Received: 21 October 2022 | Revised: 13 December 2022 | Accepted: 21 December 2022 | Published online: 29 December 2022

RESEARCH ARTICLE

Artificial Intelligence and Applications
2023, Vol. 00(00) 1–18
DOI: 10.47852/bonviewAIA2202480

BON VIEW PUBLISHING

Recent Landscape of Deep Learning Intervention and Consecutive Clustering on Biomedical Diagnosis

Ayan Mukherji[1], Arindam Mondal[2], Rajib Banerjee[3] and Saurav Mallik[4,*]

[1]*Department of Information Technology, Haldia Institute of Technology, India*

[2]*Department of Electrical and Electronics Engineering, Pailan College of Management Technology, India*

[3]*Department of Electronics and Communication Engineering, Dr. B.C. Roy Engineering College, India*

[4]*Department of Environmental Health, Harvard T H Chan School of Public Health, USA*

Abstract: Consecutive clustering is one type of learning method that is built on neural network. It is frequently used in different domains including biomedical research. It is very useful for consecutive clustering (adjacent clustering). Adjacent clustering is highly used where there are various specific locations or addresses denoting each individual features in the data that need to be grouped consecutively. One of the useful consecutive clusterings in the field of biomedical research is differentially methylated region (DMR) finding analysis on various CpG sites (features). So far, many researches have been carried out on deep learning and consecutive clustering in biomedical domain. But for epigenetics study, very limited survey papers have been published till now where consecutive clustering has been demonstrated together. Hence, in this study, we contributed a comprehensive survey on several fundamental categories of consecutive clustering, for example, convolutional neural network, autoencoder, restricted Boltzmann machines and deep belief network, recurrent neural network, deep stacking networks, long short-term memory/gated recurrent unit network, etc., along with their applications, advantages, and disadvantages. Different forms of consecutive clustering algorithms that are covered in the second section (viz., supervised and unsupervised DMR finding methods) and used for DNA methylation data have been described here along with their advantages, shortcomings, and overall performance estimation (power, time). Our survey paper provides a latest research work that has been done for consecutive clustering algorithms for healthcare purposes. All the usages, benefits, and shortcomings along with their performance evaluation of each algorithm have been elaborated in this paper by which new biomedical researchers can understand and use those tools and algorithms for their research prospective.

Keywords: deep learning, DNA methylation, consecutive clustering, differentially methylated region (DMR), supervised and unsupervised DMR finding algorithm, power

1. Introduction

Nowadays, deep learning (DL) is considered to be one of the most prominent and emerging sub-fields of machine learning. It makes sense of data like sounds, texts, and photos by employing multilayered deep neural networks (DNNs) to construct abstraction from big-size data. The main features of DL are: (1) components for nonlinear processing with several layers and (2) each layer presents features of supervised or unsupervised learning [1]. The initial framework for DL was developed in the 1980s using artificial neural networks (ANNs) [2], but the credibility of DL's actual effects only emerged in 2006 [3]. Following that, a vast number of domains have adopted DL like image recognition, drug discovery, natural language processing (NLP), automatic speech recognition, and bioinformatics [4, 5].

Epigenetic procedure transforms expression of gene without altering the DNA gene's sequence. In the earlier time era of 1950s, the idea of epigenetic systems like intermediaries of cellular identity as well as cellular memory arose. Afterward, in the earlier era of 1970s, DNA methylation (DNAm) was suggested to be a transcriptional regulatory procedure that would be managed by cell division. Through associating an epigenetic event from DNAm, the designation of epigenetics turns into a molecular one. DNAm takes place while a methyl group is joined to the 5th carbon of cytosine residual which is combined through a phosphate to a guanine nucleotide (a CpG dinucleotide) through DNA methyl transfers (DNMT1, DNMT3A, and DNMT3B) [6]. DNAm varies with the position of the CpG site. CpG sites does not hold by many of the

*Corresponding author: Saurav Mallik, Department of Environmental Health, Harvard T H Chan School of Public Health, USA. Email: sauravmtech2@gmail.com, smallik@hsph.harvard.edu

genome. Anyway, collection of the CpG sites, defined "CpG islands," exist as well as generally spread the house-keeping genes promoters [7].

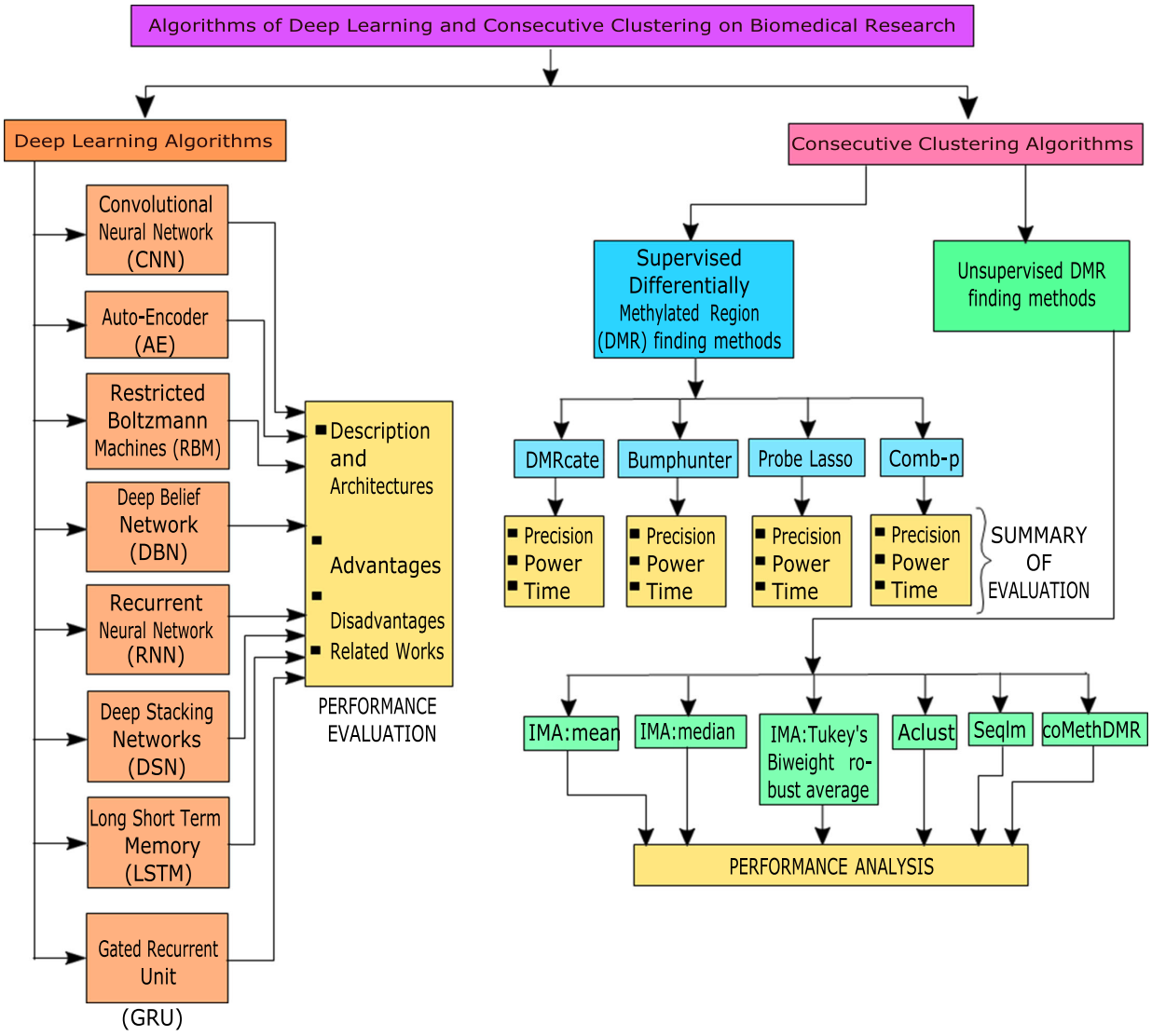
DNAm is a useful cellular procedure which is related to procedure such as X-chromosome deactivate as well as genomic marking. This has been associated with various diseases like cancer, schizophrenia, and diabetes [8, 9]. In the last few years, the purpose of methylation in several diseases has acquired from the research community to significant interest. One of the special consecutive clusterings or adjacent clusterings used for biomedical research is differentially methylated region (DMR) finding [10]. DMR finding has been pursued by two-step process: first, differentially methylated CpGs (DMCs) are identified by analogy of alignment results between samples; then, DMCs at neighbor positions are arranged as neighboring DMRs by specific distance criteria.

Furthermore, as compared to another overview works on the consecutive clustering [10, 11], our effort takes an extensive view of all sector and applications to which consecutive clustering has been applied. Specially, other works concentrate on the advancements as well as importance of a unique learning technique or modality, or to enhancements in a unique application [12]. Instead, in this paper, we basically observed a comprehensive interpretation of consecutive clustering applications and techniques toward highlight region, in which consecutive clustering still creates remarkable performance.

The remainder of this paper is as follows. We provide a quick overview of DL in Section 2. We describe the fundamentals of DNAm and DMR in Section 3 of this paper. In Section 4, we categorize various methods for identifying DMRs including different supervised and unsupervised DMR finding tools. In Section 5, we present comparative study of supervised and unsupervised consecutive clustering methods depending upon literature search and evidence. Finally, we offer our final thoughts and the potential future direction of this research in Section 6.

The flowchart of the survey paper is depicted in Figure 1.

Figure 1
The flowchart of Deep Learning Algorithms and Consecutive Clusterings



2. Overview of DL

DL is a large field of research that applies to both machine learning and artificial intelligence. Due to the following traits, DL mechanisms have reached their pinnacle of success in a wide range of application fields. New areas including imbalance problems, human activity recognition, transfer learning, and decision fusion class have achieved advancement in the accuracy and the performance. DL has impressive tools in different areas that strongly built on the neural networks with the inclusion of greater than two layers that could make use of big datasets efficiently and effectively. It supports pure learning ability and learns feature extraction techniques from the optimized outcomes, data, to fix highly computational tasks from top human ability. DL networks are built on the nature of the network structure, data representation, and activation methods, illustrate very different features in a less parameters, predictions performance could be greatly improved, and support secure and strong generalization efficiency with less of a requirement of training dataset. It is also stable than machine learning model in feature representation. DL networks do not depend on earlier data as well as knowledge with highest-level abstraction; these networks could derive complex features and better detection capacity techniques in the larger data era. We will now discuss the six fundamental types of DL architectures as well as associated research in many fields. A brief information of different DL methods are given on Table 1.

2.1. Convolutional neural network

Convolutional neural network (CNN) is a multilayered neural network based on the visual brain of animals. LeCun et al. [3] suggested the first CNN. Document analysis, picture recognition, and face recognition are the three main application areas of CNN.

