some of the open issues and hints on the future perspectives; and finally, the article is concluded in section 9.

2 Overview of Deep Learning

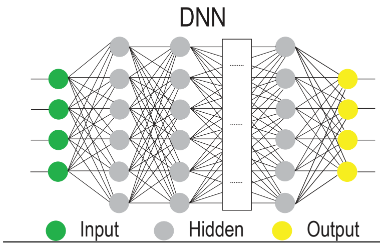
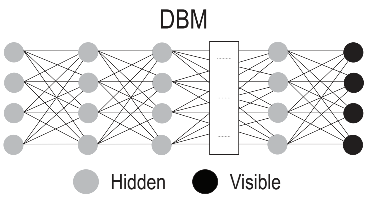
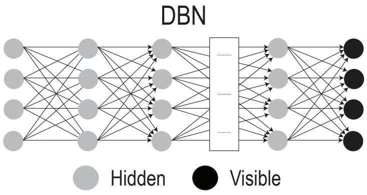
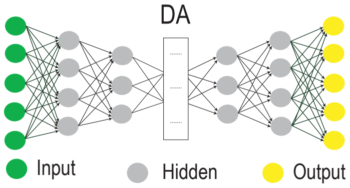
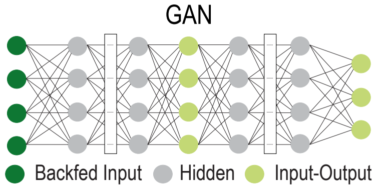
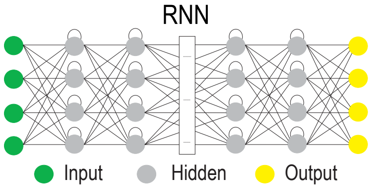
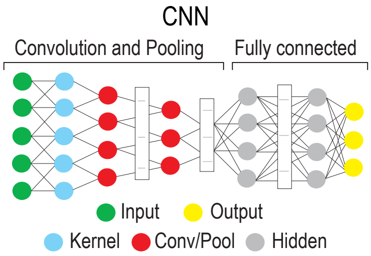
In DL the data representations are learned with increasing abstraction levels, i.e., at each level more abstract representations are learned by defining them in terms of less abstract representations at lower levels [62]. Through this hierarchical learning process a system can learn complex representations directly from the raw data [63].

Though many DL architectures have been proposed in the literature for various applications, there has been a consistent preference to use particular variants for biological data. As shown in Fig. 2, the most popular models have been identified as– Deep Neural Network (DNN), Deep Boltzmaan Machine (DBM) and Deep Belief Network (DBN), Deep Autoencoder (DA), Generative Adversarial Network (GAN), Recurrent Neural Network (RNN, including LSTM), and Convolutional Neural Network (CNN). Each of these models' architectures and their respective pros and cons are listed in Table 1. Therefore, the following subsections introduces the reader to each of these most frequently used DL architectures in mining biological data.

2.1 Deep Neural Network (DNN)

A DNN [64] is inspired by the brain's multilevel visual processing mechanism starting with the cortical area 'V1' and then to area 'V2', and so on [65]. Mimicking this, the traditional artificial neural network or NN is

Table 1 Keypoints and applications of different deep learning architectures

Architecture	Pros.	Cons.
	– DNN can learn high level feature representation and apply transfer learning. – It can be used for healthcare and visual recognition.	– It requires very substantial volume of training data. – Significant computational power is required. – The learning process is slow.
	– Graphical model, undirected links across a set of visible nodes and a set of hidden nodes. – Used mainly for dimensionality reduction and classification.	– High time complexity for inference than DBN – Learning information does not reach to the lower layer – Tends to do overfitting
	– Easy to code and work sufficiently well for just a few layers – Higher performance gain by adding layers compared to Multilayer perceptron. – Robustness in classification.	– It can be trained greedily, one layer at a time. – Hard to deduce posterior distribution for configurations of hidden causes.
	– Learn data encoding, reconstruction and generation at same time. – Training is stable without label data. – Variant: Sparse, Denoising and Contractive DA.	– Requires pre-training stage due to the chances of vanishing error. – Each application requires redesigned and retrained the model. – The DA can more sensitive to input errors.
	– The main benefit is the data augmentation. – GANs are an unsupervised learning method. – GANs learn density distributions of data.	– Hard to train as optimizing loss function is hard and requires a lot of trial-and-errors.
	– It can process inputs of any length. – RNN can use internal memory and performs well for stream time series data.	– Computation is slow and training a model can be difficult. – It becomes very difficult to process sequences that are very long. – Prone to problems such as exploding and gradient vanishing.
 CNN Convolution and Pooling Fully connected Input Output Kernel Conv/Pool Hidden	– CNN can capture hierarchical information. – Open source resources are available. – CNN can share pre-trained weight which is required for transfer learning – Requires less neuron connections compared to DNN	– Larger labelled dataset is required for training. – The mechanism of CNN is not clear.

Legend: DA: Deep Autoencoder; DBN: Deep Belief Network; RNN: Recurrent Neural Network; DNN: Deep Neural Network; DBM: Deep Boltzmann Machine; CNN: Convolutonal Neural Network.

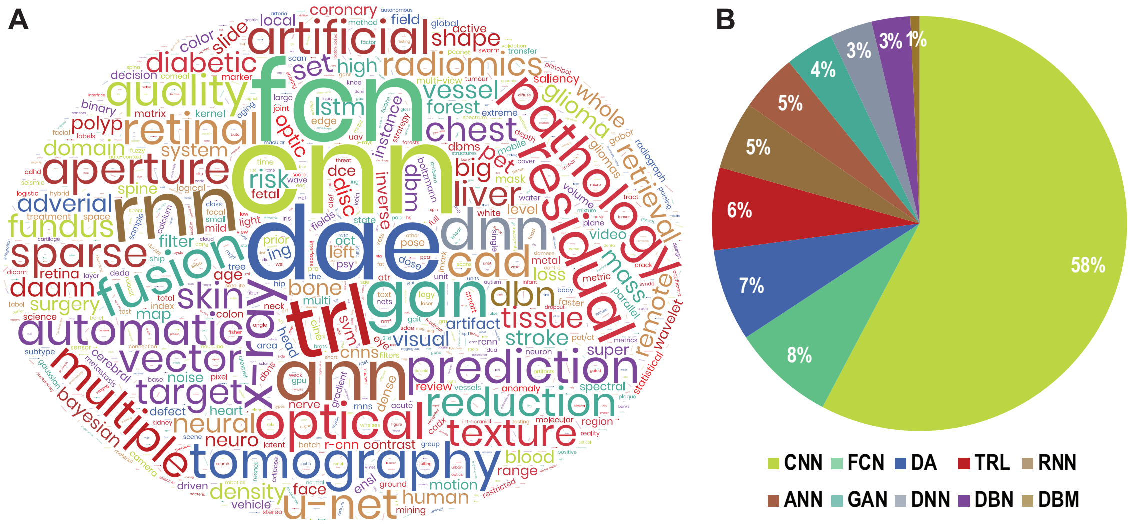


Fig. 2 Application of different DL models to biological data. (A) Wordcloud generated using author keywords extracted from research papers published between January 2011 and March 2020 which mentioned analysis of biological data (images, signals and sequences) using DL techniques and indexed in the Scopus database. The keywords were pruned to highlight the analysis methods. (B) Distribution of published papers mentioning the usage of top 10 techniques. The colours of the individual pies match the colours in the wordcloud. Legend— CNN: Convolutional Neural Network, FCN: Fully Connected Network, DA[E]: Deep Autoencoder, TRL: Transfer Learning, RNN: Recurrent Neural Network (including Long-Short Term Memory or LSTM), ANN: Artificial Neural Network, GAN: Generative Adversarial Network, DNN: Deep Neural Network, DBN: Deep Belief Network, DBM: Deep Boltzmaan Machine.

extended with additional hidden layers containing nonlinear computational units in each of these hidden layers to learn a subset of the given representations. Despite its successful usage in a range of different applications, the main drawback has been the slow and cumbersome training process [66].

2.2 [Restricted] Boltzmann Machines ([R]BM)

[R]BM represents specific probability distributions through a undirected probabilistic generative model [67]. Considered as a nonlinear feature detector, [R]BM is trained based on optimising its parameters for a set of given observations to obtain the best possible fit of the probability distribution through a Markov Chain Monte Carlo method known as Gibbs sampling [68,69]. With symmetrical connections among subsequent units in multiple hidden layers, BM has only one visible layer. The main drawback of the standard BM is that, the learning process is computationally expensive and quite slow. Due to this a BM requires a long period to reach equilibrium statistics [62]. However, this learning inefficiency can be solved by forming a bipartite graph (i.e., restricting to have one hidden layer and one visible layer) [67]. To extend this shallow architecture to a deep one, multiple RBMs as unitary learning elements are stacked together and this yields the following two DL architectures.

2.2.1 Deep Boltzmann Machine (DBM)

DBM [70] is a stack of undirected RBMs which supports a feedback mechanism among the layers to facilitate inference from higher level units to propagate to lower level units. This allows an input to be alternatively interpreted through concurrent competition at all levels of the model. Despite this powerful inference mechanism, estimating model parameters from data remains a challenge and cannot be solved using traditional gradient based methods (e.g., persistent contrastive divergence [71]) [70]. Though this learning problem is overcome by pretraining each RBM in a layerwise greedy fashion, with outputs of the hidden variables from lower layers as input to upper layers [67], the time complexity remains high and the approach may not be suitable for large training datasets [72].

2.2.2 Deep Belief Network (DBN)

DBN [73], in contrast to the DBM, is formed by stacking several RBMs together in a way that one RBM's latent layer is linked to the next RBM's visible layer. As the top two layers of DBN are undirected, the connections are downward directed to its immediate lower layer [73,74]. Thus, the DBN is a hybrid model with the first two layers as a undirected graphical model and the rest being directed generative model. The different layers are learned in a layerwise greedy fashion and fine-tuned based on required output [75], however, the training procedure is computationally demanding.

2.3 Deep Autoencoder (DA)

DA is a DL architecture [76] obtained by stacking a number of data driven Autoencoders which are unsupervised elements. DA is also known as DAE, and is designed to reduce data dimension by automatically projecting incoming representations to a lower dimensional space than that of the input. In an Autoencoder, equal amounts of units are used in the input/output layers and less units in the hidden layers. (Non)linear transformations are embodied in the hidden layer units to encode the given input into smaller dimensions [77]. Despite the fact that it requires a pre-training stage and suffers from a vanishing error, this architecture is popular for its data compression capability and has many variants, e.g., Denoising Autoencoder [76], Sparse Autoencoder [78], Variational Autoencoder [79], and Contractive Autoencoder [80].

2.4 Generative Adversarial Network (GAN)

GAN [81] is an effective generative model. Generative models perform an unsupervised learning task, where they automatically discover and learn existing patterns in data and then use that knowledge to generate new examples of the learnt pattern as if they were drawn from the original dataset. Using GAN, the problem is seen as a supervised learning problem with two strands– (i) the generator, which generates new examples as trained, and (ii) the discriminator, which classifies generated examples to two classes (real or fake). These generator and discriminator models are trained together in a zero-sum game (i.e., in an adversarial fashion) such that the examples generated by the generator model maximise the loss of the discriminator model [82,83].

2.5 Recurrent Neural Network (RNN)

The RNN architecture [84] is designed to detect spatio-temporal alignments in streams of data [85]. Unlike feedforward NN which performs computations unidirectionally from input to output, an RNN computes the current state's output depending on the outputs of the previous states. Due to this 'memory'-like property, despite learning problems related to vanishing and exploding gradients, RNN has gained popularity in many fields involving streaming data (e.g., text mining, time series, genomes, financial etc.). In recent years, two main variants, bidirectional RNN (BRNN) [86] and long short-term memory (LSTM) [87] have also been applied [48,88,89].

2.6 Convolutional Neural Network (CNN)

CNN [90] is a multilayer NN model [91] which has gained popularity in analysing image based data. Inspired by the neurobiology of the visual cortex, the CNN consists of convolutional layer(s) containing a set of learnable filter banks and followed by fully connected layer(s). These filter banks convolve with the input data and pass the results to activation functions (e.g., ReLU, Sigmoid, and Tanh). There also exist subsampling steps in between these layers. The CNN outperforms DNNs, which as they do not scale well with multidimensional locally correlated input data. To address the scaling problem of DNNs, the CNN approach has been quite successful in analysing datasets with a high number of nodes and parameters (e.g., images). As the images are 'stationary,' convolution filters (CF) can easily learn data-driven kernels. Applying such CF along with a suitable pooling function reduces the features that are supplied to the fully connected network to classify. However, in case of large datasets even this can be daunting and can be solved using sparsely connected networks. Some of the popular CNN configurations include: AlexNet [92], VGGNet [93] GoogLeNet [94] etc. (see Table 2 for a complete list of CNN's variations with relevant details).

3 Deep Learning and Biological Data

Many studies have been reported in the literature which employ diverse DL architectures with related and varied parameter sets (see section 2) to analyze patterns in biological data. For most of the DL architectures, as shown in Fig. 3, the number of publications are increasing steadily over the years. A set of randomly selected representative studies from the large amount of reported literature are described below and summarised in Table 3. These studies belong to the three data types we have considered within the context of this paper, that is, images, signals and sequences.

3.1 Images

CNN was used by on histology images of the breast to find mitosis [108,142] and to segment neuronal structures in Electron Microscope Images (EMI) [103]. Havaei

Table 2 Keypoints of different deep CNN architectures

Architecture	Network Design	Parameters	Key points
LeNet (1998)	LeNet-5 is first CNN architecture with two convolution and three fully connected layers. parameters.	0.06 Millions	– Feed-forward NN – Connection between layers are Sparsed to reduce the complexity of computational
AlexNet (2012)	AlexNet has 8 layers consists of 5 convolutional and 3 fully-connected layers.	0.6 Millions	Deeper than the LeNet and Aliasing artifacts in the learned feature-maps due to large filter size.
VGG-16 (2014)	VGG-16 has 13 convolutional layers (+ max pooling) and 2 fully connected layers followed by 1 output layer with softmax activation. while padding was performed to maintain the spatial resolution	138 Millions	– Roughly twice deeper network can be designed compared to the AlexNet – A deeper variant of VGG is VGG-19. – Computationally expensive and can not be used with low resource systems
Inception-v1 (2014)	It is also called GoogleNet which includes multi-scale convolutional transformations using split, transform and merge concept. It has 22 layers (27 layers including the pooling layers). At the end of the architecture, it employes global average pooling at the last layer instead of using a fully connected layer to reduce connection's density.	4 Millions	– It uses sparse connections to overcome redundant information problem and omits irrelevant feature maps. – high accuracy with a reduced computational cost – drawback due to heterogeneous topology that needs to be customized from module to module
Inception-v3 (2015)	In Inception-V3, 1×1 convolutional operation was used, which maps the input data into 3 or 4 separate spaces that are smaller than the original input space, and then maps all correlations in these smaller 3D spaces, via regular (3×3 or 5×5) convolutions.	23.6 Millions	– increased accuracy and reduced the computational complexity than Inception-v1. – Reduce representational bottleneck. – Replace large size filters with small filters – Complex architecture design and Lack of homogeneity
ResNet-50 (2015)	ResNeXt (Aggregated Residual Transform Network) is an improvement over the Inception Network by including cardinality (i.e., the size of the set of transformations) along with split-transform-merge. It consists of 5 stages: each stage contains a convolution (3 convolution layers) and a identity block (3 convolution layers).	23 Millions	– It accelerates the training speed. – Reducing the effect of Vanishing Gradient Problem. – High accuracy performance in image classification.
Xception (2016)	The Xception architecture has 36 convolutional layers forming the feature extraction base of the network. The 36 convolutional layers are structured into 14 modules, all of which have linear residual connections around them, except for the first and last modules.	22.8 Millions	– It is an open-source implementation using Keras and TensorFlow under the MIT license. – Compared to Inception V3, Xception shows small gains in classification performance on the ImageNet dataset and large gains on the JFT dataset.
Inception-v4 (2016)	Added stem group and more more inception blocks than Inception-v3	43Million	Deep hierarchies of features, multilevel feature representation. Learning speed is slow.
Inception-ResNet-V2 (2016)	Adding more Inception modules and converting Inception modules to Residual Inception blocks. Also, add a new type of Inception module (Inception-A) after the Stem module.	56 Millions	It improves training speed. Deep hierarchies of features, multilevel feature representation.
ResNeXt (2017)	It Scales up the number of parallel towers ("cardinality") within a module	68.1 Millions	Homogeneous topology;Grouped convolution.
DenseNet (2017)	DenseNet connected each preceding layer to the next coming layer in a feed-forward fashion; thus, feature-maps of all previous layers were used as inputs into all subsequent layers	25.6 M	Introduced depth or cross-layer dimension. Ensures maximum data flow between the layers in the network. Avoid relearning of redundant feature-mapsn

et al. used CNN to segment brain tumor from MRI [100] and Hosseini et al. used it for the diagnosis of AD from MRI [97,56]. DBM [98] and RBM [99] were used in detecting Alzheimer's Disease (AD) and Mild Cognitive Impairment (MCI) from MRI and PET scans. Again, CNN was used on MRI to detect neuroendocrine carcinoma [105,55,74]. CNN's dual pathway version was used by Kamnitsas et al. to segment lesions related to tumors, traumatic injuries, and ischemic strokes [109]. CNN was also used by Fritscher et al. for volume segmentation [101] and by Cho et al. to find anatomical structures (Lung nodule to classify malignancy) [106] from CT scans. DBN was applied on MRIs to detect Attention Deficit Hyperactivity Disorder [96] and on cardiac MRIs to segment the heart's left ventricle [107]. GANs have gained popularity in image synthesis and data augmentation to reduce overfitting. GAN's application in data augmentation and image translation has been reviewed in [143] and data augmentation in the CT segmentation tasks was done using CycleGAN [144]. GAN-based framework called MedGAN was proposed for medical image-to-image translation [145]. GAN was used as survival prediction model for chest CT scan images of patients suffering from idiopathic pulmonary fibrosis [146,147]. GAN was also used by Halicek for

Table 3 Deep learning applied to biological data

Type	Data [base/set]	DL Architecture	Task
Images	ABIDE	DNN [95]	Autism disorder identification
	ADHD-200 dataset	DBN [96]	ADHD detection
	ADNI dataset	CNN [97], DBM [98], DBN [99]	AD/MCI diagnosis
	BRATS Dataset	CNN [100]	Brain pathology segmentation
	CT dataset	CNN [101]	Fast segmentation of 3D medical images
	DRIVE, STARE datasets	GAN [102]	Retinal blood vessel segmentation
	EM segmentation challenge dataset	CNN [103]	Segment neuronal membranes
		LSTM [104]	Biomedical volumetric image segmentation
	IBSR, LPBA40 & OASIS dataset	CNN [105]	Skull stripping
	LIDC-IDRI dataset	CNN [106]	Lung nodule malignancy classification
	MICCAI 2009 LV dataset	DBN [107]	Heart LV segmentation
	MITOS dataset	CNN [108]	Mitosis detection in breast cancer
	PACS dataset	CNN [106]	Medical image classification
	TBI dataset	CNN [109]	Brain lesion segmentation
Signals	BCI Competition IV	DBN [110], CNN [111, 112, 113]	Motion action decoding
	DEAP dataset	DBN [114, 115]	Affective state recognition
		CNN [116]	Emotion classification
	DECAF	GAN [117]	
	Freiburg dataset	CNN [118]	Seizure prediction
	MAHNOB-HCI	DA [119]	Emotion recognition
	MIT-BIH arrhythmia database	DBN [120, 121]	ECG Arrhythmia classification
	MIT-BIH, INCART, & SVDB	CNN [122]	Movement decoding
	Ninapro database	DBN [123], CNN [122]	Motion action decoding
Sequences	CullPDB, CB513, CASP datasets, CAMEO	CNN [124]	2ps prediction
	DREAM	CNN [125]	DNA/RNA sequence prediction
		DNN [126]	Predict effective drug combination
	ENCODE database	CNN [127, 128]	Gene expression identification
	ENCODE DGF dataset	CNN [129]	Predict noncoding-variant of Gene
	GEO database	GAN [130]	Gene expression data augmentation
	GWH & UCSC datasets	DBN [131]	Splice junctions prediction
	JASPAR database & ENCODE	CNN [132]	Predicting DNA–protein binding
	miRBoost	RNN [133]	micro-RNA Prediction
	miRNA-mRNA pairing data repository	LSTM [134]	micro-RNA target prediction
	Protein Data Bank (PDB)	DA [135]	Protein structure reconstruction
	SRBCT, Prostate Tumor, and MLL GE	DBN [136]	Gene/MiRNA feature selection
	sbv IMPROVER	DBN [137]	Human diseases & drug development
	TCGA database	DA [138]	Cancer detection & gene identification
		DBM [139]	
		DNN [140]	Drug combination estimation
	UCSC, CGHV Data, SPIDEX database	CNN [141]	Genetic variants identification

synthesizing hyperspectral images from digitized histology of breast cancer cells [148].

3.2 Signals

A stacked DA was employed to detect emotion from EEG signals after extracting relevant features using PCA and reducing nonstationary effect using covariate shift adaptation [119]. DBN was applied to decode motor imagery through classifying EEG signal [110]. For

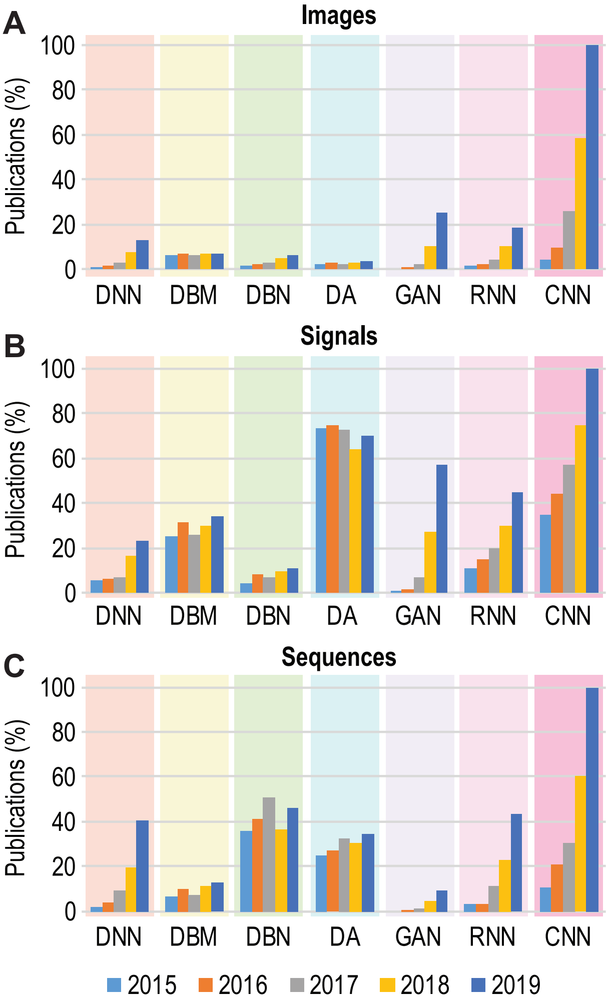


Fig. 3 Trends in publication involving different DL architectures from 2015 to 2019 in three major types of data – images (A), signals (B), and sequences (C). The number of papers have been normalised within each data type. However, it is noteworthy that the ratio of number of publications involving DL techniques applied to different data types (images, signals, and sequences) are approximately – $1{:}\frac{1}{4}{:}\frac{1}{10}$.

a similar purpose CNN was used with augmented common spatial pattern features [111]. EEG signals were also classified using DA after features such as location, time, and frequency were extracted using CNN [112]. Li et al. used DBN to extract low dimensional latent features, and select critical channels to classify affective state using EEG signals [114]. Also, Jia et al. used an active learning to train DBN and generative RBMs for the classification [115]. Tripathi et al. utilized DNN and CNN based model for emotion classification [116]. CNN was employed to predict seizures through synchronization patterns classification [118]. DBN [123] and CNN [122] were used to decode motion action from Ninapro database. The later approach was also used on MIT-BIH, INCART, & SVDB repositories [122]. Moreover, the ECG Arrhythmias were classified using DBN [120, 121] from the data supplied by MIT-BIH arrhythmia database. Zhu et al. used a GAN model with LSTM and CNN to generate ECG signals with high morphological similarity [149]. Another GAN model, RPSeqGAN, trained with SeqGAN [150] generated arrhythmic ECG data with five periods and showed high stability and data quality [151]. GAN is also used by Luo and Lu for EEG data augmentation [152]. You et al. [153] and Jiao et al. [154] utilized GAN-based model for detecting seizure using EEG signal and Driver sleepiness using EEG and EOG signals, respectively. Singh et al. proposed a new GAN framework for denoising ECG [155].

3.3 Sequences

The Stacked Denoising DA has been used to extract features for cancer diagnosis and classification along with the identification of related genes from gene expression (GE) data [138]. GAN was also used for identifying expression patterns from GE data [156]. A template-based DA learning model was used in reconstructing the protein structures [135]. Lee et al. applied a DBN based unsupervised method to perform auto-prediction of splicing junction at DNA level [131]. Combining DBN with active learning, Ibrahim et al. devised a method to select feature groups from genes or microRNAs (miRNAs) based on expression profiles [136]. For translational research, bimodal DBNs were used by Chen et al. to predict responses of human cells using model organisms [137]. Pan et al. applied a hybrid CNN-DBN model on RNAs for the prediction of RNA binding protein (RBP) interaction sites and motifs [157], and Alipanahi et al. used CNN to predict sequence specificities of [D/R]BPs [125]. Denas and Taylor used CNN to preprocess ChIP-seq data and created gene transcription factor activity profiles [127]. CNN was used by Kelley et al. to predict DNA sequence accessibility [128], by Zeng et al. to predict the DBP [132], by Zhou et al. [129] and Huang et al.[141] to find noncoding gene variation, and by Wang et al. to predict secondary protein structure (2ps) [124]. Park et al. used LSTM to predict miRNA precursor [133] and Lee et al. [134] used it to predict miRNA precursors' targets. GAN was used by Marouf et al. for the realistic generation of single-cell RNA-seq data [130], by Jiang et al. to predict disease gene from RNA-seq data [158], by Zhao et al. as a semi-supervised procedure for predicting drug target binding [159], and by Wang et al. for identifying expression patterns from GE data [156].

4 Open Access Biological Data Sources

Reproducing scientific results, reported as statistically processed quantitative data or carefully selected representative qualitative data, has been facilitated greatly by data sharing initiatives [160]. In the last few decades many open access data repositories have been made available for this purpose [161]. Indeed many research funders and journals now require data used for studies to be made openly available for verification. To facilitate method development, here we list the leading and popular open access data repositories pertaining to the Sequences, Images, and Signals data which are summarized in Tables 4, 5 and 6, respectively.

4.1 Images

Table 4 lists the leading open access data sources including databases and individual datasets that provide access to data pertaining to biological image research. For the sake of simplicity, we have grouped these sources to four broad application areas - [Bio/ Medical]-image processing & analysis, disease detection and diagnosis, neuroimage processing & analysis, and segmentation - and are briefly described below.

4.1.1 Bio/Medical]-image Processing and Analysis

The Cell Centered Database (CCDB) [162] collection provides high resolution 3-D light and electron microscopic reconstructions of cells and subcellular structures. It also contains [2/3/4]-D protein distribution and structural information from a number of different microscopic image acquisition systems.

Another image library, called the Cell Image Library (CIL) [163], presents more than 10,000 unique datasets and 20 TB of images, videos, and animations data. These data belong to a wide diversity of organisms, cell types, and cellular processes.

The Euro Bioimaging [164] database provides biological and biomedical imaging data aiming to provide collaboration among different stakeholders including scientists, industry, national and European authorities. Its mission is to give access and services to state-of-the-art imaging techniques and bioimaging data for scientists in Europe and beyond. Euro Bioimaging also includes image analysis tools.

The HAPS is a histology image database [165] contains medium/high resolution photograph of microscopic image of human cells and tissues which are free of any copyright. Another image database, the Image Data Resource (IDR) [166], contains individual datasets of cellular and tissue images. The various categories of images include time-lapse imaging, protein localization studies, digital pathology imaging, yeast study, human high-content screening, etc. It also public API which facilitates viewing, analysis, and sharing of multi-D image data for cell biology.

The SICAS Medical Image Repository (SMIR) is an image repository for medical research purpose. Two of their featured collections include post mortem Full Body CT [167] scan of 50 anonymised subjects of different age group and gender, and CT, microCT, segmentation and shape models of the cochlea [183].

The Cancer Imaging Archive (TCIA) [168] contains CT, MRI, and nuclear medicine (e.g. PET) images for clinical diagnostic, biomarker and cross-disciplinary investigation. The Stanford Tissue Microarray Database (TMA) [169] is a source for annotated microscopic tissue images and associated expression data. The data can be used for studying cell biology. The UCSB biosegmentation benchmark dataset [170] contains 2/3-D cellular, subcellular and tissue images. These datasets can be used for segmentation and classification task.

4.1.2 Disease Detection and Diagnosis

A large amount of imaging data has been acquired from patients with neurological disorders. The Autism Brain Imaging Data Exchange (ABIDE) [171] database, it includes autism brain imaging datasets for studying the autism spectrum disorder. The other dataset pertains to the Attention Deficit Hyperactivity Disorder (ADHD) [172] and includes 776 resting-state fMRI and anatomical datasets which are fused over the 8 independent imaging sites. The phenotypic information includes: age, sex, diagnostic status, measured ADHD symptom, intelligence quotient and medication status. Imaging-based diagnostic classification is the main aim of the ADHD 200 dataset. The ADNI (Alzheimer's Disease Neuroimaging Initiative [173]) is a popular database and contains neuroimaging datasets from neurodegenerative diseases, in particular, Alzheimer's Disease (AD), mild cognitive impairment, early AD and elderly control subjects. The datasets offered by this repository is mainly dedicated for development of novel methods for diseases related to AD. Another dataset focusing on AD is the Open Access Series of Imaging Studies (OASIS) [181] dataset. This contains MRI datasets and open source data management platform (XNAT) to study and analyse AD. Neurosynth [179] is yet another database which includes fMRI literature (with some datasets) and synthesis platform to study Brain structure, functions and disease. On the otherhand, the Open Neuroimaging (Open NI) [182] dataset contains imaging Modalities and brain diseases data which can

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

Table 4 Application-wise categorisation of open access data repositories and datasets pertaining to [bio/medical/health/clinical]-images

Application	Name	Description	Ref.
[Bio/Medical]-image processing & analysis	CCDB	High resolution [2/3/4]-D light and electron microscope images	[162]
	CIL	Cell image datasets and cell library app.	[163]
	Euro Bioimaging	Biological and biomedical imaging data	[164]
	HAPS	Microscopic image of human cells and tissues	[165]
	IDR	Viewing, analysis, and sharing of multi-D image data	[166]
	SMIR	Post mortem CT scans of the whole body	[167]
	TCIA	CT, MRI, and PET images of cancer patients	[168]
	TMA	Microscopic tissue images of human	[169]
	UCSB BioSeg	2D/3D cellular, subcellular and tissue images	[170]
Disease detection and diagnosis	ABIDE	Autism brain imaging datasets	[171]
	ADHD-200	fMRI/anatomical datasets fused over the 8 imaging sites	[172]
	ADNI	MCI, early AD & elderly control subjects' diagnosis data	[173]
	BCDR	Multimodal mammography and ultrasound scan data	[174]
	Kaggle CXRayP	Chest X-ray scans for pneumonia	[175]
	MITOS	Breast cancer histological images	[176]
	NAMIC	Lupus, Brain, Prostate MRI scans	[177]
	nCOV-CXray	COVID-19 cases with chest X-ray/CT images	[178]
	Neurosynth	fMRI datasets and synthesis platform	[179]
	NIH	Labelled chest x-ray images with diagnoses	[180]
	OASIS	MRI datasets and XNAT data management platform	[181]
	Open NI	Imaging Modalities and brain diseases data	[182]
	SMIR	CT of Human temporal bones	[183]
Neuroimage processing & analysis	IXI	It provides neuroimaging data and toolkit software	[184]
	LPBA40	Maps of brain regions and a set of whole-head MRI	[185]
	NeuroVault.org	API for collecting and sharing statistical maps of brain	[186]
	NITRC	MRI, PET, SPECT, CT, MEG/EEG and optical imaging	[187]
	OpenfMRI	Multimodal MRI &EEG datasets	[188]
	UK data service	fMRI dataset	[189]
Segmentation	DRIVE	Digital Retinal Images diabetic patient	[190]
	IBSR	Segmentation results of MRI data	[191]
	STARE	The dataset contains raw/labelled retinal images	[192]

Legend: CXRayP–Chest X-Ray Pneumonia; JHDTI–Johns Hopkins Diffusion Tensor Imaging;

be used to study decision support system for disease identification.

The recent COVID-19 pandemic has attracted a number of researchers to focus their attention is the detection of the novel corona virus disease. The NIH [180] nCOV Chest Xray database [178] contains COVID-19 cases with chest X-ray/CT images. The data can be used for identifying Bacterial vs Viral vs COVID-19 Pneumonia. Similar Chest Xray datasets [175] are hosted by Kaggle which include chest X-ray scans data for detecting traditional viral and bacterial pneumonia.

Breast cancer is also another important disease which can be addressed through imaging and this has attracted a number of databased hosting breast cancer images. The Breast Cancer Digital Repository (BCDR) [174] database contains multimodal mammography and ultrasound scan, patient history etc. data collected from 1734 anonymised patients. The data can be used for disease detection and diagnosis methods. Another dataset, MITOS [176] contains breast cancer histological images (haematoxylin and eosin stained slides). The detection of mitosis and evaluation of nuclear atypia are key uses.

4.1.3 Neuroimage Processing and Analysis

The Information eXtraction from Images (IXI) dataset [184] provides 600 MRI images from healthy subjects to study brain functions. These images saved in NIFTI file format and were acquired using protocol - T1, T2, proton-density weighted images; magnetic resonance angiography images; and diffusion weighted images. These images have been collected from three different hospitals in London, UK. Another database, called the Loni Probabilistic Brain Atlas (LPBA40) [185], contains maps of brain anatomic regions of 40 human volunteers. Each map generates a set of whole-head MRI whereas each MRI describes to identify 56 structures of brain, most of them lies in the cortex. The study of skull-stripped MRI volumes, and classification of the native-space MRI, probabilistic maps are key uses of LPBA40. The NeuroVault.org [186] is a web-based repository (API) for collecting and sharing statistical maps of the human brain to study human brain regions. The Neuroimaging Informatics Tools and Resources Clearing house (NITRC) [187], provides range of imaging data from MRI to PET, SPECT, CT, MEG/ EEG and optical imaging for analysing functional and structural neuroimages. The Open fMRI [188] dataset contains MRI images acquired using different modalities including Diffusion-weighted, T1-weighted magnetization prepared rapid acquisition with gradient echo (MPRAGE) MRI, and multi-echo fast low angle shot (FLASH) MRI. It also contains biosignal datasets to study brain regions and its functions. These can be used as a benchmark dataset in order to differentiate outcome from various neuroimaging analysis tools. The UK data service [189] contains T1/2, Diffusion Tensor Imaging and fMRI datasets from 22 patients suffering from brain tumors which can be useful for studying brain tumour surgical planning.

4.1.4 Segmentation

Segmentation is an important step in any image processing pipeline. Many datasets mentioned above can be used for segmentation purposes.