As the architecture [1], early layers are used to analyze features like edges, while subsequent layers are used to combine features to create high level properties of the input that the classification is attempting to categorize. Pooling will then be done to lessen the extension of the acquired features. The following stage involves using convolution and then pooling, which are fed into a multilayer perceptron with only linked layers. The final layer, known as an output layer, is in charge of identifying the features of the image using backpropagation methods. CNN's deep processing layers, convolutional layer, pooling, and fully connected classification layer recognize a variety of applications, including face recognition, video identification, and distinct NLP tasks. Because of CNN's special features, including shared weights and local connectivity, the system operates more effectively and efficiently. Compared to other DL algorithms, it works substantially better. In comparison to other architectures, it is the one that is used the most.

2.1.1. CNN used for document analysis

Oleveira et al. [13] proposed document image layout analysis which uses a process that starts with the segmentation of document pictures into content blocks and ends with categorization. Each step is explained as follows.

Segmenting each document image page into its content blocks was the first stage completed. To locate locations with a high likelihood of having information, single pages are converted into grayscale images and subsequently improved using the running length technique described in Wong et al. [14]. Both horizontally and vertically, the method is applied, and the resulting binary pictures are joined with the AND operator. The generated binary picture is then subjected to two applications of a 3×3 dilation procedure to create content blobs. Finally, the largest connected component in the binary image is identified as a block of content by iteratively locating it. Up until no more related components are found in the image, the discovery process is repeated. A CNN model is utilized to categorize the document image's blocks of content into three different classes: text, tables, and images.

An architecture is constructed for the bidimensional baseline that takes a bidimensional image tile as input and creates it using a series of three 2D convolutional layers with 50 filters and ReLu [15] activation. Each bidimensional convolutional layer in this model is followed by a MaxPooling layer [16] with a 2 pixel kernel and a 0.1 dropout [17] for regularization. For improved model generalization, a 0.3 dropout is also recommended between the two fully connected layers.

In Oleveira et al. [13], a one-dimensional (1D) CNN architecture is suggested that uses picture tiles' vertical and horizontal projections to distinguish between various content blocks. For such projections, text, table, and image tiles have very distinctive and highly discriminative signatures. Due to text lines, text tiles typically have a roughly constant-signal-like form in the horizontal projection and a squared-signal-like shape in the vertical projection. Due to the structure of the columns, table tiles also have a roughly squared-signal form in the horizontal projection and a squared-signal shape in the vertical projection. In both horizontal and vertical projections, figures do not display any distinctive patterns. Two 1D arrays corresponding to the horizontal and vertical projections of a certain image tile are provided as input to the 1D CNN architecture. Each projection is created using a separate convolutional track that uses three 1D convolutional layers with 50 filters each and ReLu activation. The outputs from each track are combined and sent to a fully connected layer with 50 nodes connected to a fully connected layer with three nodes using softmax activation for categorical categorization of three classes. Each 1D convolutional layer in this model is followed by a MaxPooling layer for regularization, with a kernel size of 2 pixels and a 0.1 dropout. For greater generalization, there is a 0.3 dropout between the two fully connected layers, just like in the bidimensional model. The 3×1 pixel convolutional kernels were employed throughout all trials.

2.1.2. CNN used for image recognition

The automatic extraction, analysis, and comprehension of significant information from images are the subject of computer vision, a multidisciplinary topic of machine learning and artificial intelligence. With recent technological advancements, digital information, particularly in the form of photographs and movies, is rapidly expanding. Recognizing and analyzing images are a key challenge for computers in the field of computer vision when compared to humans. So, with human assistance, the classification of photographs will be carried out. The real-time picture datasets (MNIST digit images) are used by humans for training and testing purposes. The MNIST dataset provided as input is produced by the grayscale images. In the beginning, a classifier will be trained by a human to look for the necessary pattern in the images. Using the pattern that had been discovered in earlier stages, the images were then categorized. Regarding the patterns found, the outcomes will vary, and they entirely depend on the classification expert's knowledge. Krizhevsky et al. [18] employ different layers in a CNN to extract new features from picture datasets and have presented a DL architecture for image categorization.

In Ramprasath et al. [19], grayscale images of 28 × 28 pixels were used as input images. Thirty-two filters were applied to the input images in CCN's first layer, which resulted in 32 feature maps with a combined size of 26 × 26. The second layer uses 64 filters, each measuring 3 × 3, to create 64 feature maps, each measuring 24 × 24. By utilizing a sub-sampling window of size 2 × 2, the third layer, MaxPooling layer, is utilized to down sample the pictures to 12 × 12. Layer 4 is a fully connected layer with 128 neurons that classifies images using the sigmoid activation function to create the final image. In typical neural networks, each hidden layer is made up of a group of neurons that are individually connected to the corresponding hidden layer. The final layer of the network is completely connected and used to categorize images. Basically, the first hidden layer would have 28 × 28 × 1 = 784 weights if the input image was 28 × 28 × 1 (28 wide, 28 high, and 1 color channel). This number of weights appears still attainable. Longer images, 400 × 400 × 3, necessitate 400 × 400 × 3 = 4,80,000 weights; nevertheless, this completely connected layer does not scale very well. CNNs differ from conventional neural networks in that they accept input in the form of images of various sizes. Neurons are arranged in three dimensions—width, height, and depth—in the layers of a CNN. The word "depth" refers to the third dimension of an activation volume rather than the depth of a complete neural network, which can refer to the network's entire number of layers.

2.2. Autoencoder

One of the most popular unsupervised machine learning algorithms on the ANN is the autoencoder (AE) [20]. A trained AE will redesign the output close to the original input. The input layer, output layer, and hidden layer—which typically has a dimension less than the input layer—make up an AE. An AE's ability to find data structures by reducing data using nonlinear transformations is one of its advantages over principle component analysis [21]. Backpropagation is used in the approach, which is based on the encoder decoder paradigm, and the goal value is set to equal the input. To reproduce the original data, the input is first encoded into a lower-dimensional layer (decoder). After the layer has been trained, the output is passed on to the following layer to create a highly nonlinear dependence model on the input. The goal of this process is to reduce the size of the input data. The AE's middle-layer encoded layer is considered as an extracted feature for classification.

2.2.1. AE used for feature extraction

In Kunang et al. [22], preprocessing feature selection from a pool of 42 accessible features is the first step in this feature extraction procedure. One hot encoding is employed in this investigation. Another study used ordinal coding as a technique for variable encoding [23]. This method was selected because it produces better results than ordinal coding [24]. Here, all features even non-numerical ones are taken care of. Non-numerical features data in the dataset are transformed to numerical data. The variable on the map generates a binary dummy variable for each particular level. Scaling all the features is necessary so that some features do not outweigh the others. Feature scaling is the name of this process. Standardization, scaling, and normalization are three other extensively used alternatives to feature scaling. Z score normalization is used here. The majority of machine learning algorithms employ this method (such as SVM, logistic regression, and neural network) [25]. By calculating each feature's mean and standard deviation, normalized values are calculated. Dimension reduction using the AE model comes next. The AE model employed here uses a straightforward AE with 3 layers and 1 hidden layer. There are 120 input neurons used in the AE approach. The X transformation is carried out in the hidden layer. The features that were extracted based on the number of neurons are what output in the hidden layer is. The smaller dimensions of the features would be the data that have been extracted. Training and testing can be used to prepare the output.

2.3. Restricted Boltzmann machines and deep belief network

The restricted Boltzmann machine (RBM) [26] is an undirected graphical structure that consists of a visible layer, a hidden layer, and a balanced link between the levels. The hidden layer in RBM is not connected to an input in any way. The deep belief network (DBN) shows a multilayer network architecture that creates a special training technique with numerous hidden levels. Every connected layer pair is a RBM, often known as a stack of RBMs. Basic sensory information is provided by the input layer, and an abstract description of this information is provided by the hidden layer. The output layer's sole purpose is to carry out network classification. Unsupervised pretraining and supervised fine-tuning are the two steps in the training process. RBM can regenerate its input in unsupervised pretraining starting from the first hidden layer. Similar to the first RBM, the second one can be performed by using the outputs from the first hidden layer as the input and visible layer for the RBM. Every layer has prior training or experience. After the pretraining is finished, the supervised fine-tuning process starts. In this stage, the nodes that represent the output are identified with values or labels to aid in the learning process, and full network training is then carried out using the backpropagation algorithm or gradient descent learning.

2.3.1. RBM used for classification

In Koziol et al. [27], based on serum samples from newly diagnosed hepatocellular carcinoma (HCC) patients and healthy controls, the significance of an enlarged panel of 12 antibody profiles for cancer diagnosis of HCC is explored. Here, it is also applied the approach of RBMs [28, 29] to the classification problem.

2.3.2. DBN used for image compression

DBNs [30] can be thought of as a series of layers, each of which is made up of RBMs. The network is trained in this case layer by layer, with each layer attempting to determine how the input is distributed via unsupervised learning. Each layer functions as a hidden layer for the one that comes before it and as a visible layer for the layer that follows it, acting as an input for the subsequent layer. Each layer's nodes are linked to those in the one below them, resulting in a fully bipartite graph. For image recognition and image creation, DBNs are used. They can reduce the number of dimensions if the highest layer's count is low. DBNs do not perform well with neural networks that include stochastic or randomly initialized variables, but they are well suited for unsupervised learning. In order for the DBNs to adapt to the characteristics of the images/data that they are supposed to extract, they must go through a pre-learning phase. A method known as greedy learning was used to speed up this pre-learning stage. The DNN used in this application was trained using the MNIST database, an open-source dataset comprising

60,000 images. The fundamental idea behind greedy learning is to train each layer separately so that it can develop its input through a number of backpropagations. As a result, each layer's variables are changed in a certain way. Then, throughout the duration of the set, these variables are maintained frozen. The DBNs can be used to compress any image included in the dataset after they have been trained, offering flexibility in terms of compression ratio, compression time, and compression loss. In a DNN, each layer is made up of nodes, which are often computing units similar to brain neurons and which light up when a certain input flows through them. These nodes combine input data with weights or coefficients that, when multiplied with the input data, can amplify or corrupt the input. The sums of these products are then transferred through a node activation mechanism. The output of each layer then functions as the subsequent layer's input. Here, the weights of the edges create a weight matrix that is used to build the original raw data after the first DBN has compressed it. A deep autoencoder combines two DBNs—symmetrical DBNs with aggressive shallow layers for the decoding half and progressive shallow layers for the encoding half—into one. The frames of DBNs are layers made up of closed Boltzmann machines. They have a pair of encoders and decoders. Encoders in this instance take raw data as input and extract features from it. The decoder uses these extracted characteristics as input and reconstructs the data back to the original form as output. The encoder and decoder are built using numerous layers of RBMs. The training procedure happens in a layer by layer manner. Until the final criteria are satisfied, this training and feeding method is used.