Focusing on eye diseases, the Digital Retinal Images for Vessel Extraction (DRIVE) contains JPEG Compressed retinal images of 400 diabetic patient of 25-90 years old. The dataset can be used to understand segmentation of blood vessels in retinal images and identify diabetic retinopathy. Another dataset called STructured Analysis of the Retina (STARE) was initiated in 1975. The project contains datasets of 400 raw retinal images, 10 labelled images of artery/vein and 80 images with ground truth. Each image is annotated and features are shown in image by the expert. The dataset can be used for blood vessel segmentation and optic nerve detection.

The Internet Brain Segmentation Repository (IBSR) gives segmentation results of MRI data. Development of segmentation methods is the main application of this IBSR.

4.2 Signals

Table 5 lists leading open access data repositories and datasets (also referred as data sources) pertaining to biological signals. These sources are broadly mapped to six application areas – Anomaly detection, human machine interfacing which includes brain machine interfacing as well as rehabilitation research, emotion/affective state detection, motor imagery classification, neurological condition evaluation, and signal processing and classification – which are described in the following subsections.

4.2.1 Anomaly Detection

Anomaly detection is one of the major application areas in which scientists have devoted much efforts. In this process, a number of open access data sources, largely containing EEG and ECG data, have been frequently used.

Starting with the EEG signals, the SAD mc-EEG [193] dataset contains 32 channel EEG signals from 27 subjects recorded while they were test-driving. That is, signals were acquired when each subject attended two 90 minutes virtual reality session for sustained-attention driving. The TUH EEG corpus [194] is also an open-source clinical EEG data repository for clinical EEG data, tool and documentation. The major datasets include seizure detection, abnormal EEG, EEG with artifacts (introduced by eye movement, chewing, shivering, electrode pop, electrode static, and lead artifacts, and muscle artifacts), EEG for epilepsy, etc.

Regarding the ECG signals, the MIT-BIH arrhythmia [195] arrhythmia database includes 2 channel ambulatory ECG recording taken from 47 subjects for studying arrhythmia. There are 48 complete ECG records and about 24 recordings are freely available. The PTB diagnostic ECG database [196] comprises of 549 ECG recording taken from 290 subjects of age ranged from 17 to 87 years using the conventional 12 leads ECG recorder. Each recording includes 15 signals when the subject was given 1 to 5 records. Both the datasets can be used for anomaly detection. Another ECG dataset, the TELE-ECG dataset [197], includes 250 ECG records with annotated QRS and artifact masks. It also includes QRS and artifact detection algorithms to Study QRS and artifact detection from the ECG signal.

4.2.2 Human Machine Interfacing

The application area of Human Machine Interfacing focuses on body and brain machine interfacing and rehabilitation. This is done largely through EMG and and sometimes with EEG signals.

Table 5 Application-wise categorisation of open access data repositories and datasets pertaining to biological signals

Application	Name	Description	Ref.
Anomaly detection	SAD mc-EEG	Multi-channel EEG data for sustained-attention driving task	[193]
	TUH EEG Corpus	Repository for EEG datasets, tools and documents	[194]
	MIT-BIH-ARH	ECG database of 48 recordings	[195]
	PTB D-ECG	ECG Database of 549 ECG recording	[196]
	TELE ECG	250 ECG recordings with annotated QRS and artifact masks	[197]
Human Machine Interfacing	BNCI	Various BMI signals datasets	[198]
	EMG DataRep	Various EMG datasets	[199]
	Facial s-EMG	Contains EMG data from 15 participants	[200]
	Ninapro database	Kinematic as well as the sEMG data of 27 subjects	[201]
Emotion/affective state detection	DEAP	Simultaneously recorded EMG/EEG data	[202]
	DECAF	MEG, hEOG, ECG, Trapezius muscle-EMG, face video data	[203]
	Imagine	EEG datasets of 31 subjects while listening voice	[204]
	MAHNOB-HCI	EMG, ECG and respiration and skin temperature data	[205]
	SEED	EEG dataset for emotion and vigilance	[206]
Motor imagery classification	EEG-BCI-MI	EEG signals from 13 subjects with 60,000 MI examples	[207]
	EEG-MI-BCI	EEG data from BCI for MI tasks	[208]
	EEG-MMI	EEG data from PhysioNet for MI task	[209]
Neurological condition evaluation	V-P300 BCI	16-electrodes dry EEG from 71 subjects (SP mode)	[210]
		32-electrodes wet EEG from 50 subjects (SP mode)	[211]
		32-electrodes wet EEG from 38 subjects (MPC mode)	[212]
		32-electrodes wet EEG from 44 subjects (MPCC mode)	[213]
Signal processing & classification	BCI Competition	EEG, ECoG and MEG data from a range of BCI applications	[214]
	BCI-NER challenge	56 EEG Channels's dataset decoded by a P300 speller	[215]
	DRYAD	EEG datasets of 13 subjects recorded under various condition	[216]
	Physionet	Various EEG, ECG, EMG and sEMG datasets	[217]
	UCI ML	Various ECG, EMG, sEMG datasets	[218]

Legend: MI– Motor Imagery; MMI– Motor Movement/Imagery; ERP– Event Related Potentials, SADmc-EEG– Sustained-Attention Driving multi-channel EEG; V-P300– Visual P300; SP– Single Player; MP– Multi-Player; BCI-SSVEP–Steady State Visual Evoked Potentials; EMG DataRep–EMG Datasets Repository; ARH– arrhythmia; D-ECG– Diagnostic ECG.

The BNCI Horizon 2020 database contains more than 25 datasets such as stimulated EEG datasets, ECoG-based BCI datasets, ERP-based BCI datasets, mental arithmetic, motor imagery (extracted from EEG, EOG, fNIRS, EMG) datasets, EEG/EOG datasets of Neuroprosthetic control, speller datasets. Modelling and designing of BMI devices are the key application of this database. While the BNCI contains a variety of signals, the EMG Datasets Repepository [199] includes single/ multi finger movements datasets of 2 channels, 10 classes and 8 channels, 15 classes; single/ multi fingers pressure on a steering wheel; EMG controlled multifunctional upper-limb prostheses and EMG pattern recognition datasets.

For surface EMG (sEMG), the facial s-EMG dataset contains facial s-EMG signals from the muscles corrugator supercilii, zygomaticus major, orbicularis oris, orbicularis oculi, and masseter. Archived data is from 15 participants (8 females and 7 males) aged between 26 and 57 years (mean age 40.7 ± 9.6 years). This data can be used for rehabilitation research. Also, the NinaPro database includes kinematic as well as sEMG data of 27 subjects while these subjects were moving finger, hand and wrist. These data can be employed to study biorobotics and activity detection.

4.2.3 Emotion/Affective State Detection

Emotion and affective state detection has been a very active research field over the years. A combination of different signals has been utilised in detecting emotion and affective states and a number of data sources providing these signals are described below.

A Database for Emotion Analysis using Physiological Signals (DEAP) provides various datasets for analyzing the human affective states. It provides EEG and sEMG signals of 32 volunteers while they were watching music videos to analyse the affective states. These volunteer also rated the video and The front face was also recorded for 22 volunteers. DECAF is a multimodal dataset for decoding user physiological responses to affective multimedia content. It contains magnetoencephalogram (MEG), horizontal electrooculogram (hEOG), ECG, Trapezius muscle-EMG, near-infrared face video data to study physiological and mental states. Another multimodal dataset is the MAHNOB-HCI [205] dataset which includes ECG, respiration and skin temperature data in addition to 32-channel EEG signals from 30 subjects while they were watching movie clips and photos. The different sensors were synchronised to record a synchronised multimodal dataset. The subjects were asked to label their own emotion state.

On the other hand, the Imagined Emotion [204] dataset provides EEG signals recorded when subjects were listening to voice recording. The SJTU Emotion EEG Dataset [206] contains three individual datasets (SEED, SEED-IV and SEED-VIG) of EEG signals. In the SEED dataset EEG signals were recorded while the subjects were watching movie clips and annotated their emotional state as positive, negative and neural. In case of SEED-IV, four emotional states such as happy, sad, fear, and neutral were annotated. Whereas, the SEED-VIG dataset contains EEG signals related to vigilance when the subjects were driving.

4.2.4 Motor Imagery Classification

Motor imagery (MI) is yet another very active area of research. As an outcome of a large number of community contributors, many datasets have been developed from which the popular ones are described below.

The Electroencephalographic brain-computer interface mental imagery (EEG-BCI-MI [207] dataset contains 60 hours of EEG recording from 13 subjects and 75 experiments. This contains around 60,000 mental imagery examples which is approximately 4.8 hours of EEG recordings (with 4600 MI examples) per participant. The datasets can be used for the rehabilitation of patients having movement disorders. Another EEG datasets for MI brain computer interface (EEG-MI-BCI) [208] contains EEG signals with 3-D electrode location and EEG for non-task related states as well. The datasets were recorded from 52 participants which also contain [physio/psyco]logical data and EMG signals in addition to the EEG. These datasets can be employed to find the human factors which influences MI BCI performances. Yet another EEG signal centric dataset is called, EEG motor movement/ imagery (EEG-MMI) dataset [209], incorporates 1500 ($1-2$ minutes) EEG recordings taken from 109 volunteers. The dataset can be used in designing BCI-systems for rehabilitation purposes.

4.2.5 Neurological condition evaluation

A number of visual P300-based datasets are available with open access attributes to perform a range of neurological condition evaluation. These datasets, V-P300 BCI, are composed of data recorded using dry or wet electrode with 16 or 32 channels while the subjects were playing the Brain Invaders game [219]. These datasets were recorded using different playing modalities such as single player (16 dry electrodes [210] from 71 subjects and 32 wet electrodes [211] from 50 subjects), multiplayer in collaborative mode (32 wet electrodes from 38 subjects [212]), and multiplayer cooperation and competition mode (32 wet electrodes from 44 subjects [213]).

4.2.6 Signal Processing and Classification

To solve various signal processing and classification problems, a number of datasets have been made available under open access. Most of these problems are released to the community in the form of challenges with relevant datasets to solve them. The competitions during the BCI meetings have served this purpose for several years and have released datasets (the BCI competition datasets [214]) which are still available with relevant problem statements and sample codes for others to use. The challenge datasets provided by the IEEE Neural Engineering Conference (NER2015) is known as BCI-NER dataset [215]. This dataset was mainly intended for methodological development of an error potential detection algorithm suitable for the P300-based BCI systems. The BCI Competition datasets include EEG datasets (e.g., cortical negativity or positivity, feedback test trials, self-paced key typing, P300 speller paradigm, motor/ mental imagery data, continuous EEG; EEG with eye movement), ECoG datasets (e.g., finger movement, motor/ mental imagery signals in the form of EEG/ ECoG) and MEG dataset (e.g., wrist movement). These datasets can be used for signal processing and classification methods for BMI. Similarly, the BCI-NER Challenge [215] dataset provides 56-channel EEG signals from 26 subjects using a P300 speller.

In addition to the datasets released for challenges and competitions, there are repositories which provide rich datasets for this application area. The DRYAD [216] is a versatile repository which has been recently unveiled. It contains a range of EEG recorded datasets when 19 subjects listen to natural speech time-reversed speech, cocktail party attention, and noisy audiovisual speech. The Physionet repository [217] contains a large number of neuroelectric and myoelectric datasets. As the name suggests, it is mainly for physiological data.

These datasets mainly pertain to signals such as EEG, ECoG, EMG, ECG and are acquired from many diverse experimental settings. The UCI ML repository [218] contains a large number of diverse datasets with direct application to machine learning methods. Some relevant biosignal datasets include ECG, EEG, (s)EMG signals from diverse experimental and physiological conditions.

4.3 Sequences

Table 6 lists the leading popular open access data sources pertaining to the various omics related research which includes genomics, proteomics, and metabolomics. Grouped to six broad application areas, namely – bioassay analysis and drug design, genetic disorder analysis, nucleic acid research, protein structure analysis, signal transduction pathway study, and single-cell omics, the following subsections provide brief discussions about the leading open access omics data sources.

4.3.1 Bioassay Analysis and Drug Design

Since Decemeber 2019, the world has experienced a pandemic caused by the SARS-CoV-2 (COVID-19) virus. Triggered by the necessity to facilitate the ongoing researches, the SARS-CoV-2 [220] dataset provides gene sequence, proteins, pathway and bioassay for SARS-CoV-2 along with compounds used in clinical trials. This dataset can be used for studying biological/chemical process and drug design.

The PubChem database [221] contains millions of compound structures and descriptive datasets of chemical molecules and their activities against biological assays. Maintained by the National Center for Biotechnology Information of the United States National Institutes of Health, it can be freely accessed through a web user interface and downloaded via FTP. It also contains software services (such as plotting and clustering). It can be use for [gen/prote]-omics study and drug design.

4.3.2 Genetic Disorder Analysis

The cancer gene expression (GE) [222] serves as a small repository containing several cancer GE datasets which can be employed for designing tool/algorithm for cancer detection. The cancer genome atlas (TCGA) [224] repository contains more than 2.5 petabytes of genomic, epigenomic, transcriptomic, and proteomic data. It contains data about 33 different cancer types and over 20,000 samples. These data are generated by the National Cancer Institute and the National Human Genome Research Institute. This repository is used in facilitating genomic study for improving the prevention, diagnosis, and treatment of cancer. To analyse region specific diseases, the Indian Genetic Disease Database (IGDD) [223] tracks mutations in the normal genes for genetic diseases reported in India.

4.3.3 Nucleic Acid Research

The Berkeley Drosophila Transcription Network Project (BDTNP) [225] database contains datasets pertaining to 3D Gene expression data, in-vivo and in-vitro DNA binding data as well as Chromatin Accessibility data (ChAcD). Research on GE and anomaly detection are the key applications of the datasets provided by this database.

The Encyclopedia of DNA Elements (ENCODE) [226] is a whole-genome database curated by the ENCODE Consortium. It contains a large number of datasets pertaining to functional genomics and characterisation data including meta data of human, worm, mouse, and fly. Another database, called the Exome Sequencing Project (ESP) [227], includes genome datasets which can be used to find lung and blood disorders and their management and treatment. The Gene Expression Omnibus (GEO) [228] is a open access functional genomics (microarray and sequence) data repository. This database can be used for functional genomic and epigenomic studies such as genome methylation, chromatin structure, and genome–protein interactions. It is supported by the National Center for Biotechnology Information at the National Library of Medicine of the USA [228]. The Genome Aggregation Database (gnomAD) [229] database contains large scale exome and genome sequencing data from different sequencing projects. The dataset can be used for disease diagnosis and genetic studies. The Genotype-Tissue Expression (GTEx) [230] database contains GE datasets of 54 healthy tissue sites collected from 1000 subjects and histology images. It also includes samples from GTEx biobank.

The Harmonizome [231] database provides details about genes and proteins from 114 datasets provided by 66 online resources with 71927784 associations between 295496 attributes and 56720 genes. The International Nucleotide Sequence Database [232], popularly known as INSDC, corroborates biological data from three major sources: i) DNA Databank of Japan [247], ii) European Nucleotide Archive [248], and iii) GenBank [249]. These sources provide the spectrum of data raw reads, though alignments and assemblies to functional annotation, enriched with contextual information relating to samples and experimental configurations. Similar to this, the International Genome Sam-

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

Table 6 Application-wise categorisation of open access data repositories and datasets pertaining to Omics data

Application	Name	Description	Ref.
Bioassay analysis & drug design	COVID-19	Gene sequence, pathway and bioassay datasets of COVID-19	[220]
	PubChem	Contains compound structures, molecular datasets and tool	[221]
Genetic disorder analysis	Cancer GeEx	Different cancer genome datasets	[222]
	IGDD	Mutation data on common genetic diseases	[223]
	TCGA	Contains Cancer Genome data	[224]
Nucleic acid research	BDTNP	3D Gene expression, DNA binding data & ChAcD	[225]
	ENCODE	Human genome dataset	[226]
	ESP	Contains sequencing data	[227]
	GEO	Contains high-throughput GE and functional genomics datasets	[228]
	gnomAD	Large scale exomes and genomes sequencing data	[229]
	GTEx	Gene expression datasets	[230]
	Harmonizome	Collection of genes and proteins datasets	[231]
	INSDC	Contains nucleotide sequence data	[232]
	IGSR	Genome data of various ethnicity, age and sex	[233]
	JASPAR	transcription factor DNA-binding preferences dataset	[234]
	NIHREM	Human genome datasets	[235]
	NSD	Includes omics and Health science data	[236]
	SysGenSim	Bioinformatics tools and gene sequence dataset	[237]
Protein structure analysis	PDB	Proteins, nucleic acids, and complex assemblies data	[238]
	SCOP2	Contains structural classification of proteins	[239]
	SCOPe		[240]
	UCI MB	2ps and splice-junction gene sequences	[241]
Signal transduction pathway study	NCI Nature	Molecular interactions and reactions of cells	[242]
	NetPath	Signal transduction pathways in humans	[243]
	Reactome	Database for reactions, pathways and biological processes	[244]
Single-cell omics	miRBoost	The genomes of eukaryotes containing at least 100 miRNAs	[245]
	SGD	Provides biological data for budding yeast and analysis tool	[246]

ple Resource (IGSR) [233] includes genome sequencing data from 1000 genomes project. The genome data was taken from people of various ethnicity, age and sex with the final dataset contains gene sequencing data from 2,504 individuals from 26 populations. These data can be used for disease diagnosis and genetic studies. Also, the SysGenSim [237] database includes bioinformatics tool, and Pula-Magdeburg single-gene knockout, Stat-Seq and DREAM 5 benchmark datasets for studying Gene Sequence.

JASPAR [234] is a database for transcription factor DNA binding profile. The data spans through six different taxonomic groups covering Vertebrata, Nematoda, Insecta, Plantae, Fungi, and Urochordata. The database can be used for translational genomics research.

The NIH Roadmap Epigenomics Mapping repository (NIHREM) [235] includes 2,804 datasets, i.e., 1,821 histone modification, 360 DNase, 277 DNA methylation, and 166 RNA-Seq datasets. The repository provides 3,174-fold 150.21 billion mapped sequencing the human and tools for analyzing these datasets. It can be used for stem cell mapping, selection of tissues that are responsible for human disease. Also, the database known as Nature scientific data (NSD) [236] includes datasets pertaining to omics, taxonomy and species diversity, mathematical and modelling resources, cytometry, organism-focused resources and health science data. This can be used for studying and modelling different aspect of genomics.

4.3.4 Protein Structure Analysis

The Protein Data Bank (PDB) [238] contains 3-D structural data proteins and nucleic acids. These data are obtained tools such as X-ray crystallography, NMR spectroscopy, and cryo-electron microscopy. It includes more than 135 thousand data of proteins, nucleic acids, and complex assemblies. These can be used to understand all aspects of biomedicine and agriculture.

Structural classification of proteins or in short SCOP is a repository which hosts manually classified protein structure datasets. The classification was done based on amino acid sequences and their structural similarity. The main objective is to find the evolutionary relationship between the proteins. Currently two versions of SCOP are maintained. The SCOP Version 2 (shortly called SCOP2) [239] is the up to date Structural Classification of Proteins database released at the first quarter of 2020. In contrast, the SCOP-extended (SCOPe) [240] is an extended version of the original SCOP maintained by UC Berkeley. SCOPe includes many new classified protein structures via a fusion of manual and automation curation.

Molecular Biology Databases at the UCI (UCI MB) contains three individual databases: i) Secondary Protein Structure [241], which is a bench repository that classifies secondary structure of certain globular proteins; ii) Splice-Junction Gene Sequences [250], which contains primate splice-junction gene sequences (DNA) with associated imperfect domain theory; and iii) Promoter Gene Sequences [251], which contains E. Coli promoter gene sequences (DNA) with partial domain theory. Objectives- i) Sequencing and predicting the secondary structure of certain proteins; ii) Study primate splice-junction gene sequences (DNA) with associated imperfect domain theory; iii) Study E. Coli promoter gene sequences (DNA) with partial domain theory.

4.3.5 Signal Transduction Pathway Study

The NCI-Nature Pathway Interaction Database [242] hosts cellular signaling (molecular interactions/reactions) pathways in humans. The database can be employed for cancer research. The database was created by the U.S. National Cancer Institute, NIH with the collaboration of Nature Publishing Group and published in the last quarter of 2006. Another database, NetPath [243], also contains signal transduction pathways in humans. Created jointly by Johns Hopkins University and the Institute of Bioinformatics (IOB) in India, it includes 45 signaling pathway ranging from protein-protein interactions to enzyme-protein substrate reactions including 10 major pathway of immune system and 10 pathway relevant to cancer regulation. The other one, Reactome [244], is an open access database hosting biological pathways of metabolic processes to hormonal signalling in humans. Created through a collaboration between North America and Europe, it can be used for cancer research and treatment.

4.3.6 Single-cell Omics

The miRBoost dataset [245] contains the genomes of eukaryotes containing at least 100 miRNAs. This dateset is used for Studying post-transcriptional gene regulation (PTGeR) and miRNA-related pathology. Saccharomyces Genome Database (SGD) [246] also provides complete biological information for the budding yeast *Saccharomyces cerevisiae*. They also give an open source tool for searching and analyzing these data, and thereby enable the discovery of functional relationships between sequence and gene products in fungi and higher organisms. The Study of Genome expression, transcriptome and computational biology are main function of the SGD.

5 Open Source Deep Learning Tools

Due to surging interest and concurrent multidisciplinary efforts towards DL in the recent years, several open source libraries, frameworks, and platforms have been made available to the community. However, for a newcomer to the field of biological data mining using these tools, it is not always straight forward to know their characteristics, advantages and disadvantages. In this process, one of the main hurdles for an early analyst is to select the appropriate DL architecture/model and relevant library providing suitable implementations of the selected architecture. Towards introducing a beginner to the field of biological data analysis using these open source tools, this section describes them in tutorial style indicating their characteristics, pros and cons. The focus of the section has been to review and summarize the most popular open source tools, which aim to facilitate the technological developments for the community. This comprehensive collection contains tools (also developed by individuals) which are well maintained with a reasonable amount of implemented algorithms. For the sake of brevity, the individual publication references of the tools are omitted and interested readers may consult them at their respective websites from the provided urls.

Table 7 summarizes the main features and differences of the various tools. To measure the impact and acceptability of a tool in the community, we provide GitHub based measures such as, numbers of Stars, Forks, and Contributors. These numbers are indicative of the popularity, maturity, and diffusion of a tool in the community.

5.1 Caffe

Caffe (`http://caffe.berkeleyvision.org/`) is scalable, written in C++ and provides bindings for Python as well as Matlab. Dedicated for experiment, training,

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

Table 7 Summary of Open Source Deep Learning Tools

Tool	Platform	Language(s)	Stars*	Forks*	Contrib.*	Supported DL Architecture
Caffe[2]	L, M, W, A	Py, C++, Ma	30100	18200	266	CNN, RNN, GAN
Chainer[3]	L	Py	5300	1400	251	DA, CNN, RNN, GAN
DL4j[1]	L, M, W	Ja	11500	4800	32	DA, CNN, RNN, RBM, LSTM, GAN
DyNet [1]	L	C++	3000	687	117	CNN, RNN, LSTM
H_2O[1]	L, M, W	Ja, Py, R	4700	1700	132	CNN, RNN
Keras[3]	L, M, W	Py	47500	18000	816	CNN, RNN, DBN, GAN
Lasagne[1]	L, M	Py	3700	980	68	CNN, RNN, LSTM, GAN
MCT[3]	W	C++	16720	4400	197	CNN, DBN, RNN, LSTM
MXNet[1]	L, M, W, A, I	C++	18500	6600	780	DA, CNN, RNN, LSTM, GAN
Neon[1]	L, M	Py	3800	846	78	DA, CNN, RNN, LSTM, GAN
PyTorch[2]	L, M	Py	37400	9500	1345	CNN, RNN, LSTM, GAN
Singha[1]	L, M, W	Py, C++, Ja	2000	499	46	CNN, RNN, RBM, DBM
TensorFlow[1]	L, M, W	Py, C++	14300	80600	2450	CNN, RNN, RBM, LSTM, GAN
TF.Learn[3]	L, M	Py, C++	9400	2400	120	CNN, BRNN, RNN, LSTM, GAN
Theano[2]	L, M, W	Py	9103	2500	332	CNN, RNN, RBM, LSTM, GAN
Torch[2]	L, M, W, A, I	Lu, C, C++	8495	2400	130	CNN, RNN, RBM, LSTM, GAN
Veles[1]	L, M, W, A	Py	891	185	10	DA, CNN, RNN, LSTM, RBM

*GitHub parameters (as of 1 April. 2020); [1]Apache2 License; [2]BSD License; [3]MIT License;
Legends: L–Linux/Unix; M–MacOSX; W–Windows; A–Android; I–iOS; CP–Cross-platform; Py–Python; Ja–Java; Lu–Lua; Ma–Matlab.

and deploying general purpose DL models, this framework allows switching between development and deployment platforms. Targeting computer vision applications, it is considered as the fastest implementation of the CNN.

Pros.

- Easy to deploy;
- Pre-trained models are available;
- Faster training speed;
- Used for feedforward networks.

Cons.

- Requires to writing code for generating new layers;
- Less support for recurrent networks;
- No support for distributed training.

5.2 Chainer

Chainer (`http://chainer.org/`) is a DL framework provided as Python library. Besides the availability of popular optimization techniques and NN related computations (e.g., convolution, loss, and activation functions), dynamic creation of graphs makes Chainer powerful. It supports a wide range of DL architectures including CNN, GAN, RNN, and DA.

Pros.

- One of the tool for leading dynamic computation graphs/networks;
- Notably faster than other Python-oriented frameworks.

Cons.

- Open Computing Language framework/Open Multi-Processing API is not supported.

5.3 DeepLearning4j

Deeplearning4j (DL4J, `https://deeplearning4j.org/`), written in Java with core libraries in C/C++, is a distributed framework for quick prototyping that targets mainly nonresearchers. Compatible with JVM supported languages (e.g., Scala/Clojure), it works on distributed processing frameworks (e.g., Hadoop and Spark). Through Keras (section 5.6) as a Python API, it allows importing existing DL models from other frameworks. It allows creation of NN architectures by combining available shallow NN architectures.

Pros.

- Support integration with Big learning frameworks Apache Spark and Hadoop;
- Support distributed GPUs and CPUs platforms and able to work with tensor.

Cons.

- Open Computing Language framework is not supported;
- GUI is for supported for workflow and visualisation.

5.4 DyNet

The DyNet library (`https://dynet.readthedocs.io/`), written in C++ with Python bindings, is the successor of 'C++ neural network library'. In DyNet, computational graphs are dynamically created for each training example, thus, it is computationally efficient and flexible. Targeting NLP applications, its specialty is in CNN, RNN, and LSTM.
Pros.

- Designed to be efficient for running on CPU or GPU.
- Dynamic computation graph lilke PyTorch and Chainer.

Cons.

- In terms of Tensorflow, limited functions are available.

5.5 H_2O

H_2O (`www.h2o.ai`) is an ML software that includes DL and data analysis. It provides a unified interface to other DL frameworks like, TensorFlow, MXNet, and Caffe. It also supports training of DL models (CNN and RNN) designed in R, Python, Java, and Scala.
Pros.

- Due to its in-memory distributed parallel processing capacities, it can be used for real-time data;
- GUI is supported (Called Flow) for workflow and visualization;
- GPU support for Deep Water and NVIDIA;
- Fast training, memory-efficient DataFrame manipulation;
- Easy to use algorithms and well documented;

Cons.

- Lacks the data manipulation capabilities of R and Pandas DataFrames;
- Slow learning and support limited model model running at a time.

5.6 Keras

The Python based Keras (`https://keras.io/`) library is used on top of Theano or TensorFlow. Its models can be imported to DL4J (section 5.3). It was developed as a user friendly tool enabling fast experimentation, and easy and fast prototyping. Keras supports CNN, GAN, RNN, and DBN [252].
Pros.

- Rich documentation;
- A high-level API for neural networks;
- Ability to run on top of state-of-art deep learning libraries/frameworks such as TensorFlow, CNTK or Theano.

Cons.

- Cannot utilize a multi-GPU directly;
- Require Theano as backend for OpenMP support and Theano/Tensor Flow/PlaidML as backend for OpenCL.

5.7 Lasagne

Lasagne (`http://lasagne.readthedocs.io`) DL library is built on top of Theano. It allows multiple input, output, and auxiliary classifiers. It supports user defined cost functions and provides many optimization functions. Lasagne supports CNN, GAN, RNN, and LSTM.
Pros.

- Lasagne is a lightweight library to build and train DL algorithms in Theano;
- Layers, regularizers, optimizers can be used independently;
- Clear documentation is available;
- Supports training the network on a GPU.

Cons.

- Small community then Tensor Flow.

5.8 Microsoft Cognitive Toolkit

Replacing CNTK, the Microsoft Cognitive Toolkit (MCT, `https://cntk.ai/`) is mainly coded in C++. It provides implementations of various learning rules and supports different DL architectures including DNN, CNN, RNN, and LSTM.
Pros.

- It is a framework for feed-forward DNNs, CNN and RNN;
- Can train production systems very fast;
- Can achieve state-of-the-art performance on benchmark tasks;
- Allow directed graph visualization.

Cons.

- Less community support;
- Difficult to install;
- Draw lass interest among the research community.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

5.9 MXNet

MXNet (`https://mxnet.io/`) framework allows defining, training, and deploying deep NN (DA, CNN, GAN, RNN and LSTM) on a wide range of devices– from cloud infrastructure to mobile or even embedded devices (e.g., Raspberry Pi). Written in C++, it is memory efficient and supports Go, JavaScript, Julia, Matlab, Perl, Python, R, and Scala.

Pros.

- A DL framework which has a high-performance imperative API;
- Rich Language support;
- MXNet features advanced GPU support;
- Highly scalable.

Cons.

- Small community then Tensor Flow;
- Poor API documentation available;
- Less popular among the research community.

5.10 Neon

Neon (`www.nervanasys.com/technology/neon/`) is a DL framework written in Python. It provides implementations of various learning rules, along with functions for optimization and activation. Its support for DL architecture includes CNN, GAN, RNN, LSTM, and DA.

Pros.

- Better visualization properties than other framework;
- Apply optimization at data loading level,

Cons.

- Small community then Tensor Flow;
- Less popular among the research community.

5.11 PyTorch

PyTorch (`http://pytorch.org/`) provides Torch modules in Python. More than a wrapper, its deep integration allows exploiting the powerful features of Python. Inspired by Chainer, it allows dynamic network creation for variable workload, and supports CNN, GAN, RNN and LSTM.

Pros.

- Pretrained models are available;
- OpenCL support Via separately maintained package.
- Easily combine modular pieces;
- Easy to create a layer and run on GPU.

Cons.

- Require to write training code;
- Less documentation available.

5.12 Singa

Singa (`https://singa.incubator.apache.org/`), it is a distributed DL platform written in C++, Java, and Python. It's flexible architecture allows synchronous, asynchronous, and hybrid training frameworks to run. It supports a wide range of DL architectures including CNN, RNN, RBM, and DBM.

Pros. (10.1145/2733373.2807410)

- Pre-trained models are available;
- Support model/data or hybrid partitioning, and synchronous/asynchronous/hybrid training;
- Distributed deep learning system and handle Big data.
- Widely used for healthcare data analytic.

Cons.

- No Open Multi-Processing support.

5.13 TensorFlow

TensorFlow (`www.tensorflow.org`), written in C++ and Python, was developed by Google and supports very-large-scale deep NN. Amended recently as 'TensorFlow Fold', its capability to dynamically create graphs made the architecture flexible, allowing deployment to a wide range of devices (e.g., multi-CPU/GPU desktop, server, mobile devices, etc.) without code rewriting [253, 254]. Also contains a data visualization tool named TensorBoard and supports many DL architectures including CNN, GAN, RNN, LSTM, and RBMs [255].

Pros.

- Handle large scale date and operate in heterogeneous environments;
- Faster compile time than Theano;
- Computational graph abstraction;
- Support parallelism.
- TensorBoard is used for workflow and visualization.

Cons.

- Large memory footprint;
- Less number of pretrained models are available;
- Computational graph can be slow;
- No support for matrix operations;
- Difficulties in Debugging.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

5.14 TF.Learn

TF.Learn (`www.tflearn.org`) is a TensorFlow (section 5.13) based high level Python API. It supports fast prototyping with modular NN layers and multiple optimizers, inputs, and outputs. Supported DL architectures include CNN, GAN, BRNN, and LSTM.
Pros.

– Modular and transparent DL library built on the top of Tensorflow;
– Provides a higher-level API to TensorFlow.

Cons.

– Slower compared to its competitors.

5.15 Theano

Theano (`www.deeplearning.net/software/theano/`) is a Python library that builds on core packages like NumPy and SymPy. It defines, optimizes, and evaluates mathematical expressions with tensors, and served as foundation for many DL libraries.
Pros.

– High flexibility;
– High computational stability;
– Well suited for tensor based mathematical expressions;
– Open-source libraries such as Keras, Lasagne and Blocks builts on the top of Theano;
– Able to visualize convolutional filters, images, and graphs;
– High level wrappers like Keras and Lasagne increases usability.

Cons.

– Difficult to learn;
– Difficult to deploy;
– Deployed on single GPU;
– Slower compilation time than Tensor Flow.

5.16 Torch

Started in 2000, Torch (`http://torch.ch/`), a ML library and scientific computing framework, has evolved as a powerful DL library. Core functions are implemented in C and the rest via LuaJIT scripting language made Torch super fast. Software giants like Facebook and Google use Torch extensively. Recently Facebook's DL modules (fbcunn) focusing on CNN have been open-sourced as a plug-in to Torch.
Pros.

– User friendly;
– Convenient for employ with GPUs;
– Pretrained models are available;
– Highly modular;
– Easy to create a layer and run on GPU.

Cons.

– Special data format and requires conversion;
– Require to write training code;
– Less documentation available.

5.17 Veles

Veles (`https://github.com/Samsung/veles`) is a Python based distributed platform for rapid DL application development. It provides machine learning and data processing services and supports IPython notebooks. Developed by Samsung, one of its advantages is that, it supports OpenCL for cross-platform parallel programming, and allows execution across heterogenous platforms (e.g., servers, PC, mobile, and embedded devices). The supported DL architectures include– DA, CNN, RNN, LSTM, and RBM.
Pros.

– Distributed platform support;
– Support Jupyter Notebook;
– Supports OpenCL for cross-platform parallel programming.

Cons.

– Less community support;
– Draw lass interest among the research community.

6 Relative Comparison of DL Tools

To perform relative comparison among the available open-source DL tools, we selected four metics which are detailed below: trend in their usage, community participation in their development, interoperability among themselves, and their scalability (see Fig. 4).