2.4. Recurrent neural network

The fundamental network architecture is recurrent neural network (RNN) [31] which includes a wide range of architecture. Recurrent networks are advantageous because they have connections that, in contrast to complete feed-forward connections, can be used as feedback into earlier levels. It organizes the issues chronologically and captures the past memory of input. With conventional backpropagation, also known as backpropagation through time (BPTT) [32], these networks can be improved, trained, and expanded.

2.4.1. RNN used for speech recognition

Speech recognition aims to decipher the linguistic message in the form of text from the speech signal by analyzing the speech signal's sequence of sound units. In Venkateswarlu et al. [33], at input acquisition stage, following the capturing of the speech using a microphone .wav files are used to store data. With the use of the Praat object software program, the speech data are converted to an analog signal. The signal is then converted to a mono voice signal using a 11 kHz frequency.

The acoustic speech signal is still there as air pressure changes during the front-end processing stage. The microphone converts these pressure changes into a pressure-related electric current. These pressure changes are converted by the ear into a series of nerve impulses that are sent to the brain. The choice of features is very helpful in the work of voice recognition. To perform well for recognition, good features are required. The fundamental issue with voice recognition is identifying the right features for the purpose and devising a strategy to extract these features from the speech signal.

A database with 18 characters drawn from 4 primary sets and said 10 times by 6 speakers, including 3 men and 3 women of varying ages, serves as the source of data for the speech utterance (data collecting) step. The speaker-dependent data will be used during the training and testing phases. In speaker-dependent form, the network is trained using the first four utterances of each of the 18 characters spoken by each speaker, and it is tested using the remaining utterances. Therefore, the speech database contains 1080 utterances that may be used for testing as well as 1080 utterances that can be used for training the network.

In the preprocessing stage, the speech signals are captured using high-quality recording equipment in a low-noise setting and 11 kHz samples are used for the signals. When the input data are surrounded by quiet, isolated word recognition can get reasonable results.

In sampling stage, 150 samples are chosen at a sampling rate of 11 kHz, which is sufficient to represent all speech sounds.

In the windowing stage, we window each frame to improve the correlation of the Mel-frequency cepstral coefficients (MFCCs) in order to reduce the discontinuity of a signal at the start and end of each frame. MFCCs are frequently employed as features in speech recognition systems, such as those that can automatically identify telephone digits as they are spoken. Measures are taken by the MFCC between successive frames [34]. The signal should be reduced to zero or close to zero to minimize discontinuities at the ends of speech segments and so lessen the mismatch. The Praat object software tool selects a window length of 0.015 for the 12 Mel-frequency coefficients that are provided and for time 0.005 seconds.

The feature extraction stage entails assessing representations of the voice signal that are sensitive to linguistic content yet broad to acoustic change. When using a bank of filters in the frequency domain, the Mel-filter is used to detect band filtering. On a curvilinear frequency scale, the filter functions employed have a triangle form. The lower frequency, the center frequency, and the higher frequency are the three factors that influence the filter function. The separations between the lower and center frequencies and the higher and center frequencies on a Mel scale are equal.

RNN architecture has actually been utilized for visual pattern recognition; however in this case, BPTT is being used as the learning method to use RNN architecture for speech recognition, specifically for English speech recognition. By applying the backpropagation method, it has also been shown that this architecture performs more accurately in phoneme recognition than multilayer perceptron [35]. The BPTT algorithm relies on folding the network over time to convert it from a feedback system to a completely feed-forward system. Since the feedback connections must be adjusted so that they are feed-forward connections from one network to the following network if the network is to provide a signal that is time steps long. If the network is a single, substantial feed-forward network, the updated weights can then be treated as shared weights during training.

2.5. Deep stacking networks

Deep convex networks is another name for deep stacking networks (DSNs) [36, 37]. Compared to other traditional DL frameworks, DSN is unique. It is referred to as deep because it contains numerous deep separate networks, each of which has its own hidden layers. The DSN believes that training involves a variety of distinct unique training challenges rather than being a single, isolated issue. The DSN is made up of a fusion of modules that are situated in the architecture and are a component of the network. The DSN has three components that each performs a specific task. A single hidden zone, an output zone, and an input

zone are present in every module of the model. The input to each module is taken from the outputs of the preceding layer and the real input vector, with the subroutines stacked one on top of the other. Every module in DSN is trained separately to ensure its effectiveness and efficiency as well as its capacity for cooperative work. Each module, rather than the entire network, uses the supervised method of training for backpropagation. DSNs are a relevant and desirable network design since they perform better than normal DBNs.

2.5.1. DSN used for information retrieval

In Deng et al. [38], compared to voice and image classification tasks, regularization in DSN learning is found to be considerably more beneficial for information retrieval. The low dimensionality of the output vectors associated with each module in the DSN causes problems in information retrieval. Stacking information from a lower module of the DSN to its upper module is reduced by the low dimensionality, as opposed to speech tasks where there are typically many more classes that need to be detected. Here, significant DSN studies are carried out on a sponsored web information retrieval job. Commercial search engines provide additional sponsored results in addition to the organic web search results for the user's query. Advertisers who place bids to have their advertising appear on the search result pages build a database from which the sponsored results are picked. The process of finding relevant adverts for a query is the same as regular online search. Here, a DSN model of advertisement relevancy is displayed to help improve the sponsored search system. Given a search query, this model is trained to distinguish between relevant and irrelevant adverts and to provide an ad's relevance score.

The baseline for this DSN-based IR system is LambdaRank [39]. These algorithms produce their targets using annotated data that have been judged by experts to be relevant or irrelevant for each query-ad pair. Text features and user click features make up the two main categories of the ranking features utilized in network models. Here, a set of text features have been suggested in Hillard et al. [40] and Hillard et al. [41]. They also include features for query length (the amount of characters and words) and three sets of features for text matching, which compare the query text to each of the three text streams of an advertisement. These two categories of user click attributes were both extracted from click through logs. Click through features are the first category. In Gao et al. [42], a click stream is generated for each advertisement, consisting of a list of searches with clicks on the advertisement. A set of 30 features are then extracted by comparing the click stream to the input query. A set of translation probabilities between query and ad based on translation models read on the query-ad pairings obtained from click through logs is the second type of click feature employed here [43].

2.6. Long short-term memory/gated recurrent unit network

Hochreiter and Schimdhuber worked to describe the long short-term memory (LSTM) [44], which is used in a variety of contexts. IBM chose LSTMs that are primarily employed in voice recognition. A memory component known as a cell that can retain its value for a sufficient amount of time and considers it as a function of its input is benefited by the LSTM. This helps the unit remember the most recent calculation's result. Three ports, referred to as gates, make up a memory unit or cell. These ports control the movement of information into and out of the cell. The gate or input port regulates how new data enter the memory. The second gate, known as the forgets port, is used when an existing piece of information is forgotten and aids the cell in gathering the fresh information. The output gate's job is to once more control the information that is contained in the cell and used as the cell's output. The cell's weight can be used for controlling purposes. There is a need for the training technique, known as BPTT, which increases weight. The optimization process of the method requires network output error. The update gate and reset gate are two gates that are combined to form the gated recurrent unit (GRU) [45]. An update gate's purpose is to alert the user when the previous cell's contents are needed for maintenance. The shifting of previous cell contents with fresh input is defined by the reset gate. By initializing the reset gate to 1 and the update gate to 0, the GRU simulates a typical RNN. Compared to the LSTM, the GRU model's operational functionality is simple. It is thought to be more effective in terms of execution and can become skilled quickly.

2.6.1. LSTM used for handwriting recognition

In Carbune et al. [46], the architecture is comparable to that which is frequently used for acoustic designing in voice recognition [47], thus, it is referred to as a CLDNN (convolutions, LSTMs, and DNNs). The model takes a time series of length T as input and runs it through several bidirectional LSTM layers [48] which interpreted the character structure. After passing the results of the final LSTM layer through a softmax layer, a series of probability distributions over characters are produced at each time step. The softmax outputs are combined with character-based language models, word-based language models, and knowledge of language-specific characters for connectionist temporal classification decoding using beam search [49].

LSTMs have grown to be one of the most popular RNN cells due to their simplicity in training and improved performance [50]. The input sequence is processed both forward and backward in this experiment, and the output states of each layer are combined before being fed to the next layer using bidirectional LSTMs. For each script, the precise number of layers and nodes is determined empirically. A softmax layer receives the output of the LSTM layers at each time step to determine a probability distribution across the script's potential characters. The prior information relevant to a language is combined with the softmax layer logits. A weight (referred to as a "decoder weight" in the following) is captured for each of these extra information sources, and they are linearly combined. The combination that was learned is used as described in Graves et al. [51, 52] to model the beam search during decoding. It is possible to train one recognition model per script using a combination of different knowledge sources and then use that model to deliver numerous languages.

2.6.2. GRU used for speech recognition

The model proposed in Ravanelli et al. [53] is a revised version of the GRUs. The main modifications concern reset gate and ReLU activations. When there are several discontinuities in the sequence, the reset gate can be crucial. When switching from one text to another that is not semantically relevant, this may happen in language design. In these circumstances, it is wise to clear the stored memory to avoid making a choice that is influenced by an unrelated past. Yet, for some tasks such functionality might not be important. For instance, in Zhou et al. [45] discarding reset gate vector from the GRU model

has led to a single-gate architecture called minimal gated recurrent unit (M-GRU), which accomplishes a performance comparable to that achieved by standard GRUs in handwritten digit recognition along with in a sentiment classification job. A voice signal is, in fact, a sequence that develops quite slowly (the features are essentially calculated every 10 ms), in which the prior history can almost never be unnecessary. It can be dangerous to completely erase the past even when there are stable discontinuities, such as those that can be seen at the border between a vowel and a fricative. The first modification to standard GRUs proposed in this work thus relates the elimination of the reset gate, which benefits in reducing the redundancy in the gating mechanism. The main benefits of this intervention are in the improved computational efficiency, which is accomplished with the help of the few parameters required to complete the function of a typical GRU.

In the state update equations, the second modification involves restoring the conventional hyperbolic tangent with ReLU activations. Tanh activations are critical since their saturation minimizes the training process and causes gradient issues. Due to numerical instabilities caused by the unbounded ReLU functions utilized over large time series, RNNs did not previously embrace ReLU-based neurons, which have proven helpful in resolving such restrictions.

Batch normalization [54] has just been put forth in the machine learning field and proposes a solution to the so-called internal covariate shift problem by normalizing the mean and variance of each layer's pre-activations for each training set. Such a procedure has proven to be crucial for improving the training process as well as system performance. To RNNs, batch normalization can be applied in a variety of ways. The normalization procedure is improved to include recurrent connections in Cooijmans et al. [55], while authors in Laurent et al. [56] proposed to use it just for feed-forward connections. It is also observed that coupling the proposed model with batch-normalization encourages in bypassing the numerical issues that often appear when dealing with ReLU RNNs used for long-time sequences. Batch normalization essentially bounds the values of the ReLU neurons by rescaling the neuron pre-activations. This makes it easier for the network to profit from the notable advantages of such activations.

3. DL in the Biomedical Applications

An extremely diverse study area with several applications, medical specializations, and related disorders is nourished by the biomedical domain. Physicians are quite knowledgeable about and skilled in some of these illnesses, but not all of them. The biomedical data used by medical profession al nowadays are highly varied due to scientific and technical advancements and include a variety of biological factors, clinical evaluations and metrics, and imaging modalities. Biomedical data are typically unbalanced [57, 58] and nonstationary [59], being defined by a high complexity [57]. This is due to the abundance of these data as well as the completeness of some uncommon conditions. Machine learning offers a huge opportunity in this situation to help doctors, biologists, and medical authorities in utilizing and significantly improving big medical data analysis; lower the risk of medical errors; and produce a better harmonization of the diagnosis and prognosis protocols. DL is one of the emerging machine learning tools in various domains such as image analysis and defect diagnosis. All medical levels are covered by the DL applications in the biomedical sectors, starting with genomic applications like gene expression and ending with public medical health management like forecasting demography data or infectious disease epidemics.

The first phase of biomedical research encompasses all studies, ranging from protein structure prediction and interactions with other proteins and medications to genome sequencing and gene expression. The application of DNNs in this area of research is expanding quickly. The term "Omics" is frequently used in the literature to refer to this field of study, while others have included bioinformatics [60] and biomedicine [61]. Aiming to explore and comprehend biological processes at a molecular level to forecast and prevent diseases by involving patients in the development of more effective and individualized treatment, the Omics covers data from genetics and Omics [62]. Protein–protein interactions [63], the prediction of human drug targets and their interactions [64], and the prediction of protein function [65] all play a significant role in the field of genomics.

The examination of the cell (cytopathology) and the tissue comes next following the DNA and protein levels (histopathology). Histopathology and cytopathology are frequently utilized in the diagnosis of inflammatory disorders, cancer, and several infectious diseases. Under a microscope, the histological and cytopathological slides, which are often obtained via fine-needle aspiration biopsies, are inspected. The primary study area for DL in biomedical applications is bioimaging, as it is known in the literature. Medical imaging studies human organs by

Table 1
Different deep learning methods with brief information

Name of work	Type of deep learning method	Advantage	Disadvantage
CNN-based document analysis by Oleveira et al. [13]	CNN	Applicable for image	Not used for other data rather than image
Image classification by Ramprasath et al. [19]	CNN	Applicable for image	Not used for other data rather than image
Automatic feature extraction by Kunang et al. [22]	AE	Applicable for text data	Not used for other data rather than text
Classification of hepatocellular carcinoma by Koziol et al. [27]	RBM	Applicable for text data	Not used for other data rather than text
Image compression by Lokare et al. [30]	DBN	Applicable for image	Not used for other data rather than image
Speech recognition by Venkateswarlu et al. [33]	RNN	Used for audio data	Not applicable for other data rather than audio
Information retrieval by Deng et al. [38]	DSN	Applicable for text data	Not used for other data rather than text
Handwriting recognition by Carbune et al. [46]	LSTM	Applicable for image	Not used for other data rather than image
Speech recognition by Ravanelli et al. [53]	GRU	Used for audio data	Not used for other data rather than audio

examining many types of imaging [62]. Today, there are major medical high-resolution image acquisition systems available, including parallel MRI, multi-slice computed tomography (CT), ultrasound (US) transducer technology, digital positron emission tomography (PET), and 2D/3D X-ray. The majority of DL applications in bio and medical imaging deal with the interpretation and analysis of computer-aided images [66, 67]. A digital pathology and image analysis with a focus on research and biomarker identification [68] and evaluating histopathology images for the detection of breast cancer [69, 70] are two examples.

The brain and body machine interfaces (BBMIs), which include electrical impulses produced by the brain and muscles and obtained using the proper sensors, are the next level of biomedical applications [62]. Four components make up a BBMI system: a control system, an amplifier, a filter, and a sensing device [71]. To provide a digital interface between the brain and the computer, the system for the brain interface decodes and analyzes signals from a complicated brain mechanism [72]. The brain signals represent the conscious or unconscious neuronal actions brought about by a person's current activity. Recently, a variety of signal acquisition methods have been developed [73]: invasive methods requiring the implantation of electrodes beneath the scalp (such as electrocorticography (ECoG)) and non-invasive methods that do not need placing foreign objects within the brains of subjects. There are numerous assessment methods available, including functional magnetic resonance imaging, functional near-infrared spectroscopy, electroencephalography (EEG), and magnetoencepha lography. The second stage of DL applied to BMIs follows the brain–machine interface and focuses on anomaly detection and illness diagnosis, such as the recognition of coronary artery disease by ECG readings [74], automatic myocardial infarction diagnosis using ECG signals [75], data from EEG to identify seizures [76], and EEG diagnosis of Alzheimer's disease [77].

Analysis of extensive medical data is the goal of public and medical health management (Pm-HM), which aims to improve healthcare choices for the benefit of humanity. One of Pm-HM's challenges in the coming years will be analyzing the spread of disease in cases of epidemics and pandemics in relation to social behavior and environmental factors [78]. Electronic health records, one of the most important and comprehensive sources of patient data, contain information on a patient's medical history, including information on medications and treatment plans as well as information on allergies, radiology images, and sensors multivariate time series (such as EEG). In-depth learning in healthcare decision-making [79], knowledge-distillation approach development [80], temporal pattern discovery over Rochester epidemiology project data [81], or diagnosing given multivariate pediatric intensive care unit time series [82] are all made possible by the analysis of such clinical data against temporal dimensions. Modeling lifestyle disorders like obesity in relation to geographic locations is another aspect of PM-HM. It is now possible to monitor public health issues like contagious intestinal infections [83] or regional obesity via social media, where users' lives and social interactions are publicly disclosed online [84]. In Garimella et al. [85], geo-tagged images from Instagram are used to study the lifestyle diseases, such as obesity, drinking, or smoking. Zhao et al. [86] created the social media nested epidemic simulation (SimNest) using online semi-supervised DL. Das & Das [87, 88], proposed how Parkinson's disease is detected from hand-drawn images using DL. Das & Das [89] also proposed how breast cancer is detected from mammogram images using DL.

4. Rising Architectures: The Generative Adversarial Networks

Generative adversarial networks (GANs) are one of the newest topologies used in biomedical applications. Without heavily annotated training data, GANs offer a method of data augmentation to expand the deep representations. A GAN, first proposed in 2014 by Goodfellow et al. [90], consists of two deep networks: a discriminator and a generator. Finding the parameters of a discriminator and a generator is necessary for training a GAN, with the discriminator's goal being to optimize classification accuracy and the generator's goal being to completely puzzle the discriminator [91]. Only one of the two networks is affected by the parameters update during the training phase. The second one maintains a freeze on its own parameters.

Recently, GANs have been used in all biomedical fields. For example, Li et al. [92] used Omics to model a protein, and they viewed the loop modeling problem as an image in painting problem where the generative network had to accurately estimate the missing area and capture the context of the loop region.

In the BBMI applications, such as cardiac ECG applications, Dhamala et al. [93] employed a unique idea to estimate tissue excitability in a cardiac electrophysiological model by embedding a generative variational autoencoder within the objective function of Bayesian optimization.

The majority of GANs research focuses on applications in medical imaging. Enhancing image quality, reconstructing images, creating created images, registering images, and segmenting images are the key goals. Enhancing image quality is the GANs' primary goal in medical imaging. To reduce the metal artifacts in CT ear pictures of cochlear implant receivers (CIs) in Wang et al. [94], a conditional GAN is utilized to distinguish between the low-dose PET images in Wang et al. [95] and the full-dose PET images of good quality.

Medical image segmentation also employs GANs. For 3D left ventricle segmentation using 3D echocardiography, a unique real-time voxel-to-voxel conditional generative adversarial net is applied in Fichtinger et al. [96]. Zhao et al. [97] employed a cascaded GAN with deep-supervision discriminators to automatically segment bony structures. For reliable segmentation of several HEp-2 datasets, a novel transfer-learning architecture utilizing GANs is proposed in Li & Shen [98].

Giger et al. [99] created a conditional GAN that was trained to recognize the relationship between temporally related ultrasound and 4D MRI navigator images in order to predict the respiratory motion for tracking movable malignancies in the thorax and abdomen during radiotherapy.

Table 2 illustrates the recent published papers of DL in the biomedical applications.

Table 2
Recent papers of deep learning in biomedical applications

Name of work	Description	Advantage	Disadvantage
Deep learning networks in medical imaging by Wang et al. [100]	Through bibliometric and hot spot analysis	Assesses the DL network application in medical image analysis	Only applied in PubMed
Deep neural networks for the early detection of COVID-19 by Islam et al. [101]	Create an algorithm for automatic diagnosis of COVID-19	Speedy detection of COVID-19 because of its high accuracy	Cannot use other respiratory diseases
DL algorithm for the detection of heart diseases. by Roy et al. [102]	To determine the best classifiers for valvular heart disorders	Finds the most efficient, straightforward classifier tool	Requires more time during the learning phase
Serial electrocardiography to detect cardiac pathology by Sbrollini et al. [103]	To identify newly developing cardiac disease in serial ECGs	Identifies acute ischemia and detects newly arising heart failure	Not mentioned
Deep learning-based medical image analysis by Liu et al. [104]	CNN-based DL algorithms for clinical applications	Highly accurate, efficient, and scalable	Reduces the quality of high-quality large-scale datasets
Deep learning method for ultrasonic microbubble imaging by Dai et al. [105]	To suggest and validate a brand-new post-processing technique	Shows excellent imaging performance and great consistency	Not mentioned
A deep learning X-ray-based COVID-19 diagnosis by Hertel & Benlamri [106]	An ensemble classifier to help diagnose probable COVID-19 patients	Most publicly available COVID-19 images in the two-class and three-class datasets	To increase the pipeline's capacity, it needs to expand categorical and numerical data
Neural network model for detection of abnormal heartbeat by Malik et al. [107]	To categorize auditory signals of irregular heartbeats	Diagnoses a heart condition using sound signals	Not mentioned
GANs and its applications in biomedical informatics by Lan et al. [108]	How GAN is utilized in different fields	High-resolution image creation from low resolution images	Limited sample size will result in a relatively low level accuracy
An improved COVID-19 detection using GAN by Asghar et al. [109]	Suggests a neural network to enhance the identification of COVID-19	Good outcomes for COVID-19 detection using X-ray imaging	Does not specify the ideal automated method for finding COVID-19

5. Basics of DNAm and DMR

DNAm [110] is commonly used epigenetic mechanisms and balanced genetic attributes which cannot be illustrated by DNA sequences. DNAm is a procedure where a methyl group is appended to DNA nucleotides. DNAm is an epigenetic approach [111] that modifies gene expression and changes the function of genes by adding a methyl (CH3) group to DNA. The methyl group is added to the cytosine ring's fifth carbon, resulting in 5-methylcytosine, which is a common DNA modification process (5-mC) [112]. DNAm essentially behaves to decrease gene transcription [113, 114] while found in a gene promoter. Whole-genome bisulfite sequencing is a crucial benchmark for determining methylation status (WGBS). However, the high price of WGBS restricts its use in significant epidemiological applications.