6.1 Trend

To assess the popularity and trend of the various DL tools among the DL consumers, we looked into two different sources to assess the utilization of the tools. Firstly, we extracted globally generated search data from Google Trends[1] for five years (January 2015 to December 2019) related to search terms consisting of ⟨[tool

[1] https://trends.google.com/

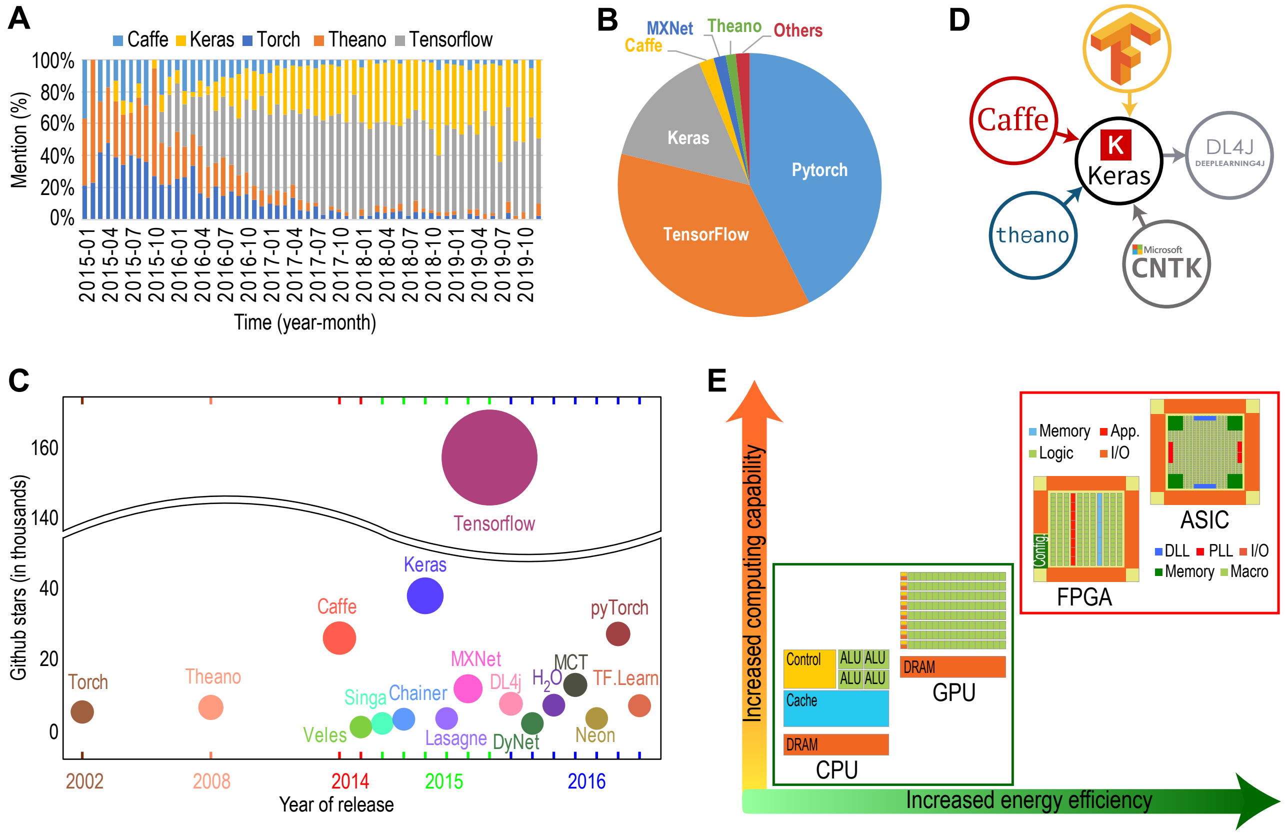


Fig. 4 Relative Comparison of DL tools. (A) Popularity trend of individual DL tools as per mention in google search generated globally (data courtesy: google trend). (B) mention in articles submitted to arXiv preprint server during the first quarter of 2020. (C) The effect of community's participation on individual tools is shown by the bubble size, which is product of normalized number of GitHub forks and contributors. (D) As for the interoperability among the DL tools, Keras allows model importing from Caffe, MCT (CNTK), Theano, Tensorflow and lets DL4j to import. (E) Regarding hardware based scalability of the DL tools, most of the tools provide CPU and GPU support, whereas FPGA and ASIC can mainly execute pre-trained models.

name] + Deep Learning⟩. The data showed a progressive increase of search about Tensorflow since it's release followed by Keras (see Fig. 4A). Secondly, mining the content of around 2,000 papers submitted to arXiv's cs.[CV | CL | LG | AI | NE], and stat.ML categories, during the first quarter of 2020 (i.e., January to March), for the presence of the tool names [256]. As seen in Fig. 4B which shows an percentage of each individual tool's mention in the papers, the top 6 tools were identified as: Pytorch, Tensorflow, Keras, Caffe, MXNet, and Theano.

6.2 Community

The community based development score for each tool discussed in Section 5 was calculated from repository popularity parameters of GitHub (https://github.com/) (i.e., star, fork, and contributors). The bubble plot shown in Fig. 4C depicts community involvement in the development of the tools indicating the year of initial stable release. Each bubble size in the figure, pertaining to a tool, represents the normalized combined effect of fork and contributors of that tool. It is clearly seen that a very large part of the community effort is concentrated on Tensorflow, followed by Keras and Caffe.

6.3 Interoperability

In today's cross-platform development environments, an important measure to judge a tool's flexibility is it-s interoperability with other tools. In this respect, Keras is the most flexible one whose high-level neural networks are capable of running on top of either Tensor or Theano. Alternatively, DL4j-model imports neural network models originally configured and trained using Keras that provides abstraction layers on top of TensorFlow, Theano, Caffe, and CNTK backends (see Fig. 4D).

6.4 Scalability

Hardware based scalability is an important feature of the individual tools (see Fig. 4E). Today's hardware for computing devices are dominated by graphics processing units (GPUs) and central processing units (CPUs). But considering increased computing capacity and energy efficiency, the coming years are expected to witness expanded role for other chipset types including application specific integrated circuits (ASICs), and field programmable gate arrays (FPGAs). So far DL has been predominantly used through software. Requirement for hardware acceleration, energy efficiency, and higher performance allowed development of chipset based DL systems.

7 Performance of Tools and Benchmark

The power of DL methods lies in their capability to recognize patterns for which they are trained. Despite the availability of several accelerating hardware (e.g., multicore [C/G]PUs/FPGAs), this training phase is very time consuming, cumbersome, and computationally challenging. Moreover, as each tool provides implementations of several DL architectures and often emphasizing separate components of them on different hardware platforms, selecting an appropriate tool suitable for an application is getting increasingly difficult. Besides, different DL tools have different targets, e.g., Caffe targets applications, whereas, Torch and Theano are more for DL research. To facilitate scientists in picking the right tool for their application, scientists benchmarked the performances of the popular tools concerning their training times [257,258]. Moreover, to the best of our knowledge, there exist two main efforts that provide the benchmarking details of the various DL tools and frameworks publicly [259,260]. Summarizing those seminal works, below we provide the time required to complete the training process as a performance measure of four different DL architectures (e.g., FCN, CNN, RNN, and DA) among the popular tools (e.g., Caffe, CNTK, MXNET, Theano, Tensorflow, and Torch) on multicore [C/G]PU platforms.

Table 8 lists the experimental setups used in benchmarking the specified tools. Mainly three different setups, each with Intel Xeon E5 CPU, were utilized during the process. Though the CPU were similar, the GPU hardware were different: GeForce GTX Titan X, GTX 980, GTX 1080, Tesla K80, M40, and P100.

Stacked autoencoders or DA were benchmarked using the experimental setup number 1 in Table 8. To estimate the performance of the various tools on implementing DA, three autoencoders (number of hidden layers: 400, 200, and 100, respectively) were stacked with tied weights and sigmoid activation functions. A two step network training was performed on the MNIST dataset [261]. As reported in Fig. 5 (a, b), the performances of various DL tools are evaluated using forward runtime and training time. The forward runtime refers to the required time for evaluating the information flow through the full network to produce the intended output for an input batch, dataset, and network. In contrast, the gradient computation time measures the time that required to train DL tools. The results suggest that, regardless of the number of CPU threads used or GPU, Theano and Torch outperforms Tensorflow both in gradient and forward times (see Fig. 5 a, b).

Experimental setup number 2 (see Table 8) was used in benchmarking RNN. The adapted LSTM network [262] was designed with 10000 input and output units with two layers and ∼13 millions parameters. As the performance of RNN depends on the input length, an input length of 32 was used for the experiment. As the results indicate (see Fig. 5 c-f), MCT outperforms other tools on both CPU and all three GPU platforms. On CPUs, Tensorflow performs little better than Torch (see Fig. 5 c). On GPUs, Torch is the slowest with Tensorflow and MXNet performing similarly (see Fig. 5 d-f).

Still a large portion of the pattern analysis is done using CNN, therefore, we further focused on CNN and investigated how the leading tools performed and scaled in training different CNN networks in different GPU platforms. Time speedup of GPU over CPU is considered as a metric for this purpose. The individual values are calculated using the benchmark scripts of DeepMark [259] on experimental setup number 3 (see Ta-

Table 8 Hardware configuration of the evaluating setup

ESN	Processor	Memory
1	**CPU:** E5-1650[1] @ 3.50 GHz	32 GB
	GPU: Nvidia GeForce GTX Titan X[2]	
2	**CPU:** E5-2630[3] @ 2.20 GHz	128 GB
	GPU: Nvidia GeForce GTX 980[4]	
	GPU: Nvidia GeForce GTX 1080[5]	
	GPU: Tesla K80 accelerator with GK210 GPUs[6]	
3	**CPU:** E5-2690[3] @ 2.60 GHz	256 GB
	GPU: Tesla P100 accelerator[7]	
	GPU: Tesla M40 accelerator[8]	
	GPU: Tesla K80 accelerator with GK210 GPUs[6]	

Legends: ESN: Experimental Setup Numbers; [1]: Intel Xeon CPU v2; [2]: 3072 cores, 1000 MHz base clock, 12 GB memory; [3]: Intel Xeon CPU v4; [4]: 2048 cores, 1126 MHz base clock, 4 GB memory; [5]: 2560 cores, 1607 MHz base clock, 8 GB memory; [6]: Tesla K80 accelerator has two Tesla GK210 GPUs with 2496 cores, 560 MHz base clock, 12 GB memory; [7]: 3584 cores, 1189 MHz base clock, 16 GB memory; [8]: 3072 cores, 948 MHz base clock, 12 GB memory.

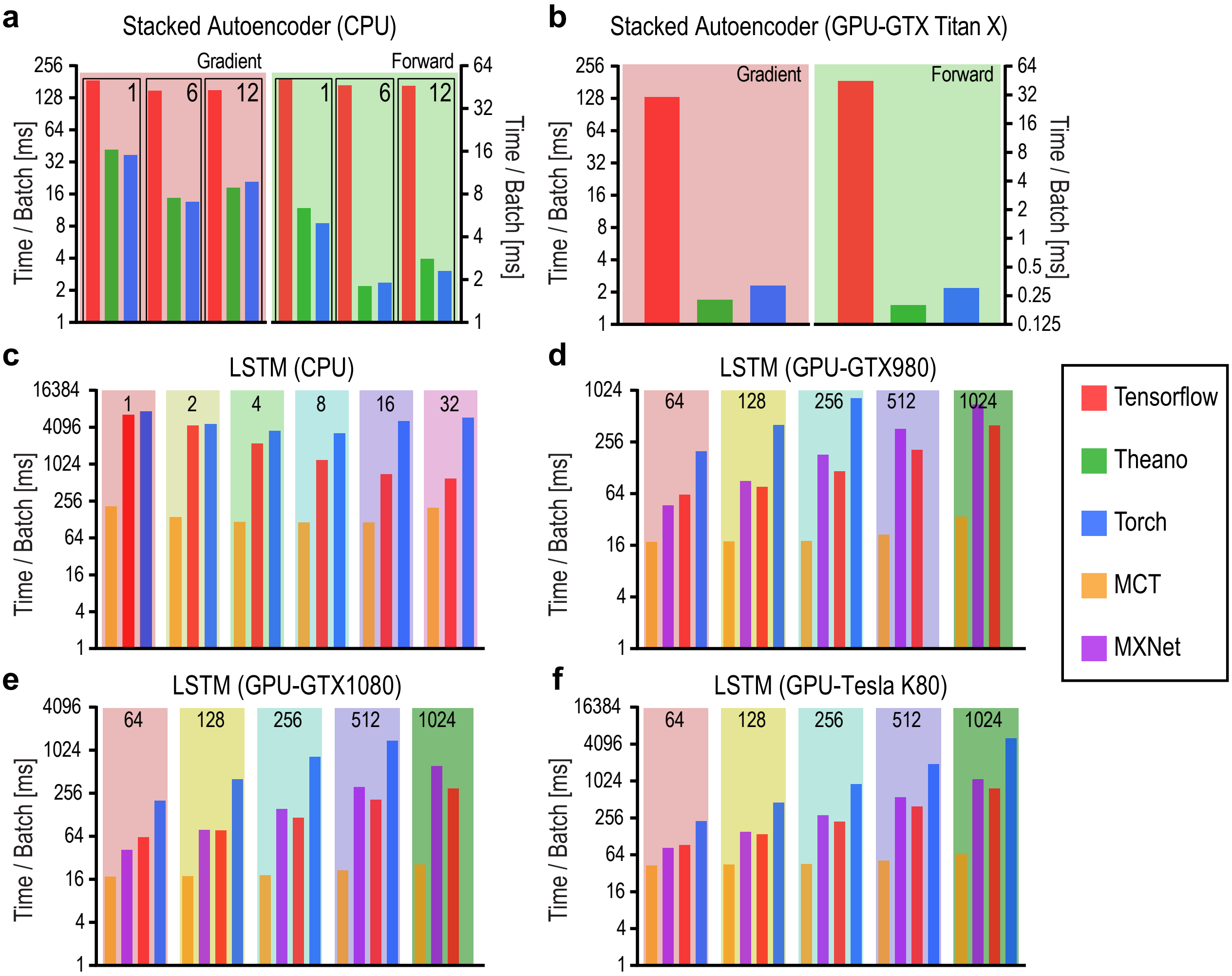


Fig. 5 Benchmarking Stacked Autoencoder or DA (a, b) and LSTM (c-f) in CPU and GPU platforms. The numbers in (a, c) denote the number of CPU threads employed in the benchmarking process, and in (d-f) denote the batch size. In case of DA the batch size was 64.

ble 8) for one training iteration per batch. The time needed to execute a training iteration per batch equals the time taken to complete a forward propagation operation followed by a backpropagation operation. Figure 6 summarizes the training time per iteration per batch for both CPU and GPUs (left y-axis), and the corresponding GPU speedup over CPU (right y-axis).

These findings for four different CNN network models (i.e., Alexnet [92], GoogLeNet [94], Overfeat [263], and VGG [93]) available in four tools (i.e., Caffe, Tensorflow, Theano, and Torch) [264] clearly suggest that network training process is much accelerated in GPUs in comparison to CPUs. Moreover, another important message is that, all GPUs are not the same and all tools don't scale up at the same rate. The time required to train a neural network strongly depends on which DL framework is being used. As for the hardware platform, the Tesla P100 accelerator provides the best speedup with Tesla M40 being the second and Tesla K80 being the last among the three. In CPUs, TensorFlow achieves the least training time indicating a quicker training of the network. In GPUs, Caffe usually provides the best speedup over CPU but Tensorflow and Torch perform faster training than Cafee. Though Tensorflow and Torch have similar performances (indicated by the height of the lines), Torch slightly outperforming Tensorflow in most of the networks. Finally, most of the tools outperform Theano.

8 Open Issues and Future Perspectives

The brain has the capability to recognize and understand patterns almost instantaneously. Over several decades, scientists have been trying decode the biological mechanism of natural pattern recognition that takes place

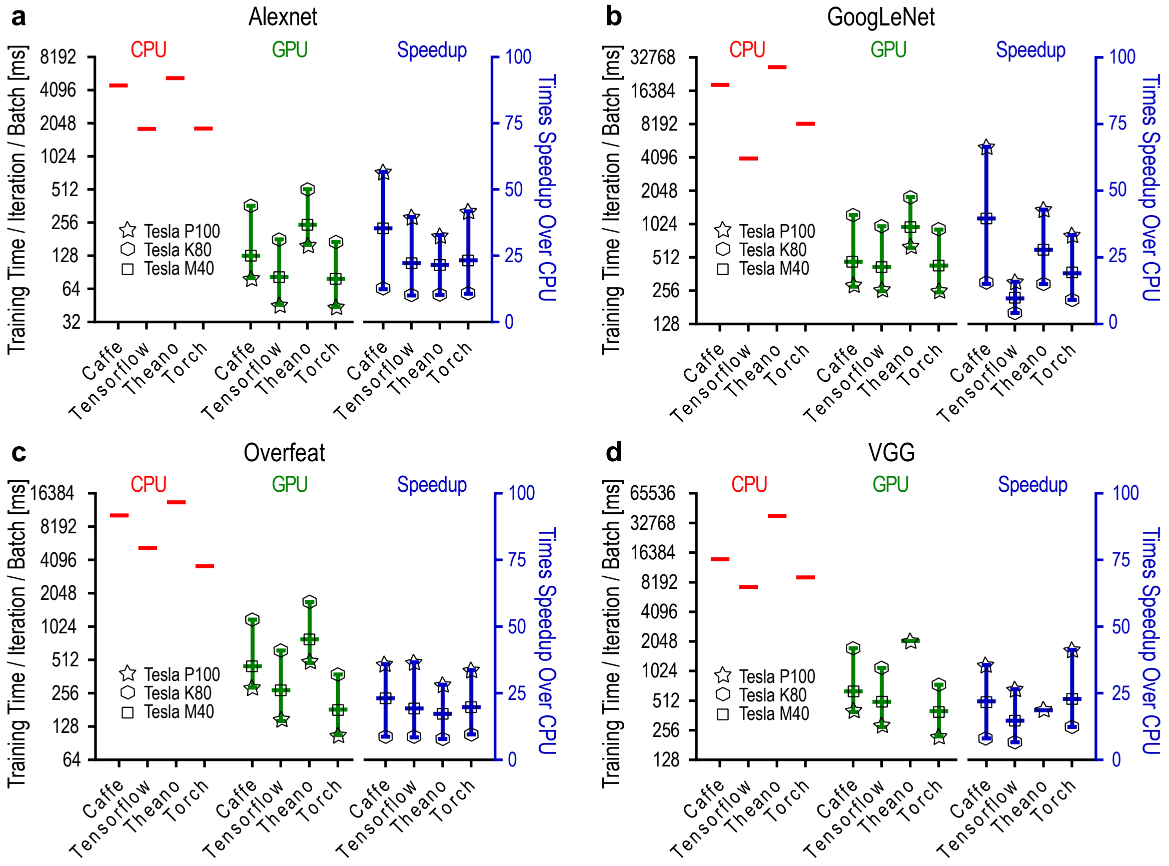


Fig. 6 The speedup of CNN training in different DL tools across various GPUs in comparison to CPU. The reported values were calculated for a batch size of 128, except for VGG for which the batch size was 64.

in the brain and translate those principles into AI systems. The increasing knowledge about the brain's information processing policies enabled this analogy to be adopted and implemented in computing systems. Recent technological breakthroughs, seamless integration of diverse techniques, better understanding of the learning systems, declination of computing costs, and expansion of computational power empowered computing systems to reach human level computation in certain scenarios [265]. Nonetheless, many of these methods require improvements. Though admittedly, there are distinctions on how a DL-based method can be used and applied on biological data, however, the common open issues and challenges are equally applicable and important for biological data. We identify below shortcomings and bottlenecks of the popular methods, open research questions and challenges, and outline possible directions which requires attention in the near future.

First of all, DL methods usually require large datasets. Though the computing cost is declining with increasing computational power and speed, it is not worthwhile to apply DL methods in cases of small to moderate sized datasets. This is particularly so as considering that many of the DL methods perform continuous geometric transformations of one data manifold to another with an assumption that there exist learnable transfer functions which can perform the mapping [266]. However, in cases when the relationships among the data are causal or very complex to be learned by the geometric transformations, the DL methods fail regardless the size of the dataset [267]. Also, interpreting high level outcomes of DL methods are difficult due to inadequate in-depth understanding of the DL theories which causes many of such models to be considered as 'Black box' [268]. Moreover, like many other ML techniques, DL is also susceptible to misclassification [269] and over-classification [270].

Additionally, the ability to exploit the full benefits offered by open access data repositories, in terms of data sharing and re-use, are often hampered by the lack of unified reporting data standards and non-uniformity of reported information [271]. Data provenance, curation, and annotation of these biological big data is a huge challenge too [272].

Furthermore, except for very few large enterprises, the power of distributed and parallel computation through cloud computing remains largely unexplored for the DL techniques. Due to the fact that the DL techniques require retraining for different datasets, repeated training becomes a bottleneck for cloud computing environments. Also, in such distributed environments, data privacy and security concerns are still prevailing [273], and real-time processing capability of experimental data is underdeveloped [274].

To mitigate the shortcomings and address the open issues, the existing theoretical foundations of the DL methods need to be improved. The DL models are required not only to be able to describe specific data but also generalize them on the basis of experimental data which is crucial to quantify the performances of individual NN models [275]. These improvements should take place in several directions and address issues like–quantitative assessment of individual model's learning efficiency and associated computational complexity in relation to well defined parameter tuning strategies, the ability to generalize and topologically self-organize based on data-driven properties. Also, to facilitate intuitive and less cumbersome interpretation of the analysis results, novel tools for data visualization should be incorporated in the DL frameworks.

Recent developments in combined methods pertaining to deep reinforcement learning (deep RL) have been popularly applied to many application domains (for a review on deep RL, see [276]). However, deep RL methods have not yet been applied to biological pattern recognition problems. For example, analyzing and aggregating dynamically changing patterns in biological data coming from multiple levels could help to remove data redundancy and discover novel biomarkers for disease detection and prevention. Also, novel deep RL methods are needed to reduce the currently required large-set of labeled training data.

Renewing efforts are required for standardization, annotation, curation, and provenance of data and their sources along with ensuring uniformity of information among the different repositories. Additionally, to keep up with the rapidly growing big data, powerful and secure computational infrastructures in terms of distributed, cloud, and parallel computing tailored to such well-understood learning mechanisms are badly needed. Lastly, there are many other popular DL tools (e.g., Keras, Chainer, Lasagne) and architectures (e.g., DBN) which need to be benchmarked providing the users with a more comprehensive list to choose. Also, the currently available benchmarks are mostly performed on non biological data, and their scalability to biological data is poor, thus, specialized benchmarking on biological data are needed.

In order to derive insights from an image, a sequence or a signal analysis problem, a selected DL algorithm using a library or a tool (e.g., TensorFlow, Keras, pyTorch, etc.) may need to integrate with a big data framework (e.g., Hadoop, Spark, etc.). In such cases, troubleshooting in the model and debugging the code may be very challenging for the system designer due to the parallel execution of multiple threads which may not always execute in an orderly fashion. The lack of documentation and model transparency of these libraries may make it impossible for the project manager to estimate efforts required in successful completion of a project.

9 Conclusion

The biological big data coming from different application domains are multimodal, multidimentional, and complex in nature. At present, a great deal of such big data are publicly available. The affordable access to these data came with a huge challenge to analyze patterns in them which require sophisticated ML tools to do the job. As a result, many ML based analytical tools have been developed and reported over the last decades and this process has been facilitated greatly by the decrease of computational costs, increase of computing power, and availability of cheap storage. With the help of these learning techniques, machines have been trained to understand and decipher complex patterns and interactions of variables in biological data. To facilitate a wider dissemination of DL techniques applied to biological big data and serve as a reference point, this article provides a comprehensive survey of the literature on those techniques' application on biological data and the relevant open access data repositories. It also lists existing open source tools and frameworks implementing various DL methods, and compares these tools for their popularity and performance. Finally, it concludes by pointing out some open issues and proposing some future perspectives.

Acknowledgements The authors would like to thank the members of the acslab (`http://www.acslab.info/`) for valuable discussions.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Authors and Contributors: This work was carried out in close collaboration among all authors. M.M. and

M.S.K. conceived the idea, developed the method and experiments, analysed the obtained data, and wrote the manuscript. T.M.M and A.H. edited the manuscript. All authors have contributed to, seen, and approved the paper.

Ethical Approval: This article does not contain any studies with human participants or animals.

Informed Consent: As this article does not contain any studies with human participants or animals, the informed consent is not applicable.

References

1. Coleman W. Biology in the nineteenth century : problems of form, function, and transformation. Cambridge ; New York: Cambridge University Press; 1977.
2. Magner LN. A history of the life sciences. 3rd ed. New York: M. Dekker; 2002.
3. Brenner S. History of science. The revolution in the life sciences [J. Article]. Science. 2012;338(6113):1427–8.
4. Shendure J, Ji H. Next-generation DNA sequencing. Nat Biotechnol. 2008 Oct;26(10):1135–1145.
5. Metzker ML. Sequencing technologies — the next generation. Nat Rev Genet. 2010 Jan;11(1):31–46.
6. Vadivambal R, Jayas DS. Bio-imaging : principles, techniques, and applications. Boca Raton, FL: CRC Press, Taylor & Francis Group; 2016.
7. Poldrack RA, Farah MJ. Progress and challenges in probing the human brain. Nature. 2015 Oct;526(7573):371–379.
8. Lebedev MA, Nicolelis MAL. Brain-Machine Interfaces: From Basic Science to Neuroprostheses and Neurorehabilitation. Phys Rev. 2017;97(2):767–837.
9. Quackenbush J. Extracting biology from high-dimensional biological data. J Exp Biol. 2007;210:1507–17.
10. Mattmann CA. Computing: A vision for data science. Nature. 2013 Jan;493(7433):473–475.
11. Li Y, Chen L. Big Biological Data: Challenges and Opportunities. Genomics Proteomics Bioinformatics. 2014 Oct;12(5):187–189.
12. Marx V. Biology: The big challenges of big data. Nature. 2013 Jun;498(7453):255–260.
13. Tarca AL, Carey VJ, Chen Xw, Romero R, Draghici S. Machine learning and its applications to biology. PLoS Comput Biol. 2007;3(6):e116.
14. Hopfield JJ. Artificial neural networks. IEEE Circuits Devices Mag. 1988 Sep;4(5):3–10.
15. Hecht-Nielsen R. Theory of the backpropagation neural network. In: Proc. IJCNN 1989; 1989. p. 593–605.
16. Hopfield JJ. Neurons with graded response have collective computational properties like those of two-state neurons. PNAS. 1984 May;81(10):3088–3092.
17. Ackley DH, Hinton GE, Sejnowski TJ. A Learning Algorithm for Boltzmann Machines. Cogn Sci. 1985 Jan;9(1):147–169.
18. Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann Machines for Collaborative Filtering. In: Proc. ICML; 2007. p. 791–798.
19. Maass W. Networks of spiking neurons: The third generation of neural network models. Neural Netw. 1997 Dec;10(9):1659–1671.
20. Heckerman D. A Tutorial on Learning with Bayesian Networks. In: Jordan MI, editor. Learning in Graphical Models. 89. Springer Netherlands; 1998. p. 301–354. DOI: 10.1007/978-94-011-5014-9_11.
21. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995 Sep;20(3):273–297.
22. Yuan GX, Ho CH, Lin CJ. Recent Advances of Large-Scale Linear Classification. Proc IEEE. 2012 Sep;100(9):2584–2603.
23. Fisher RA. The Use of Multiple Measurements in Taxonomic Problems. Ann Eugenics. 1936 Sep;7(2):179–188.
24. Uysal I, Güvenir HA. An Overview of Regression Techniques for Knowledge Discovery. Knowl Eng Rev. 1999 Dec;14(4):319–340.
25. Rish I. An empirical study of the naive Bayes classifier. In: Proc. 2001 IJCAI. vol. 3; 2001. p. 41–46.
26. Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans Inf Theory. 1967 Jan;13(1):21–27.
27. Rabiner L, Juang B. An introduction to hidden Markov models. IEEE ASSP Mag. 1986 Jan;3(1):4–16.
28. Kohavi R, Quinlan JR. Data Mining Tasks and Methods: Classification: Decision-tree Discovery. In: Klösgen W, Zytkow JM, editors. Handbook of Data Mining and Knowledge Discovery. New York, NY, USA: Oxford University Press, Inc.; 2002. p. 267–276.
29. Hinton GE. Connectionist Learning Procedures. Artif Intell. 1989 Sep;40(1-3):185–234.
30. Dempster AP, Laird NM, Rubin DB. Maximum Likelihood from Incomplete Data via the EM Algorithm. J R Stat Soc Series B Methodol. 1977;39(1):1–38.
31. Tishby N, Pereira FC, Bialek W. The Information Bottleneck Method. In: Proc. 37th ACCCC. Urbana-Champaign, Illinois, US; 1999. p. 368–377.
32. Kohonen T. Self-organized formation of topologically correct feature maps. Biol Cybernet. 1982;43(1):59–69. Available from: https://doi.org/10.1007/BF00337288.
33. Agrawal R, Imieliński T, Swami A. Mining Association Rules Between Sets of Items in Large Databases. In: Proc. ACM SIGMOD '93. New York, NY, USA: ACM; 1993. p. 207–216.
34. Gordon AD. A Review of Hierarchical Classification. J R Stat Soc Series A General. 1987;150(2):119–137.
35. Ball G, Hall D. ISODATA, a novel method of data anlysis and pattern classification. Stanford, CA: Stanford Research Institute; 1965.
36. Dunn JC. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. J Cybernet. 1973 Jan;3(3):32–57.
37. John A H. Clustering algorithms. New York, NY, USA: John Wiley & Sons, Inc.; 1975.
38. Kriegel HP, Kroger P, Sander J, Zimek A. Density-based clustering. WIRES Data Min Knowl. 2011;1(3):231–240.
39. Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: Ordering Points to Identify the Clustering Structure. In: Proc. ACM SIGMOD'99. New York, NY, USA: ACM; 1999. p. 49–60.
40. Horgan RP, Kenny LC. 'Omic' technologies: genomics, transcriptomics, proteomics and metabolomics. Obstet Gynecol. 2011 Jul;13(3):189–195.
41. Libbrecht MW, Noble WS. Machine learning applications in genetics and genomics. Nat Rev Genet. 2015 Jun;16(6):321–332.
42. Lemm S, Blankertz B, Dickhaus T, Müller KR. Introduction to machine learning for brain imaging. NeuroImage. 2011 May;56(2):387–399.

43. Erickson BJ, Korfiatis P, Akkus Z, Kline TL. Machine Learning for Medical Imaging. RadioGraphics. 2017 Feb;37(2):505–515.
44. Kan A. Machine learning applications in cell image analysis. Immunol Cell Biol. 2017 Mar;.
45. Vidaurre C, Sannelli C, Müller KR, Blankertz B. Machine-Learning-Based Coadaptive Calibration for Brain-Computer Interfaces. Neural Computat. 2010 Dec;23(3):791–816.
46. Mala S, Latha K. Feature Selection in Classification of Eye Movements Using Electrooculography for Activity Recognition. Com Math Met Med. 2014 Dec;2014.
47. Mahmud M, Vassanelli S. Processing and Analysis of Multichannel Extracellular Neuronal Signals: State-of-the-Art and Challenges. Front Neurosci. 2016;10.
48. Lecun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015 5;521(7553):436–444.
49. Yahaya SW, Lotfi A, Mahmud M. A consensus novelty detection ensemble approach for anomaly detection in activities of daily living. Applied Soft Computing. 2019;83:105613.
50. Fabietti M, Mahmud M, Lotfi A, Averna A, Guggenmo D, Nudo R, et al. Neural Network-based Artifact Detection in Local Field Potentials Recorded from Chronically Implanted Neural Probes. In: Proc. IJCNN; 2020. p. 1–8.
51. Mahmud M, Kaiser MS, Hussain A, Vassanelli S. Applications of deep learning and reinforcement learning to biological data. IEEE transactions on neural networks and learning systems. 2018;29(6):2063–2079.
52. Mahmud M, Kaiser MS, Hussain A. Deep Learning in Mining Biological Data. arXiv:200300108 [cs, q-bio, stat]. 2020 Feb;p. 1–36. ArXiv: 2003.00108. Available from: `http://arxiv.org/abs/2003.00108`.
53. Dey N, Rajinikanth V, Fong SJ, Kaiser MS, Mahmud M. Social-Group-Optimization Assisted Kapur's Entropy and Morphological Segmentation for Automated Detection of COVID-19 Infection from Computed Tomography Images. Cogn Comput. 2020;p. 1–12. Available from: `https://doi.org/10.1007/s12559-020-09751-3`.
54. Aradhya MVN, Mahmud M, Guru D, S Agrawal B, Kaiser MS. One Shot Cluster based Approach for the Detection of COVID-19 from Chest X-Ray Images. Cognitive Computation. 2020;p. 1–8.
55. Noor MBT, Zenia NZ, Kaiser MS, Mahmud M, Al Mamun S. Detecting Neurodegenerative Disease from MRI: A Brief Review on a Deep Learning Perspective. In: Liang P, Goel V, Shan C, editors. Brain Informatics. Cham: Springer International Publishing; 2019. p. 115–125.
56. Ali HM, Kaiser MS, Mahmud M. Application of Convolutional Neural Network in Segmenting Brain Regions from MRI Data. In: Liang P, Goel V, Shan C, editors. Brain Informatics. Cham: Springer International Publishing; 2019. p. 136–146.
57. Miah Y, Prima CNE, Seema SJ, Mahmud M, Kaiser MS. Performance Comparison of Machine Learning Techniques in Identifying Dementia from Open Access Clinical Datasets. In: Proc. ICACIn. Springer, Singapore; 2020. p. 69–78.
58. Watkins J, Fabietti M, Mahmud M. SENSE: a Student Performance Quantifier using Sentiment Analysis. In: Proc. IJCNN; 2020. p. 1–6.
59. Rabby G, Azad S, Mahmud M, Zamli KZ, Rahman MM. TeKET: a Tree-Based Unsupervised Keyphrase Extraction Technique. Cogn Comput. 2020;12(5):811–833.
60. Orojo O, Tepper J, McGinnity TM, Mahmud M. A Multi-recurrent Network for Crude Oil Price Prediction. In: Proc. SSCI; 2019. p. 2940–2945.
61. Ching T, et al . Opportunities and Obstacles for Deep Learning in Biology and Medicine. bioRxiv. 2017;p. 142760.
62. Bengio Y. Learning Deep Architectures for AI. Found Trends Mach Learn. 2009 Jan;2(1):1–127.
63. Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge, USA: MIT Press; 2016.
64. Saxe AM, McClelland JL, Ganguli S. Exact solutions to the nonlinear dynamics of learning in deep linear neural nets. In: Proc. ICLR; 2014. p. 1–22.
65. Schmidhuber J. Deep Learning in neural networks: An overview. Neural Netw. 2015;61:85–117.
66. Zeng D, Zhao F, Shen W, Ge S. Compressing and Accelerating Neural Network for Facial Point Localization. Cogn Comput. 2018 Apr;10(2):359–367.
67. Salakhutdinov R, Hinton GE. Deep Boltzmann Machines. In: Proc. AISTATS2009; 2009. p. 448–455.
68. Geman S, Geman D. Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans Pattern Anal Mach Intell. 1984;6(6):721–741.
69. Fischer A, Igel C. An Introduction to Restricted Boltzmann Machines. In: Proc. CIARP 2012; 2012. p. 14–36.
70. Desjardins G, Courville AC, Bengio Y. On Training Deep Boltzmann Machines. CoRR. 2012;abs/1203.4416.
71. Tieleman T. Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient. In: Proc. ICML; 2008. p. 1064–1071.
72. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS. Deep learning for visual understanding: A review. Neurocomputing. 2016;187:27–48.
73. Hinton GE, Osindero S, Teh YW. A Fast Learning Algorithm for Deep Belief Nets. Neural Comput. 2006 Jul;18(7):1527–1554.
74. Bi X, Zhao X, Huang H, Chen D, Ma Y. Functional Brain Network Classification for Alzheimer's Disease Detection with Deep Features and Extreme Learning Machine. Cogn Comput. 2019 Nov;.
75. Ravi D, Wong C, Deligianni F, Berthelot M, et al. Deep Learning for Health Informatics. IEEE J Biomed Health Inform. 2017 Jan;21(1):4–21.
76. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. J Mach Learn Res. 2010;11:3371–3408.
77. Baldi P. Autoencoders, Unsupervised Learning and Deep Architectures. In: Proc. ICUTLW; 2012. p. 37–50.
78. Ranzato M, Poultney C, Chopra S, Cun YL. Efficient Learning of Sparse Representations with an Energy-based Model. In: Proc. NIPS; 2006. p. 1137–1144.
79. Kingma DP, Welling M. Auto-Encoding Variational Bayes. CoRR. 2014;abs/1312.6114.
80. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y. Contractive auto-encoders: explicit invariance during feature extraction. In: Proc. ICML; 2011. p. 833–840.
81. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. In: Advances in neural information processing systems; 2014. p. 2672–2680.
82. Isola P, Zhu JY, Zhou T, Efros AA. Image-to-image translation with conditional adversarial networks. In: Proc.IEEE CVPR; 2017. p. 1125–1134.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