The majority of epigenome-wide association studies (EWASs) conducted today use array-based technologies, which provide a full, affordable, and efficient replacement. The MethylationEPIC BeadChip (Infinium) microarray (850K) was created at this time [115]. In addition to the additional 333 265 probes targeting sites in regulatory areas already discovered by the ENCODE [116, 117] and FANTOM5 [118] programs, this enhanced array includes more than 90% of the 450K array probes.

Currently, it has been observed that adjacent epigenetic regions with variable levels of methylation (DMRs) are associated with various disorders [119–121]. Additionally, hypermethylation of candidate gene promoter region aids in the development of neoplasms and contributes to the carcinogenesis of colon cancer [122]. In postmortem brain cells from patients with autism spectrum disorder, Ladd-Acosta et al. [12] identified and cloned three genomic sites with significant DNA reversal in neuro-developmental disorders. The vast histocompatibility complex region has two clusters whose differential methylation likely resolves the hereditary hazard for rheumatoid arthritis, according to Liu et al. [123].

The different tolls have been created for the study of DMRs. There are two types of DMR finding techniques: supervised and unsupervised approaches. The unsupervised techniques discover each genomic region for organization with phenotypic information after organizing the CpG probes into genomic regions. On the other hand, supervised algorithms calculate a p-value or related t-statistic at each CpG first, after which user-specified parameters are used to identify the sections of the genome with a string of tiny p-values or t-statistics. The portions of the genome with a string of modest p-values or t-statistics are identified using user-specified parameters after supervised algorithms generate a

p-value or associated t-statistic at each CpG. Two earlier researches [124, 125] analyzed unsupervised DMR search techniques that check for prefixed genomic regions. Yet, there is presently a shortage of supervised DMR finding techniques calculation.

In this survey, we prepare a thorough evaluation of the key software tools for supervised DMR study, including bump hunting [126], comb-p [127], DMRcate [10], and Probe Lasso [128]. We elect these mechanisms using a variety of different principles: (i) it is possible to use them to assess Illumine methylation arrays. (ii) To find DMRs, they employ the supervised search technique. (iii) The open source code software can be expanded to identify an EWAS with a balanced sample size in a reasonable amount of time. R or Python programming languages are the most popular programming languages for epigenetic research.

In essence, we identify that the algorithms associated with these methylation analysis tools frequently require a variety of criteria to be presented. We intend to conduct a thorough evaluation of the tools with various parameter settings in order to help users relate to and find the optimal parameter settings for DNAm analysis tools. By using these multiple methods, we may compare additional characteristics of the analysis results, such as the size of the DMRs and overlap between the techniques. In the Methods section, we go into detail about the simulation technique, and in the Results section, we go into detail about the results of the simulation research and a real-world dataset. In the final section, we provide a succinct summary of our key findings and highlight potential future study avenues in this broad field.

6. Methods for Identifying DMRs

6.1. DMRcate

The DMRcate [10] technique is used in the Bioconductor package DMRcate. The empirical Bayesian methodology from the limma R package was used to fit a linear model at each CpG before employing the DMRcate procedure. In our study, this model incorporated group status as an independent variable and methylation M value as an outcome variable. In the study of methylation data, M values, which are logit transformed beta values with the formula $M = log(beta/1 - beta)$, have been shown to have improved statistical features such as homoscedasticity [129]. After that, for each position, the statistic $Y = t2$ is calculated, where t is the t-statistic from the linear model associated with the group effect. In the following phase, DMRcate applies kernel smoothing using a bandwidth-scaled Gaussian smoother. The p-values at each position are then calculated by moment-matching using Satterthwaite's approach [130]. Then, the Benjamini & Hochberg [131] technique is used to select the CpG sites with multiple comparison-corrected p-values as significant CpGs. By splitting adjacent important CpGs that are within nucleotides of one another, regions for DMRs are discovered. Stouffer's technique [132] is used to calculate the p-value for DMR.

6.2. Bumphunter

Bumphunter [126] is applied in the Bioconductor packages bumphunter and minfi. In this bumphunter method, differential methylation between case and control groups at each CpG site is first modeled using a linear regression model degenerating the M value on group. Then candidate regions (bumps), which are groups of subsequent probes for which all t statistics exceeded a user-defined threshold, are examined. The statistical significance of the candidate regions separated by at least maxgap base pairs is then estimated using permutation tests, which permute sample labels to produce the null distribution of candidate regions. Spatial correlation structures are used in the description of regions to visualize correlations of methylation levels between nearby CpGs. We looked at how bumphunter behaved in the simulation research in relation to the parameters maxGap, pickCutoffQ, and B (the number of resamples) used to estimate DMR P-values.

6.3. Probe Lasso

Probe Lasso [128] is utilized in the Bioconductor package ChAMP. In the Probe Lasso method, the differential methylation between case and control groups at each CpG site is first determined using a linear model regressing beta value on group. Based on the sort of genomic characteristics the probe is situated in (such as TSS200, 3 UTR), the Probe Lasso describes adjustable borders around each probe. The Probe Lasso algorithm "throws" a lasso around each probe with dynamic bounds that is focused on the target probe, much like a real lasso would. If there are more significant probes stored within the Probe Lasso boundary than the user-specified threshold, a region surrounding the target probe is selected. Using the Stouffer's method, Probe Lasso calculates a p-value for each region by weighting individual probes by the inverse sum of their squared correlation coefficient in the correlation matrix. This is done for each region by first calculating a correlation matrix of normalized beta values within that region. In order to achieve Probe Lasso, we looked at the effects of the parameters adjPvalProbe (significance threshold for probes to be included in DMRs), meanLassoRadius (radius around each useful probe to discover a DMR), and minDmrSep (the minimal separation in base pairs between nearby DMRs).

6.4. Comb-p

Comb-p [127] is a command-line utility and a Python module. However, none of the three methods described above allow for the determination of p-values for individual CpGs. Nevertheless, comb-p requires a .BED file as input, which contains the p-values and chromosome locations of the CpG sites. The Stouffer–Liptak–Kechris correction [133] is used to determine adjusted p-values at each CpG site after the comb-p tool generates correlations at various distance lags. If the neighboring CpG sites also have relatively low p-values, the corrected p-value at a CpG site will be lower than the initial p-value. However, if nearby p-values are similarly high, the adjusted p-value at a CpG site will continue to be high. The false discovery rate (FDR) is then calculated at each CpG site, after which a peak-finding technique is used to identify regions enhanced with low p-values. Following the discovery of the regions, the Stouffer–Liptak adjustment is used to determine the final p-value for each region. We examined how the parameters distance (which increased a region if there was another p-value narrower than seed within this distance) and seed (the p-value implication threshold to start an area) affected the performance of comb-p.

6.5. IMA: Different index metrics

IMA [134] is used in R and executed on any platform that already has R and Bioconductor installation. The user has the

option to specify alternative tracks in the parameter file or run the pipeline with default settings. Below is an analysis of the IMA pipeline:

The methylation values of specific sites expressed by Illumina BeadStudio or GenomeStudio software are used as input by IMA during the preprocessing stage. For the purposes of particulate quality control, it allows the user to choose different filtering steps or modify the filtering criteria. Users can choose to filter out the loci whose DNAm levels are determined by probes holding SNP(s) at/near the specified CpG site because they may not be sufficient to calculate DNAm value. Additionally provided is the option for sample-level quality control. Although Illumina will recommend using the raw values for evaluation, the user has the option to structure the methylation level as the response in a linear model using the arcsine square root transformation [135, 136]. The option of logit transformation is also available [137]. The default setting for IMA is that no normalization will be used, while quantile normalization is an alternative preprocessing option that is available. Quantile normalization has been shown to be insufficient for eliminating all of the additional technical variation across samples [138]. The development of a normalization approach for DNAm research is a fruitful field of expanding inquiry [139].

Three different index metrics are now used in IMA: mean, median [134], and Tukey's biweight robust average [140–142]. Typically, the region's methylation index will be determined by taking the median value. For each particular region, the Wilcoxon rank-sum test (the default), the Student's t-test, and empirical Bayes statistics can be used in the differential methylation analysis step. The implication criterion can be mentioned by users in the parameter file. To find site-level differential methylation inference, the same statistical supposition and multiple test correction approaches described above can be used to each individual site. For each of the three aforementioned sections, specific output files are provided during the output stage. The result for the preprocessing stage includes a matrix of methylation values for approved loci across approved samples. There is a matrix of methylation index across the samples for each region class of relevance for the part on methylation index calculation. The differential methylation values as well as the raw and modified p-values of each region of interest are given for the differential methylation analysis section.

6.6. Aclust

There are two stages to the epigenome-wide investigation. Using the Adjacent sites algorithm ("A-clustering," or shortly "Aclust") [143], clusters of associated methylation sites are identified in the first stage. In the next stage, generalized estimating equations (GEEs) are used to examine these clusters for finding results [144].

The grouping of nearby linked CpG sites is the initial step. This clustering algorithm, Aclust, is identical to the agglomerative nesting clustering algorithm [145]. However, it is constrained in such a way that only probes that are adjacent to one another can be grouped together, or more broadly, only adjacent clusters can be merged to form a larger one-cluster arrangement. If the distance between two clusters is less than a set threshold, neighboring clusters are combined. In this case, the clustering is used by cycling over the sites, A-clustering ordered by location. The distance metric in this case is identical and is based on the actual methylation levels found in the sample. The distance metric between two clusters depends on how far apart the cluster's probes are from one another.

The algorithm next examines the impact of exposure on the identified clusters of methylation sites, where one can choose the minimum cluster sizes, after determining clusters of probes, whether with or without an initial dbp-combine step. This technique involves first fitting a GEE model that dominates individual location effects for each site as well as common exposure and covariate effects on all sites within a cluster. If the batch effect was previously removed, say via ComBat [146], further fitting for batch serves no purpose. The p-value for the exposure variable from the GEE model is then the raw p-value. Using established techniques, such as control of the FDR, it is corrected for multiple testing after providing raw p-values for each of the clusters.