83. Wang Z, Healy G, Smeaton AF, Ward TE. Use of Neural Signals to Evaluate the Quality of Generative Adversarial Network Performance in Facial Image Generation. Cogn Comput. 2020 Jan;12(1):13–24.
84. Pascanu R, Gulcehre C, Cho K, Bengio Y. How to Construct Deep Recurrent Neural Networks. In: Proc. ICLR; 2014. p. 1–13.
85. Elman JL. Finding Structure in Time. Cognitive Sci. 1990 Mar;14(2):179–211.
86. Schuster M, Paliwal KK. Bidirectional recurrent neural networks. IEEE Tran Signal Proces. 1997 Nov;45(11):2673–2681.
87. Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput. 1997 Nov;9(8):1735–1780.
88. Lipton ZC, Berkowitz J, Elkan C. A Critical Review of Recurrent Neural Networks for Sequence Learning. CoRR. 2015 May;CoRR: 1506.00019.
89. Ma Y, Peng H, Khan T, Cambria E, Hussain A. Sentic LSTM: a Hybrid Network for Targeted Aspect-Based Sentiment Analysis. Cogn Comput. 2018 Aug;10(4):639–650.
90. Wiatowski T, Bölcskei H. A mathematical theory of deep convolutional neural networks for feature extraction. IEEE Trans Inf Theory. 2017;64(3):1845–1866.
91. LeCun Y, Bengio Y. Convolutional Networks for Images, Speech, and Time Series. In: Arbib MA, editor. The Handbook of Brain Theory and Neural Networks. Cambridge, MA, USA: MIT Press; 1998. p. 255–258.
92. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Proc. NIPS; 2012. p. 1097–1105.
93. Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR. 2014;abs/1409.1556.
94. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proc. CVPR2015; 2015. p. 1–9.
95. Heinsfeld AS, Franco AR, Craddock RC, Buchweitz A, Meneguzzi F. Identification of autism spectrum disorder using deep learning and the ABIDE dataset. NeuroImage: Clin. 2018;17:16 – 23.
96. Kuang D, He L. Classification on ADHD with Deep Learning. In: Proc. CCBD; 2014. p. 27–32.
97. HosseiniAsl E, Gimelfarb GL, El-Baz A. Alzheimer's Disease Diagnostics by a Deeply Supervised Adaptable 3D Convolutional Network. CoRR. 2016;abs/1607.00556.
98. Suk HI, Lee SW, Shen D. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. NeuroImage. 2014;101:569 – 582.
99. Li F, Tran L, Thung KH, Ji S, Shen D, Li J. A Robust Deep Model for Improved Classification of AD/MCI Patients. IEEE J Biomed Health Inform. 2015 Sep;19(5):1610–1616.
100. Havaei M, Guizard N, Larochelle H, Jodoin PM. Deep Learning Trends for Focal Brain Pathology Segmentation in MRI. In: Holzinger A, editor. Machine Learning for Health Informatics: State-of-the-Art and Future Challenges. Cham: Springer; 2016. p. 125–148.
101. Fritscher K, Raudaschl P, Zaffino P, Spadea MF, Sharp GC, et al. Deep Neural Networks for Fast Segmentation of 3D Medical Images. In: Proc. MICCAI; 2016. p. 158–165.
102. Iqbal T, Ali H. Generative Adversarial Network for Medical Images (MI-GAN). J Med Syst. 2018 Oct;42(11):231.
103. Ciresan D, Giusti A, Gambardella L, Schmidhuber J. Deep Neural Nets Segment Neuronal Membrane in Electron Microscopy Images. In: Proc. NIPS; 2012. p. 2843–2851.
104. Stollenga MF, Byeon W, Liwicki M, Schmidhuber J. Parallel Multi-Dimensional LSTM, With Application to Fast Biomedical Volumetric Image Segmentation. In: Proc. NIPS; 2015. p. 2980–88.
105. Kleesiek J, Urban G, Hubert A, Schwarz D, Maier-Hein K, Bendszus M, et al. Deep MRI brain extraction: A 3D convolutional neural network for skull stripping. NeuroImage. 2016;129:460 – 469.
106. Cho J, Lee K, Shin E, Choy G, Do S. Medical Image Deep Learning with Hospital PACS Dataset. CoRR. 2015;abs/1511.06348.
107. Ngo T, et al. Combining deep learning and level set for the automated segmentation of the left ventricle of the heart from cardiac cine mr. Med Image Anal. 2017;35:159–171.
108. Ciresan D, Giusti A, Gambardella L, Schmidhuber J. Mitosis detection in breast cancer histology images with deep neural networks. In: Proc. MICCAI; 2013. p. 411–4188.
109. Kamnitsas K, Ledig C, Newcombe VFJ, Simpson J, et al . Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation. Med Image Anal. 2017;36:61–78.
110. Lu N, Li T, Ren X, Miao H. A Deep Learning Scheme for Motor Imagery Classification based on Restricted Boltzmann Machines. IEEE Trans Neural Syst Rehabil Eng. 2016;PP(99):1–1.
111. Yang H, Sakhavi S, Ang KK, Guan C. On the use of convolutional neural networks and augmented CSP features for multi-class motor imagery of EEG signals classification. In: Proc. 37th IEEE EMBC; 2015. p. 2620–2623.
112. Tabar YR, Halici U. A novel deep learning approach for classification of EEG motor imagery signals. J Neural Eng. 2017;14(1):016003.
113. Sakhavi S, Guan C, Yan S. Parallel convolutional-linear neural network for motor imagery classification. In: Proc. EUSIPCO; 2015. p. 2786–2790.
114. Li K, Li X, Zhang Y, Zhang A. Affective state recognition from EEG with deep belief networks. In: Proc. BIBM; 2013. p. 305–310.
115. Jia X, Li K, Li X, Zhang A. A Novel Semi-Supervised Deep Learning Framework for Affective State Recognition on EEG Signals. In: Proc. IEEE BIBE; 2014. p. 30–37.
116. Tripathi S, Acharya S, Sharma R, Mittal S, et al. Using Deep and Convolutional Neural Networks for Accurate Emotion Classification on DEAP Dataset. In: Proc. 29th IAAI; 2017. p. 4746–4752.
117. Chen G, Zhu Y, Hong Z, Yang Z. EmotionalGAN: Generating ECG to Enhance Emotion State Classification. In: Proc. 2019 Int. Conf. Artif. Intell. Comput. Sci. AICS 2019. New York, NY, USA: Association for Computing Machinery; 2019. p. 309–313.
118. Mirowski P, Madhavan D, LeCun Y, Kuzniecky R. Classification of patterns of EEG synchronization for seizure prediction. Clin Neurophysiol. 2009;120(11):1927 – 1940.
119. Jirayucharoensak S, Pan-Ngum S, Israsena P. EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation. Scientific World J. 2014;p. 1–10.
120. Wu Z, Ding X, Zhang G. A Novel Method for Classification of ECG Arrhythmias Using Deep Belief Networks [Journal Article]. J Comp Intel Appl. 2016;15:1650021.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

121. Yan Y, Qin X, Wu Y, Zhang N, Fan J, et al. A restricted Boltzmann machine based two-lead electrocardiography classification. In: Proc. BSN; 2015. p. 1–9.
122. Atzori M, Cognolato M, Müller H. Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands. Front Neurorobot. 2016;10:9.
123. Huanhuan M, Yue Z. Classification of Electrocardiogram Signals with DBN. In: Proc. IEEE CSE; 2014. p. 7–12.
124. Wang S, Peng J, Ma J, Xu J. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. Scientific Reports. 2016 Nov;6(1).
125. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nature Biotechnol. 2015;33(8):831–838.
126. Chen G, Tsoi A, Xu H, Zheng WJ. Predict effective drug combination by deep belief network and ontology fingerprints. J Biomed Inform. 2018;85:149 – 154.
127. Denas O, Taylor J. Deep modeling of gene expression regulation in an Erythropoiesis model. In: Proc. ICMLRL; 2013. p. 1–5.
128. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 2016;26(7):990–9.
129. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. Nature Methods. 2015 Aug;12(10):931–934.
130. Marouf M, et al. Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. Nat Commun. 2020;11:166.
131. Lee T, Yoon S. Boosted Categorical Restricted Boltzmann Machine for Computational Prediction of Splice Junctions. In: Proc. ICML; 2015. p. 2483–2492.
132. Zeng H, Edwards MD, Liu G, Gifford DK. Convolutional neural network architectures for predicting DNA-protein binding. Bioinformatics. 2016;32(12):121–127.
133. Park S, Min S, Choi H, Yoon S. deepMiRGene: Deep Neural Network based Precursor microRNA Prediction. CoRR. 2016;abs/1605.00017.
134. Lee B, Baek J, Park S, Yoon S. deepTarget: End-to-end Learning Framework for miRNA Target Prediction using Deep Recurrent Neural Networks. CoRR. 2016;abs/1603.09123.
135. Li H. A Template-Based Protein Structure Reconstruction Method Using DA Learning. J Proteomics Bioinform. 2016;9(12).
136. Ibrahim R, Yousri NA, Ismail MA, El-Makky NM. Multi-level gene/MiRNA feature selection using deep belief nets and active learning. In: Proc. IEEE EMBC; 2014. p. 3957–3960.
137. Chen L, Cai C, Chen V, Lu X. Trans-species learning of cellular signaling systems with bimodal deep belief networks. Bioinformatics. 2015 sep;31(18):3008–3015.
138. Danaee P, Ghaeini R, Hendrix DA. A Deep Learning Approach For Cancer Detection And Relevant Gene Identification. In: Proc. Pac. Symp. Biocomput.. vol. 22; 2016. p. 219–229.
139. Li Y, Fauteux F, Zou J, Nantel A, Pan Y. Personalized prediction of genes with tumor-causing somatic mutations based on multi-modal deep Boltzmann machine. Neurocomputing. 2019;324:51 – 62.
140. Zhang T, Zhang L, Payne PRO, Li F. Synergistic Drug Combination Prediction by Integrating Multiomics Data in Deep Learning Models. arXiv:181107054 [cs, q-bio, stat]. 2018 Nov;ArXiv: 1811.07054. Available from: `http://arxiv.org/abs/1811.07054`.
141. Huang Y, Gulko B, Siepel A. Fast, scalable prediction of deleterious noncoding variants from functional and population genomic data. Nature Genet. 2017;49:618–624.
142. Le EPV, Wang Y, Huang Y, Hickman S, Gilbert FJ. Artificial intelligence in breast imaging. Clin Radiol. 2019;74(5):357 – 366.
143. Yi X, Walia E, Babyn P. Generative adversarial network in medical imaging: A review. Med Image Anal. 2019;58:101552.
144. Sandfort V, Yan K, Pickhardt PJ, Summers RM. Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks. Sci Rep. 2019 Nov;9(1):1–9. Number: 1 Publisher: Nature Publishing Group.
145. Armanious K, et al. MedGAN: Medical image translation using GANs. Comput Med Imaging Graph. 2020;79:101684.
146. Uemura T, et al. GAN-based survival prediction model from CT images of patients with idiopathic pulmonary fibrosis. In: Chen PH, Deserno TM, editors. Medical Imaging 2020: Imaging Informatics for Healthcare, Research, and Applications. vol. 11318. SPIE; 2020. p. 354 – 359. Backup Publisher: International Society for Optics and Photonics.
147. Thambawita V, Hammer HL, Riegler M, Halvorsen P. GANEx: A complete pipeline of training, inference and benchmarking GAN experiments. In: 2019 CBMI. IEEE; 2019. p. 1–4.
148. Halicek M, et al. Conditional generative adversarial network for synthesizing hyperspectral images of breast cancer cells from digitized histology. In: Tomaszewski JE, Ward AD, editors. Medical Imaging 2020: Digital Pathology. vol. 11320. SPIE; 2020. p. 198 – 205.
149. Zhu F, Ye F, Fu Y, Liu Q, Shen B. Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network. Sci Rep. 2019 May;9(1):1–11. Number: 1 Publisher: Nature Publishing Group.
150. Yu L, Zhang W, Wang J, Yu Y. Seqgan: Sequence generative adversarial nets with policy gradient. In: Proc. 31st AAAI Conf. AI; 2017. p. 2852–2858.
151. Ye F, Zhu F, Fu Y, Shen B. ECG Generation With Sequence Generative Adversarial Nets Optimized by Policy Gradient. IEEE Access. 2019;7:159369–159378. Conference Name: IEEE Access.
152. Luo Y, Lu BL. EEG Data Augmentation for Emotion Recognition Using a Conditional Wasserstein GAN. In: 2018 IEEE EMBC; 2018. p. 2535–2538. ISSN: 1558-4615.
153. You S, et al. Unsupervised automatic seizure detection for focal-onset seizures recorded with behind-the-ear EEG using an anomaly-detecting generative adversarial network. Comput Methods Programs Biomed. 2020;p. 105472.
154. Jiao Y, Deng Y, Luo Y, Lu BL. Driver sleepiness detection from EEG and EOG signals using GAN and LSTM networks. Neurocomputing. 2020;.
155. Singh P, Pradhan G. A New ECG Denoising Framework Using Generative Adversarial Network. IEEE/ACM Trans Comput Biol Bioinform. 2020;p. 3114–3128.
156. Wang X, Ghasedi Dizaji K, Huang H. Conditional generative adversarial network for gene expression inference. Bioinformatics. 2018 09;34(17):i603–i611.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

157. Pan X, Shen HB. RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. BMC Bioinform. 2017;18(1).
158. Jiang X, Zhao J, Qian W, Song W, Lin GN. A Generative Adversarial Network Model for Disease Gene Prediction With RNA-seq Data. IEEE Access. 2020;8:37352–37360.
159. Zhao L, Wang J, Pang L, Liu Y, Zhang J. GANsDTA: Predicting Drug-Target Binding Affinity Using GANs. Front Genet. 2020;10:1243.
160. Editorial. Sharing data. Nat Cell Biol. 2009 11;11(11):1273.
161. Lord PW, et al . Large-Scale Data Sharing in the Life Sciences: Data Standards, Incentives, Barriers and Funding Models (The 'Joint Data Standards Study'); 2005. Available from: http://www.nesc.ac.uk/technical_papers/UKeS-2006-02.pdf.
162. Martone ME, Ellisman MH, Sosinsky GE, Gupta A, Tran J, Wong W, et al. Cell Centered Database. UC San Diego Library Digital Collections. 2017;Https://doi.org/10.6075/J0S180PX.
163. Ellisman M, et al. Cell Image Library; 2016. (Accessed on: 04/01/2020). Available from: http://www.cellimagelibrary.org/.
164. ERIC. EuroBioimaging; 2016. (Accessed on: 04/01/2020). Available from: http://www.eurobioimaging.eu/.
165. Karkow W. HAPS Histology Image Database; 2008. (Accessed on: 23/01/2017). Available from: http://hapshistology.wikifoundry.com/.
166. of Dundee U. IDR: Image Data Resource; 2016. Available from: https://idr.openmicroscopy.org/.
167. Kistler M. SMIR Full Body CT. SMIR. 2017;Available from: www.doi.org/10.22016/smir.o.214315.
168. of Arkansas for Medical Sciences U. The Cancer Imaging Archive; 2015. (Accessed on: 04/01/2020). Available from: https://www.cancerimagingarchive.net/.
169. Marinelli RJ, et al. The Stanford Tissue Microarray Database; 2007. (Accessed on: 23/01/2017). Available from: http://tma.stanford.edu.
170. University of California SB. UCSB Bio-Segmentation Benchmark dataset; 2008. (Accessed on: 23/01/2017). Available from: https://bioimage.ucsb.edu/research/bio-segmentation.
171. ABIDE. Autism Brain Imaging Data Exchange; 2012. (Accessed on: 04/01/2020). Available from: https://goo.gl/n694sN.
172. Milham MP. ADHD200; 2011. (Accessed on: 04/01/2020). Available from: http://fcon_1000.projects.nitrc.org/indi/adhd200/.
173. ANDI. Alzheimer's Disease Neuroimaging Initiative (ADNI datasets; 2009. (Accessed on: 04/01/2020). Available from: http://adni.loni.usc.edu/.
174. López M. Breast Cancer Digital Repository; 2008. (Accessed on: 08/04/2020). Available from: https://bcdr.eu/.
175. Mooney P. Chest X-Ray Images (Pneumonia) | Kaggle; 2018. (Accessed on: 04/01/2020). Available from: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.
176. MITOS-ATYPIA. MITOS-ATYPIA-14 - Dataset; 2012. (Accessed on: 04/01/2020). Available from: https://mitos-atypia-14.grand-challenge.org/dataset/.
177. NAMIC. MIDAS - Community National Alliance for Medical Image Computing (NAMIC); 2010. Available from: http://hdl.handle.net/1926/457.
178. Cohen JP, Morrison P, Dao L. COVID-19 image data collection; 2020. (Accessed on: 04/01/2020). Available from: https://github.com/ieee8023/covid-chestxray-dataset.
179. Yarkoni T. Neurosynth; 2012. (Accessed on: 04/01/2020). Available from: http://neurosynth.org/.
180. of Health (NIH) NI. NIH chest x-ray datasets; 2017. (Accessed on: 04/01/2020). Available from: https://nihcc.app.box.com/v/ChestXray-NIHCC.
181. LaMontagne PJ, et al. Open Access Series of Imaging Studies (OASIS); 2019. (Accessed on: 04/01/2020). Available from: http://www.oasis-brains.org/.
182. Muschelli J. Open Neuroimaging Datasets; 2015. (Accessed on: 04/01/2020). Available from: https://goo.gl/azm4XW.
183. Reyes M. The HEAR-EU Multiscale Imaging and Modelling Dataset of the Human Inner Ear. SMIR. 2017;Available from: www.doi.org/10.22016/smir.o.204388.
184. Dataset I. Brain development Datasets; 2014. (Accessed on: 04/01/2020). Available from: http://brain-development.org/ixi-dataset/.
185. Shattuck DW, Mirza M, Adisetiyo V, Hojatkashani C, Salamon G, et al. Construction of a 3D probabilistic atlas of human cortical structures. NeuroImage. 2008 Feb;39(3):1064–1080. Available from: http://linkinghub.elsevier.com/retrieve/pii/S1053811907008099.
186. Gorgolewski KJ, et al. NeuroVault; 2015. (Accessed on: 04/01/2020). Available from: http://neurovault.org/.
187. Boekel W. Neuroimaging Informatics Tools and Resources Clearinghouse dataset; 2015. (Accessed on: 04/01/2020). Available from: https://goo.gl/CA2pkO.
188. Poldrack, et al. OPEN fMRI: A multi-subject, multi-modal human neuroimaging dataset; 2015. (Accessed on: 04/01/2020). Available from: https://openfmri.org/.
189. Pernet C, Gorgolewski K, Ian W. Neuroimaging dataset of brain tumour patients; 2016. (Accessed on: 04/01/2020). Available from: https://goo.gl/fmYYm4.
190. van Ginneken B, Kerkstra S, Meakin J. DRIVE - Grand Challenge; 2004. Available from: https://drive.grand-challenge.org/.
191. Repository IBS. NITRC: IBSR: Tool/Resource Info; 2007. (Accessed on: 04/01/2020). Available from: https://www.nitrc.org/projects/ibsr.
192. Goldbaum M. The STARE Project; 1975. Available from: https://cecas.clemson.edu/~ahoover/stare/.
193. Cao Z, Chuang M, King JT, Lin CT. Multi-channel EEG recordings during a sustained-attention driving task. Figshare. 2019;Collection. Available from: https://doi.org/10.6084/m9.figshare.6427334.v5.
194. Picone J. Temple University EEG Corpus; 2011. (Accessed on: 07/04/2020). Available from: https://www.isip.piconepress.com/projects/tuh_eeg/html/downloads.shtml.
195. GB M, RG M. MIT-BIH Arrhythmia Database; 1999. (Accessed on: 04/01/2020). Available from: https://doi.org/10.13026/C2F305.
196. Goldberger A, et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals; 2003. Circulation. 101(23):e215-e220. Available from: https://doi.org/10.13026/C28C71.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

197. Khamis H, Weiss R, Xie Y, Chang CW, Lovell NH, Redmond SJ. TELE ECG Database: 250 telehealth ECG records (collected using dry metal electrodes) with annotated QRS and artifact masks, and MATLAB code for the UNSW artifact detection and UNSW QRS detection algorithms; 2016. Available from: `https://doi.org/10.7910/DVN/QTG0EP`.
198. 2020 BH. BNCIHorizon2020; 2015. (Accessed on: 04/01/2020). Available from: `https://goo.gl/6gLj52`.
199. Khushaba RM. Electromyogram (EMG) Repository; 2012. (Accessed on: 06/04/2020). Available from: `https://www.rami-khushaba.com/electromyogram-emg-repository.html`.
200. Rantanen V, et al.. Mimetic Interfaces: Facial Surface EMG Dataset 2015; 2015. (Accessed on: 04/01/2020). Available from: `http://bit.ly/2npCAH8`.
201. Atzori M. NinaPro Database — Non-Invasive Adaptive Hand Prosthetics; 2012. (Accessed on: 04/01/2020). Available from: `https://www.idiap.ch/project/ninapro/database`.
202. Koelstra S, et al. Database for Emotion Analysis using Physiological Signals; 2011. (Accessed on: 04/01/2020). Available from: `http://www.eecs.qmul.ac.uk/mmv/datasets/deap/`.
203. Abadi MK, et al. MEG-based Multimodal Database for Decoding Affective Physiological Responses; 2007. (Accessed on: 04/01/2020). Available from: `http://mhug.disi.unitn.it/wp-content/DECAF/DECAF.html`.
204. HeadIT of University of California SD. Imagined Emotion; 2009. (Accessed on: 04/01/2020). Available from: `https://bit.ly/3bRj78T`.
205. Soleymani M, Lichtenauer J, Pun T, M P. HCI Tagging Database; 2012. (Accessed on: 04/01/2020). Available from: `https://mahnob-db.eu/hci-tagging/`.
206. Lu PBL. SEED Datasets; 2013. (Accessed on:04/01/2020). Available from: `http://bcmi.sjtu.edu.cn/~seed/seed.html`.
207. Kaya M, Binli MK, Ozbay E, Yanar H, Mishchenko Y. A large electroencephalographic motor imagery dataset for electroencephalographic brain computer interfaces. figshare. 2018;Collection. Available from: `https://doi.org/10.6084/m9.figshare.c.3917698.v1`.
208. Cho M H amd Ahn, Ahn S, Kwon M, C JS. Supporting data for EEG datasets for motor imagery brain computer interface. GigaScience Database. 2017;Available from: `http://dx.doi.org/10.5524/100295`.
209. Schalk G, McFarland DJ, Hinterberger T, Birbaumer N, Wolpaw JR. EEG Motor Movement/Imagery Dataset; 2009. Accessed on: 06/04/2020. Available from: `https://doi.org/10.13026/C28G6P`.
210. Korczowski L, Ostaschenko E, Andreev A, Cattan G, Rodrigues PC, Gautheret V, et al.. Brain Invaders calibration-less P300-based BCI using dry EEG electrodes Dataset (bi2014a); 2019. (Accessed on: 04/01/2020). Available from: `https://doi.org/10.5281/zenodo.3266223`.
211. Korczowski L, Cederhout M, Andreev A, Cattan G, Rodrigues PL, Gautheret V, et al.. Brain Invaders calibration-less P300-based BCI with modulation of flash duration Dataset (bi2015a); 2019. (Accessed on: 04/01/2020). Available from: `https://doi.org/10.5281/zenodo.3266930`.
212. Korczowski L, Ostaschenko E, Andreev A, Cattan G, Rodrigues PC, Gautheret V, et al.. Brain Invaders Solo versus Collaboration: Multi-User P300-based Brain-Computer Interface Dataset (bi2014b); 2019. (Accessed on: 04/01/2020). Available from: `https://doi.org/10.5281/zenodo.3267302`.
213. Korczowski L, Cederhout M, Andreev A, Cattan G, Rodrigues PL, Gautheret V, et al.. Brain Invaders Cooperative versus Competitive: Multi-User P300-based Brain-Computer Interface Dataset (bi2015b); 2019. (Accessed on: 04/01/2020). Available from: `https://doi.org/10.5281/zenodo.3266762`.
214. BCI Competitions. BCI Competition datasets; 2008. (Accessed on: 04/01/2020). Available from: `http://www.bbci.de/competition/`.
215. BCI Challenge NER2015. BCI Challenge @ NER 2015; 2015. (Accessed on: 06/04/2020). Available from: `https://kaggle.com/c/inria-bci-challenge`.
216. Broderick MPea. Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech, v3, Dryad; 2020. (Accessed on: 04/01/2020). Available from: `https://doi.org/10.5061/dryad.070jc`.
217. for Complex Physiologic Signals RR. Physionet; 1999. (Accessed on: 04/01/2020). Available from: `https://physionet.org/physiobank/database/`.
218. Aha D. UCI ML repository; 1987. (Accessed on: 04/01/2020). Available from: `https://archive.ics.uci.edu/ml/datasets.php`.
219. Congedo M, et al. " Brain Invaders": a prototype of an open-source P300-based video game working with the OpenViBE platform. In: Proc. BCI 2011; 2011. p. 280–283.
220. PubChem. PubChem Data Sources; 2020. (Accessed on: 04/01/2020). Available from: `https://pubchemdocs.ncbi.nlm.nih.gov/covid-19`.
221. PubChem. PubChem Data Sources; 2005. (Accessed on: 04/01/2020). Available from: `https://pubchem.ncbi.nlm.nih.gov/sources/`.
222. Biolab. Bioinformatics Laboratory; 1999. (Accessed on: 04/01/2020). Available from: `http://www.biolab.si/supp/bi-cancer/projections/`.
223. Pradhan S, et al.. Indian Genetic Disease Database; 2011. (Accessed on: 04/01/2020). Available from: `http://www.igdd.iicb.res.in/`.
224. Atlas TCG. The Cancer Genome Atlas Home Page [nci-Home]; 2005. (Accessed on: 04/01/2020). Available from: `https://cancergenome.nih.gov/`.
225. Network BDT. Berkeley Drosophila Transcription Network Project; 2001. Available from: `http://bdtnp.lbl.gov:8080/Fly-Net/`.
226. ENCODE. Encyclopedia of DNA Elements; 2003. (Accessed on: 04/01/2020). Available from: `https://genome.ucsc.edu/ENCODE/`.
227. NHLBI GO ESP. Exome Variant Server; 2011. (Accessed on: 06/04/2020). Available from: `https://evs.gs.washington.edu/EVS/`.
228. GEO. Gene Expression Omnibus; 2000. (Accessed on: 04/04/2020). Available from: `https://www.ncbi.nlm.nih.gov/geo/`.
229. Abreu M, et al. gnomAD; 2016. (Accessed on: 06/04/2020). Available from: `https://gnomad.broadinstitute.org/downloads`.
230. of MIT TBI, Harvar. GTEx Portal; 2012. (Accessed on: 04/01/2020). Available from: `https://www.gtexportal.org/home/`.
231. Rouillard AD, Gundersen GW, Fernandez NF, Wang Z, Monteiro CD, McDermott MG, et al. The harmonizome: a collection of processed datasets gathered to serve and mine knowledge about genes and proteins. Database. 2016 07;2016:baw100.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

232. INSDC. The International Nucleotide Sequence Database Collaboration; 2016. (Accessed on: 04/01/2020). Available from: http://www.insdc.org/.
233. Resource TIGS. 1000 Genomes Project; 2008. (Accessed on: 06/04/2020). Available from: https://www.internationalgenome.org/analysis.
234. JASPAR. JASPAR 2018: An open-access database of transcription factor binding profiles; 2008. (Accessed on: 04/01/2020). Available from: http://jaspar.genereg.net.
235. Consortium NREM. Roadmap Epigenomics Project - Data; 2007. (Accessed on: 06/04/2020). Available from: http://www.roadmapepigenomics.org/data/.
236. NSD. Nature Scientific data; 2014. (Accessed on: 04/01/2020). Available from: http://go.nature.com/2g6E1Vm.
237. SysGENSIM. SysGenSIM - Benchmark datasets; 2013. (Accessed on: 04/01/2020). Available from: http://sysgensim.sourceforge.net/datasets.html.
238. in Research BMSEB, Education. RCSB Protein Data Bank - RCSB PDB; 2015. (Accessed on: 04/01/2020). Available from: https://www.rcsb.org/pdb/home/home.do#Category-download.
239. Murzin AG, Brenner SE, Hubbard TJP, Chothia C. Structural Classification of Proteins database 2; 2020. (Accessed on: 10/04/2020). Available from: http://scop.mrc-lmb.cam.ac.uk/.
240. Fox NK, Brenner SE, Chandonia JM. Structural Classification of Proteins database - extended; 2018. (Accessed on: 10/04/2020). Available from: https://scop.berkeley.edu/.
241. Qian N, Sejnowski TJ. UCI Molecular Biology (UCI MB) Protein Secondary Structure Data Set; 1988. (Accessed on: 04/01/2020). Available from: https://archive.ics.uci.edu/ml/datasets/Molecular+Biology+(Protein+Secondary+Structure).
242. Schaefer CFAKKSBJDMHTBKH. NCI-Nature Pathway Interaction Database; 2009. (Accessed on: 10/04/2020). Available from: https://home.ndexbio.org/about-ndex/.
243. Kandasamy K, et al. NetPath; 2010. (Accessed on: 10/04/2020). Available from: http://www.netpath.org/.
244. Stein L, D'Eustachio P, Hermjakob H, Wu G. Reactome; 2010. (Accessed on: 10/04/2020). Available from: https://reactome.org/PathwayBrowser/.
245. Tran VD. miRBoost; 2015. (Accessed on: 04/01/2020). Available from: https://evryrna.ibisc.univ-evry.fr/evryrna/mirboost/mirboost_help.
246. SGD. Saccharomyces Genome Database; 2012. (Accessed on: 04/01/2020). Available from: https://www.yeastgenome.org/.
247. DNAD-J. DNA Databank of Japan; 1980. (Accessed on: 04/01/2020). Available from: http://www.ddbj.nig.ac.jp/.
248. ENA. European Nucleotide Archive; 1990. (Accessed on: 04/01/2020). Available from: http://www.ebi.ac.uk/ena.
249. GenBank. GenBank; 2013. (Accessed on: 04/01/2020). Available from: https://www.ncbi.nlm.nih.gov/genbank/.
250. Noordewier MO, Towell GG, Shavlik JW. UCI Molecular Biology (UCI MB) Splice-junction Gene Sequences Data Set; 1981. (Accessed on: 04/01/2020). Available from: https://archive.ics.uci.edu/ml/datasets/Molecular+Biology+(Splice-junction+Gene+Sequences).
251. UCI-MB. UCI Molecular Biology (UCI MB) Promoter Gene Sequences Data Set; 1985. (Accessed on: 04/01/2020). Available from: https://archive.ics.uci.edu/ml/support/Molecular+Biology+(Promoter+Gene+Sequences).
252. Manaswi NK. Understanding and working with Keras. In: Deep Learning with Applications Using Python. Springer; 2018. p. 31–43.
253. Kunkel R, et al. TensorSCONE: A Secure TensorFlow Framework using Intel SGX. CoRR. 2019;p. 1–12.
254. Sun X, Peng X, Ding S. Emotional Human-Machine Conversation Generation Based on Long Short-Term Memory. Cogn Comput. 2018 Jun;10(3):389–397.
255. Hao L, Liang S, Ye J, Xu Z. TensorD: A tensor decomposition library in TensorFlow. Neurocomputing. 2018;318:196 – 200.
256. Karpathy A. A Peek at Trends in Machine Learning; 2017. Available from: https://medium.com/@karpathy/a-peek-at-trends-in-machine-learning-ab8a1085a106.
257. Bahrampour S, Ramakrishnan N, Schott L, Shah M. Comparative Study of Deep Learning Software Frameworks. CoRR. 2016;abs/1511.06435. ArXiv: 1511.06435.
258. Shi S, et al . Benchmarking State-of-the-Art Deep Learning Software Tools. CoRR. 2016;abs/1608.07249.
259. Deepmark. THE Deep Learning Benchmarks; 2017. (Accessed on: 17- Dec- 2017). Available from: https://github.com/DeepMark.
260. Narang S. The source code and experimental data of Benchmarking State -of-the-Art Deep Learning Software Tools; 2017. (Accessed: 17/12/ 2017). Available from: http://dlbench.comp.hkbu.edu.hk/.
261. LeCun Y, Cortes C, Burges CJC. The MNIST database of handwritten digits; 1998. (Accessed on: 04/01/2020). Available from: http://yann.lecun.com/exdb/mnist/.
262. Zaremba W, Sutskever I, Vinyals O. Recurrent Neural Network Regularization. CoRR. 2014;abs/1409.2329.
263. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. CoRR. 2013;abs/1312.6229.
264. Murphy J. Deep Learning Benchmarks of NVIDIA Tesla P100 PCIe, Tesla K80, and Tesla M40 GPUs; 2017. (Accessed on: 04/01/2020). Available from: https://bit.ly/2WzFZ8x.
265. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, et al. Human-level control through deep reinforcement learning. Nature. 2015 Feb;518(7540):529–533.
266. Chollet F. The limitations of deep learning; 2017. (Accessed on: 12/12/2017). Available from: https://bit.ly/3bm9m24.
267. Zenil H, et al . An Algorithmic Information Calculus for Causal Discovery and Reprogramming Systems. bioRxiv. 2017;p. 185637.
268. Shwartz-Ziv R, Tishby N. Opening the Black Box of Deep Neural Networks via Information. CoRR. 2017 Mar;abs/1703.00810.
269. Nguyen AM, Yosinski J, Clune J. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Proc. CVPR; 2015. p. 427–436.
270. Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow IJ, et al. Intriguing properties of neural networks. In: CoRR. vol. abs/1312.6199; 2013. p. 1–10.
271. Baker NA, Klemm JD, Harper SL, Gaheen S, Heiskanen M, Rocca-Serra P, et al. Standardizing data. Nat Nanotechnol. 2013;8(2):73.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].

272. Wittig U, Rey M, Weidemann A, Müller W. Data management and data enrichment for systems biology projects. J Biotechnol. 2017;261:229–237.
273. Mahmud M, Rahman MM, Travalin D, Raif P, Hussain A. Service Oriented Architecture Based Web Application Model for Collaborative Biomedical Signal Analysis. Biomed Tech (Berl). 2012;57:780–783.
274. Mahmud M, Pulizzi R, Vasilaki E, Giugliano M. A Web-Based Framework for Semi-Online Parallel Processing of Extracellular Neuronal Signals Recorded by Microelectrode Arrays. In: Proc. MEAMEETING; 2014. p. 202–203.
275. Angelov P, Sperduti A. Challenges in Deep Learning. In: Proc. ESANN; 2016. p. 489–495.
276. Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep Reinforcement Learning: A Brief Survey. IEEE Signal Process Mag. 2017;34(6):26–38.

Cite this as: M Mahmud, MS Kaiser, TM McGinnity, A Hussain. (2020). Deep Learning in Mining Biological Data. Cogn Comput. doi: 10.1007/s12559-020-09773-x [epub ahead of print].