6.7. Seqlm

The Seqlm [147] technique is achieved by using the following three phase procedures: 1. The distances between adjacent CpG probes are used to divide the genome into beginning segments. 2. Based on the varied methylation patterns, these segments are divided into regions. 3. The statistical significance of differential methylation is examined for each location. The implementation of Stages 1 through 3 is rather simple.

The CpG sites on the genome are first clustered into smaller groups close to the promoters and other functional components in the initial segmentation process, and the arrays also concentrate on these regions. The distance between two successive probes must be at least 1000 bp apart in order for them to belong to the same area in this case. The precise cut-off value is assessed by examining the relationship between the genomic distance and the methylation correlation of consecutive sites in a sizable dataset [148]. As is typical, there is high correlation between relatively adjacent pairs of probes (less than 100 bp), although the favored correlation effect appears to be diminished when the distance is already over 1000 bp. This initial segmentation creates a large number of sparsely populated segments in locations where the array has adequate coverage as well as several solitary sites and brief periods. These will be sorted into regions in the following stage based on their methylation patterns.

The degree of differential methylation is fixed within each segment in the refined methylation-based segmentation stage, which divides an extended distance of CpG probes into sections with uniform methylation patterns. As long as the segment-wise linear models serve a strong match, the ideal segmentation should favor longer regions over shorter ones. The minimum description length (MDL) concept can be used to identify this aim as a model selection problem [149].

6.8. coMethDMR

The coMethDMR [150] has two vital phases: (i) find the subregion of a genomic area that has nearby and co-methylated CpGs first and (ii) analyze the relationship between the subregion's CpG methylation and phenotypic while accounting for CpG variation collectively. The genome will be divided into areas in the first stage using methylation array annotations as a guide. The regions can be chosen based on their relationships to genes or CGIs because the Illumina chips target methylation sites primarily at genomic regions and CpG islands. Alternatively, CpG probes can be arranged into islands, coasts, or shelves depending on how they relate to CGIs. First, the CpG probe clusters are extracted that are almost within these genomic areas. The genetic areas would have equal CpG concentrations,

Table 3
Different methods of DMRs

Name of the method	Source of the method	Brief definition of model/statistics	Type of method	Reference
DMRcate	rdrr.io/bioc/DMRcate/	Empirical Bayesian methodology	Supervised	Peters et al. [10]
Bumphunter	github.com/rafalab/bumphunter	Linear regression	Supervised	Jaffe et al. [126]
Probe Lasso	ChAMP/versions/1.8.2/topics/champ.lasso	Linear regression	Supervised	Butcher & Beck [128]
Comb-p	github.com/brentp/combined-pvalues	Stouffer–Liptak–Kechris correction	Supervised	Pedersen et al. [127]
IMA:mean	(rforge.net/IMA/)	Regression	Unsupervised	Wang et al. [134]
IMA:median	(rforge.net/IMA/)	Regression	Unsupervised	Wang et al. [134]
IMA:Tukey's Biweight robust average	https://rdrr.io/cran/DescTools/f/	Regression	Unsupervised	Kohl et al. [140]
Aclust	(github.com/tamartsi/Aclust/)	Regression	Unsupervised	Sofer et al. [143]
Seqlm	(github.com/raivokolde/seqlm)	Regression	Unsupervised	Kolde et al. [147]
coMethDMR	(github.com/lissettegomez/coMethDMR)	Regression	Unsupervised	Gomez et al. [150]

thanks to this procedure. In the second phase, a random coefficient combination model is suggested for evaluating groups of CpGs against phenotype in order to simultaneously design modifications among the co-methylated CpGs and relationship with phenotype. This design offers (i) values of normalized methylation as the outcome variable, (ii) a structured element that calculates the average for each group of CpGs, and (iii) the random coefficients, which determine how each CpG's slope per stage differs from the group mean, are a random component. This model is a combined effects model since it includes both fixed and random effects.

7. Comparative Study of Supervised and Unsupervised Consecutive Clustering Methods

According to the article [11], an experiment has been carried out in terms of simulation study using a real methylation dataset (see Table 3). According to that result mentioned there, the performance of different supervised consecutive clustering algorithms is described in terms of small effect size (low μ value) (in Table 4), large effect size (higher μ value) (in Table 5), and elapsed time (in Table 6).

Table 4
Comparative outcomes of different supervised consecutive clustering tools for small effect size

Method	Precision	Power
DMRcate	++	–
Bumphunter	–	–
Probe Lasso	+	–
Comb-p	++	–

" " and "-" signify very poor and poor outcome for the said standard, respectively. "+" and "++" refer to fair and excellent outcomes for the said standard, respectively.

Table 5
Comparative outcomes of different supervised consecutive clustering tools for large effect size

Method	Precision	Power
DMRcate	++	+
Bumphunter	+	–
Probe Lasso	++	+
Comb-p	++	++

"-" indicates poor outcomes for the said standard. "+" and "++" signify fair and excellent outcomes for the said standard, respectively.

A comparative study of different unsupervised techniques (i.e., IMA: Different Index Metrics, Aclust, Seqlm and coMethDMR) is illustrated in the following:

IMA: Different Index Metrics: The IMA package's primary goal is to generate a variety of frequently used DNAm microarray analysis choices for users to choose from for their previous analysis and characterization in an automated manner. It was developed in an open-source R environment, giving users the freedom to maintain and enhance the functionality to suit their own requirements. It can be used as a pipeline of differential analysis and methylation level index for later functional analysis and hypothesis formation. To locate clustered coastlines associated with the desired trait, for instance, model-based clustering [151] can be used with the matrix of methylation index for shore regions established by IMA.

The development of analytical methods for DNAm microarray research is still accelerating [152, 153]. The IMA package's capabilities will be expanded in the future by incorporating fresh preprocessing and differential analysis methods.

Aclust: A clustering technique that groups nearby methylation sites in respect to their distance typically inferred from their correlation and provides a pipeline for methylation data testing as

outcomes. By creating a smaller number of analytic units, the clustering is used to both locate coregulated regions of sites and reduce dimension. This technique can also be used with sparser sequencing data. Powerful grouping and analysis tools can determine p-values without the use of resampling methods.

Seqlm: The seqlm MDL framework is a flexible method for identifying genomic areas. The target regions' attributes can be defined using a variety of statistical models. For instance, more complex linear models can be added to examine more complex hypotheses, and clustering techniques can be specified to accomplish unsupervised region discovery. Future directions for the advancement of this methodology are numerous. Along with the two sets of data, the existing model may evaluate continuous variables as well. As a result, seqlm can be used to identify loci for methylation quantitative traits.

Table 6
Comparative outcomes of different supervised consecutive clustering tools in terms of elapsed time

Method	Speed
DMRcate	++
Bumphunter	+
Probe Lasso	++
Comb-p	+

"+" and "++" indicate higher elapsed time and low elapsed time for the given criterion, respectively.

coMethDMR: When DMR analysis is performed on array-based DNAmethylation data, coMethDMR produces an expandable, reliable, and accurate result. The entire analytical process has been resolved as an open-source R package that the scientific community can use for free. Gomez et al. [150] demonstrated that coMethDMR outperforms directly testing a genomic region with a continuous phenotype in terms of power and well-contained false positive rate. With the use of coMethDMR, epigenetic researchers can extract pertinent biological observations from vast, intricate datasets on DNAm.

8. Conclusion

DL is an emerging topic used everywhere. Here, we provided an extensive survey on DL and consecutive clustering in the field of biomedical and health sciences. Until now, various DL algorithms and consecutive clustering algorithms (viz., DMR finding algorithms) have been developed, but for the bioinformatics and biomedical research specially for the epigenetics/epigenomic study, only a few review papers have been produced so far. Thus, in this paper, we provided a comprehensive review article on various DL and consecutive clustering algorithms (DMR finding algorithms) for biomedical research (specially, for epigenetics study). To do so, initially we demonstrated several DL architectures (viz., CNN, AE, RBM and DBN, RNN, DSN, and LSTM/GRU) along with their advantages and disadvantages. Thereafter, we described different categories of consecutive clustering algorithms for epigenetics study (termed as DMR finding tools), viz., supervised DMR finding techniques (e.g., DMRcate, Bumphunter, Probe Lasso, and Comb-p) and unsupervised DMR finding methods (viz., IMA: different index metrics, Aclust, Seqlm and coMethDMR) along with their pros and cons.

In the final part of the survey, we provided a summary of various consecutive clustering algorithms applied to epigenetics study based on several evaluation metrics such as precision, power, and speed as collected from literature search. Clustering refers to a group of unsupervised classification techniques which, generally, only rely on the available data patterns to infer groups of similar patterns. Among the many available clustering algorithms, the so-called consecutive clustering algorithms have been characterized as fast and straightforward methods which produce better result with high accuracy as compared to other clustering algorithms. Moreover, our survey paper is highly useful for the bioinformatics researchers (specially, epigenetics researchers) to understand where and how to utilize DL algorithms and consecutive clustering algorithms to fulfill specific computational biomedical problems. As a future work, we will develop a new algorithm integrating DL and consecutive clustering strategies together applied on DNAm data to find DMRs (consecutive clusters of CpG sites) for any kind of tissue-specific cancer disease (e.g., breast cancer, kidney cancer, etc.) that will be benevolent for the bioinformatics researchers to catch and grasp the new domain of epigenetics.

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflict of Interest

Sarauv Mallik is an Associate Editor for *Artificial Intelligence and Applications*, and was not involved in the editorial review or the decision to publish this article. The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study

References

[1] Dargan, S., *et al.* (2020). A survey of deep learning and its applications: A new paradigm to machine learning. *Archives of Computational Methods in Engineering*, *27*, 1071–1092.
[2] Picton, P., *et al.* (2000). Neural Networks.
[3] LeCun, Y., *et al.* (2015). Deep learning. *Nature*, *521*, 1–10.
[4] Cios, K. J., *et al.* (2005). Computational intelligence in solving bioinformatics problems, *Artificial Intelligence in Medicine, 35,* 1–8.
[5] Karim, R., *et al.* (2020). Deep learning-based clustering approaches for bioinformatics. *Briefings in Bioinformatics*.
[6] Christensen, B. C., *et al.* (2011). DNA methylation, isocitrate dehydrogenase mutation, and survival in glioma. *Journal of the National Cancer Institute*, *103*, 143153.
[7] Bibikova, M., Barnes, B., Tsan, C., *et al.* (2011). High density DNA methylation array with single CpG site resolution. *Genomics*, *98*, 288–295.
[8] Portela, A., & Esteller, M. (2010). Epigenetic modifications and human disease. *Nature Biotechnology*, *28*, 1057–1068.
[9] Rakyan, V. K., Down, T. A., Balding, D. J., *et al.* (2011). Epigenome-wide association studies for common human diseases. *Nature Reviews Genetics*, *12*, 529–541.
[10] Peters, T. J., Buckley, M. J., Statham, A. L., *et al.* (2015). De novo identification of differentially methylated regions in the human genome. *Epigenetics Chromatin*, *8*.

[11] Mallik, S., *et al.* (2019). An evaluation of supervised methods for identifying differentially methylated regions in Illumina methylation arrays. *Briefings in Bioinformatics*, *20*, 2224–2235.
[12] Ladd-Acosta, C., Hansen, K. D., Briem, E., *et al.* (2014). Common DNA methylation alterations in multiple brain regions in autism. *Molecular Psychiatry*, *19*, 862–871.
[13] Oleveira, D., *et al.* (2017). *Fast CNN-based document layout analysis*. IEEE.
[14] Wong, K., *et al.* (1982). Document analysis system. *IBM Journal of Research and Development*, *26*, 646–656.
[15] Nair, V., *et al.* (2010). *Rectified linear units improve restricted Boltzmann machines*, pp. 807–814.
[16] Giusti, A., *et al.* (2013). *Fast image scanning with deep max-pooling convolutional neural networks*.
[17] Srivastava, N., *et al.* (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.
[18] Krizhevsky, A., *et al.* (2012). ImageNet classification with deep convolutional neural networks. *Proceedings of Neural Information Processing Systems*, 1106–1114.
[19] Ramprasath, M., *et al.* (2018). Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics*, *119*, 1307–1319.
[20] Wang, Y., *et al.* (2016). Auto-encoder based dimensionality reduction. *Neurocomputing*, *184*, 232–242.
[21] Manning, T., *et al.* (2017). PCA and Autoencoders.
[22] Kunang, Y., *et al.* (2018). Automatic features extraction using autoencoder in intrusion detection system.
[23] Potdar, K., *et al.* (2017). A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, *175*, 1307–1319.
[24] Potluri, S., *et al.* (2016). Accelerated deep neural networks for enhanced Intrusion Detection System.
[25] Harasimowicz, A. (2014). Comparison of data preprocessing methods and the impact on auto-encoders performance in activity recognition domain.
[26] Fischer, A., *et al.* (2012). An introduction to restricted Boltzmann machines.
[27] Koziol, J., *et al.* (2014). *Restricted Boltzmann machines for classification of hepatocellular carcinoma*.
[28] Fischer, A., *et al.* (2014). Training restricted Boltzmann machines: An introduction. Pattern Recognition, 47, 25–39.
[29] Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, *14*, 1771–1800.
[30] Lokare, V., *et al.* (2015). Application of deep belief networks for image compression. *International Journal of Computer Science and Information Technologies*, *6*, 4799–4803.
[31] Lamb, A., Goyal, A., Zhang, Y., *et al.* (2016). *Professor forcing: A new algorithm for training recurrent networks*.
[32] Werbos, P., *et al.* (1990). Backpropagation through time: What it does and how to do it.
[33] Venkateswarlu, R., *et al.* (2011). Speech recognition by using recurrent neural networks. *International Journal of Scientific Engineering Research*, *2*.
[34] Rabiner, L., *et al.* (1993). *Fundamentals of speech recognition*.
[35] Koizumi, T., *et al.* (1996). *Recurrent neural networks for phoneme recognition*.
[36] Deng, L., Yu, D., & Platt, J. (2012). Scalable stacking and learning for building deep architectures. In *Proceedings of ICASSP*.
[37] Deng, L., & Yu, D. (2011). Deep Convex Network: A scalable architecture for deep learning. In *Proceedings of Interspeech*.
[38] Deng, L., *et al.* (2013). Deep stacking networks for information retrieval.
[39] Burges, C., *et al.* (2006). Learning to rank with non-smooth cost functions.
[40] Hillard, D., *et al.* (2011). The sum of its parts: Reducing sparsity in click estimation with query segments. *Journal of Information Retrieval*, *14*, 315–336.
[41] Hillard, D., *et al.* (2010). *Improving Ad relevance in sponsored search*.
[42] Gao, J., *et al.* (2009). Smoothing clickthrough data for web search ranking.
[43] Gao, J., *et al.* (2010). Clickthrough-based translation models for web search: from word models to phrase models.
[44] Hochreiter, S., *et al.* (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.
[45] Zhou, G., *et al.* (2016). *Minimal gated unit for recurrent neural networks*.
[46] Carbune, V., *et al.* (2018). *Fast multi-language LSTM-based online handwriting recognition*.
[47] Sainath, T., *et al.* (2015). Convolutional, long short-term memory, fully connected deep neural networks.
[48] Schuster, M., *et al.* (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*, 2673–2681.
[49] Keysers, D., *et al.* (2017). Multi-language online handwriting recognition. *IEEE Transactions Pattern Analysis Machine Intelligence*, *39*, 1180–1194.
[50] Jozefowicz, R., *et al.* (2015). An empirical exploration of recurrent network architectures.
[51] Graves, A., *et al.* (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks.
[52] Graves, A., *et al.* (2014). Towards end-to-end speech recognition with recurrent neural networks.
[53] Ravanelli, M., *et al*. (2017). Improving speech recognition by revising gated recur rent units.
[54] Ioffe, S., *et al.* (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.
[55] Cooijmans, T., *et al.* (2016). Recurrent Batch normalization.
[56] Laurent, C., *et al.* (2016). Batch normalized recurrent neural networks.
[57] He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21, 1263–1284. https://doi.org/10.1109/TKDE.2008.239.
[58] Yu, H., Yang, X., Zheng, S., & Sun, C. (2018). Active learning from imbalanced data: a solution of online weighted extreme learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, *30*, 1–16. https://doi.org/10.1109/TNNLS.2018.2855446.
[59] Ditzler, G., Roveri, M., Alippi, C., & Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, *10*, 12–25. https://doi.org/10.1109/MCI.2015.2471196.
[60] Raví, D., Wong, C., Deligianni, F., Berthelot, M., *et al.* (2017). Deep learning for health informatics. *IEEE Journal of Biomedical and Health Informatics*, *21*, 4–21. https://doi.org/10.1109/JBHI.2016.2636665.
[61] Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular Pharmaceutics*, *13*, 1445–1454. https://doi.org/10.1021/acs.molpharmaceut.5b00982.

[62] Mahmud, M., Kaiser, M., Hussain, A., & Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE Transactions on Neural Networks and Learning Systems*, *29*, 2063–2079. https://doi.org/10.1109/TNNLS.2018.2790388.

[63] Bagchi, A. (2018). Protein-protein interactions: Basics, characteristics, and predictions. In H. J. Purohit, V. C. Kalia, & R. P. More (Eds.), *Soft computing for biological systems*, Springer (pp. 111–120). https://doi.org/10.1007/978-981-10-7455-4-7.

[64] Nath, A., Kumari, P., & Chaube, R. (2018). Prediction of human drug targets and their interactions using machine learning methods: Current and future perspectives. In M. Gore, U. B. Jagtap (Eds.), *Computational drug discovery and design*, Springer (pp. 21–30). https://doi.org/10.1007/978-1-4939-7756-7-2.

[65] Shehu, A., Barbará, D., & Molloy, K. (2016). A survey of computational methods for protein function prediction. In K. C. Wong (Ed.), *Big data analytics in genomics*. Springer International Publishing, pp. 225–298. https://doi.org/10.1007/978-3-319-41279-5-7.

[66] de Bruijne, M. (2016). Machine learning approaches in medical image analysis: From detection to diagnosis. *Medical Image Analysis*, *33*, 94–97. https://doi.org/10.1016/j.media.2016.06.032.

[67] Kergosien, Y. L., & Racoceanu, D. (2017). Semantic knowledge for histopathological image analysis: From ontologies to processing portals and deep learning. In *Proceedings of the 13th International Conference on Medical Information Processing and Analysis*, San Andres Island, Colombia, 5–7 October 2017, p. 105721F. https://doi.org/10.1117/12.2285916.

[68] Hamilton, P. W., Bankhead, P., Wang, Y., Hutchinson, R., Kieran, D., McArt, D. G., James, J., & Salto-Tellez, M. (2014). Digital pathology and image analysis in tissue biomarker research. *Methods*, 70, 59–73. https://doi.org/10.1016/j.ymeth.2014.06.015.

[69] Chen, J. M., Li, Y., Xu, J., Gong, L., *et al.* (2017). Computer-aided prognosis on breast cancer with hematoxylin and eosin histopathology images: A review. *Tumor Biology*, *39*, https://doi.org/10.1177/1010428317694550.

[70] Veta, M., Pluim, J. P. W, van Diest, P. J., & Viergever, M. A. (2014). Breast cancer histopathology image analysis: A review. *IEEE Transactions on Biomedical Engineering*, *61*, 1400–1411. https://doi.org/10.1109/TBME.2014.2303852.

[71] Major, T. C., & Conrad, J. M. (2014). A survey of brain computer interfaces and their applications. In *Proceedings of the IEEE SOUTHEASTCON* 2014, Lexington, KY, USA, 13–16 March 2014, pp. 1–8. https://doi.org/10.1109/SECON.2014.6950751.

[72] Kerous, B., & Liarokapis, F. (2016). Brain-computer interfaces—A survey on interactive virtual environments. In *Proceedings of the 2016 8th International Conference on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, Barcelona, Spain, 7–9 September 2016, pp. 1–4. https://doi.org/10.1109/VS-GAMES.2016.7590339.

[73] Abdulkader, S. N., Atia, A., & Mostafa, M. S. M. (2015). Brain computer interfacing: Applications and challenges. *Egyptian Informatics Journal*, *16*, 213–230. https://doi.org/10.1016/j.eij.2015.06.002.

[74] Tan, J. H., Hagiwara, Y., Pang, W., *et al.* (2018). Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals. *Computers in Biology and Medicine*, *94*, 19–26. https://doi.org/10.1016/j.compbiomed.2017.12.023.

[75] Acharya, U. R., Fujita, H., Oh, S. L., Hagiwara, Y., Tan, J. H., & Adam, M. (2017). Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. *Information Science, 415,* 190–198. https://doi.org/10.1016/j.ins.2017.06.027.

[76] Turner, J. T., Page, A., Mohsenin, T., & Oates, T. (2017). Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection. CoRR: CoRR: Leawood, KS, USA.

[77] Zhao, Y., & He, L. (2015). Deep learning in the EEG diagnosis of Alzheimer's disease. In *Asian conference on computer vision*. Springer, pp. 340–353. https://doi.org/10.1007/978-3-319-16628-5-25.

[78] Ong, B. T., Sugiura, K., & Zettsu, K. (2016). Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM2.5. *Neural Computing and Applications*, *27*, 1553–1566. https://doi.org/10.1007/s00521-015-1955-3.

[79] Liang, Z., Zhang, G., Huang, J. X., & Hu, Q. V. (2014). Deep learning for healthcare decision making with EMRs. In *Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Belfast, UK, 2–5 November 2014, pp. 556–559. https://doi.org/10.1109/BIBM.2014.6999219.

[80] Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2015). Distilling knowledge from deep networks with applications to healthcare domain. ArXiv, arXiv:1512.03542.

[81] Mehrabi, S., Sohn, S., Li, D., Pankratz, J. J., *et al.* (2015). Temporal pattern and association discovery of diagnosis codes using deep learning. In *Proceedings of the 2015 International Conference on Healthcare Informatics*, Dallas, TX, USA, 21–23 October 2015, pp. 408–416. https://doi.org/10.1109/ICHI.2015.58.

[82] Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. C. (2015). Learning to Diagnose with LSTM Recurrent Neural Networks. arXiv:1511.03677.

[83] Zou, B., Lampos, V., Gorton, R., & Cox, I. J. (2016). On infectious intestinal disease surveillance using social media content. In *Proceedings of the 6th International Conference on Digital Health Conference*, Montréal, QC, Canada, 11–13 April 2016. ACM, pp. 157–161. https://doi.org/10.1145/2896338.2896372.

[84] Phan, N., Dou, D., Piniewski, B., & Kil, D. (2015). Social restricted Boltzmann Machine: Human behavior prediction in health social networks. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, Paris, France, 25–28 August 2015, pp. 424–431. https://doi.org/10.1145/2808797.2809307.

[85] Garimella, V. R. K., Alfayad, A., & Weber, I. (2016). Social media image analysis for public health. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose, CA, USA, 7–12 May 2016.

[86] Zhao, L., Chen, J., Chen, F., Wang, W., Lu, C., & Ramakrishnan, N. (2015). SimNest: Social media nested epidemic simulation via online semisupervised deep learning. In *Proceedings of the 2015 IEEE Inter-national Conference on Data Mining*, Atlantic City, NJ, USA, 14–17 November 2015, pp. 639–648. https://doi.org/10.1109/ICDM.2015.39.

[87] Das, A., Das, H., *et al.* (2020a). Detection of breast cancer from mammogram images using deep transfer learning. *Advances in*

Signal Processing and Intelligent Recognition Systems, 18–27. https://doi.org/10.1007/978-98116-0425-6_2.

[88] Das, A., Das, H., *et al.* (2020b). Detection of Parkinson's disease from handdrawn images using deep transfer learning. *Intelligent Learning for Computer Vision*, 67–84. https://doi.org/10.1007/978-981-33-4582-9_6.

[89] Das, A., Das, H., *et al.* (2021). Performance analysis of different machine learning classifiers in detection of Parkinson's Disease from handdrawn images using histogram of oriented gradients. *Applications of Artificial Intelligence in Engineering*, 205–215, https://doi.org/10.1007/978981-33-4604-8_16.

[90] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems*, vol. 27. Curran Associates, Inc. (pp. 2672–2680).

[91] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine, 35,* 53–65. https://doi.org/10.1109/MSP.2017.2765202.

[92] Li, Z., Nguyen, S. P., Xu, D., & Shang, Y. (2017). Protein loop modeling using deep generative adversarial network. In *Proceedings of the* 2017 *IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, Boston, MA, USA, 6–8 November 2017, pp. 1085–1091. https://doi.org/10.1109/ICTAI.2017.00166.

[93] Dhamala, J., Ghimire, S., Sapp, J. L., Horácek, B. M., & Wang, L. (2018). High-dimensional Bayesian optimization of personalized cardiac model parameters via an embedded generative model. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention—MICCAI* 2018. Springer International Publishing (pp. 499–507).

[94] Wang, J., Zhao, Y., Noble, J. H., & Dawant, B. M. (2018). Conditional generative adversarial networks for metal artifact reduction in CT images of the ear. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-Lopez, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention—MICCAI* 2018. Springer International Publishing (pp. 3–11).

[95] Wang, Y., Yu, B., Wang, L., Zu, C., Lalush, D. S., Lin, W., Wu, X., Zhou, J., Shen, D., & Zhou, L. (2018). 3D conditional generative adversarial networks for high-quality PET image estimation at low dose. *NeuroImage*, *174*, 550–562. https://doi.org/10.1016/j.neuroimage.2018.03.045.

[96] Dong, S., Luo, G., Wang, K., Cao, S., Mercado, A., Shmuilovich, O., Zhang, H., & Li, S. (2018). VoxelAtlasGAN: 3D left ventricle segmentation on echocardiography with atlas guided generation and voxel-to-voxel discrimination. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention—MICCAI* 2018. Springer International Publishing (pp. 622–629).

[97] Zhao, M., Wang, L., Chen, J., Nie, D., Cong, Y., Ahmad, S., Ho, A., Yuan, P., Fung, S. H., Deng, H. H., *et al.* (2018). Craniomaxillofacial bony structures segmentation from MRI with deep-supervision adversarial learning. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention—MICCAI* 2018, Springer International Publishing, pp. 720–727.

[98] Li, Y., & Shen, L. (2018). cC-GAN: A robust transfer-learning framework for HEp-2 specimen image segmentation. IEEE Access, 6, 14048–14058. https://doi.org/10.1109/ACCESS.2018.2808938.

[99] Giger, A., Sandkühler, R., Jud, C., Bauman, G., Bieri, O., Salomir, R., & Cattin, P. C. (2018). Respiratory motion modelling using cGANs. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention—MICCAI* 2018. Springer International Publishing (pp. 81–88).

[100] Wang, L., Wang, H., Huang, Y., Yan, B., Chang, Z., Liu, Z., Zhao, M., Cui, L., Song, J., Li, F. (2022). Trends in the application of deep learning networks in medical image analysis: Evolution between 2012 and 2020. *European Journal of Radiology*, *146*, 110069. ISSN 0720-048X. https://doi.org/10.1016/j.ejrad.2021.110069.

[101] Islam, R., Abdel-Raheem, E., & Tarique, M. (2022). A study of using cough sounds and deep neural networks for the early detection of Covid-19. *Biomedical Engineering Advances*, *3*, 100025, ISSN 2667-0992. https://doi.org/10.1016/j.bea.2022.100025.

[102] Roy, T. S., Roy, J. K., & Mandal, N. (2022). Classifier identification using deep learning and machine learning algorithms for the detection of valvular heart diseases. *Biomedical Engineering Advances*, *3*, 100035. ISSN 26670992. https://doi.org/10.1016/j.bea.2022.100035.

[103] Sbrollini, A., De Jongh, M., Ter Haar, C., *et al.* (2019). Serial electrocardiography to detect newly emerging or aggravating cardiac pathology: A deep-learning approach. *BioMedical Engineering OnLine, 18*, 15. https://doi.org/10.1186/s12938-019-0630-9.

[104] Liu, X., Gao, K., Liu, B., Pan, C., Liang, K., Yan, L., Ma, J., He, F., Zhang, S., Pan, S., & Yu, Y. (2021). Advances in deep learning-based medical image analysis. *Health Data Science*, *Article ID 8786793*, 14 pages. https://doi.org/10.34133/2021/8786793.

[105] Dai, M., Li, S., Wang, Y., *et al.* (2019). Post-processing radio-frequency signal based on deep learning method for ultrasonic microbubble imaging. BioMed Eng OnLine 18, 95. https://doi.org/10.1186/s12938-019-0714-6.

[106] Hertel, R., & Benlamri, R. (2022). A deep learning segmentation classification pipeline for X-ray-based COVID-19 diagnosis. *Biomedical Engineering Advances*, *3*, 100041. ISSN 2667-0992. https://doi.org/10.1016/j.bea.2022.100041.

[107] Malik, H., Bashir, U., & Ahmad, A. (2022). Multi-classification neural network model for detection of abnormal heartbeat audio signals. *Biomedical Engineering Advances*, *4*, 100048. ISSN 2667992. https://doi.org/10.1016/j.bea.2022.100048.

[108] Lan, L., You, L., Zhang, Z., Fan, Z., Zhao, W., Zeng, N., Chen, Y., & Zhou, X. (2020). Generative adversarial networks and its applications in biomedical informatics. *Front Public Health*, *8*, 164. https://doi.org/10.3389/fpubh.2020.00164. PMID: 32478029; PMCID: PMC7235323.

[109] Asghar, U., Arif, M., Ejaz, K., Vicoveanu, D., Izdrui, D., & Geman, O. (2022). An improved COVID-19 detection using GAN-based data augmentation and novel QuNet-based classification. BioMed Research International, 2022, 1–9. https://doi.org/10.1155/2022/8925930.

[110] Riggs, A. D. (1975). X inactivation, differentiation, and DNA methylation. *Cytogenetic and Genome Researc*, *14*, 9–25.
[111] Nanney, D. L. (1958). Epigenetic factors affecting mating type expression in certain ciliates. *Cold Spring Harbor Symposia on Quantitative Biology*, *23*, 327–335.
[112] Okano, M., *et al.* (1998). Cloning and characterization of a family of novel mammalian DNA (cytosine-5) methyltransferases. *Nature Genetics*, *19*, 219–220.
[113] Maulik, U., *et al.* (2015). Analyzing gene expression and methylation data profiles using StatBicRM: Statistical biclustering-based rule mining. *PLoS One*, *10*, e0119448.
[114] Xin, S., *et al.* (2016). Cloning and functional analysis of the promoter of an ascorbate oxidase gene from Gossypium Hirsutum. *PLoS One*, *11*, e0161695.
[115] Moran, S., Arribas, C., & Esteller, M. (2016). Validation of a DNA methylation microarray for 850,000 CpG sites of the human genome enriched in enhancer sequences. *Epigenomics*, *8*, 389–399.
[116] ENCODE Project Consortium (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, *489*, 57–74.
[117] Siggens, L., & Ekwall, K. (2014). Chromatin and genome organization: Recent advances from the ENCODE project. *Epigenetics*, *276*, 201–214.
[118] Lizio, M., Harshbarger, J., Shimoji, H., *et al.* (2015). Gateways to the FANTOM5 promoter level mammalian expression atlas. *Genome Biology*, *16*.
[119] De Jager, P. L., Srivastava, G., & Lunnon, K., *et al.* (2014). Alzheimer's disease: early alterations in brain DNA methylation at ANK1, BIN1, RHBDF2 and other loci. Nature Neuroscience, 17, 1156–1163.
[120] Duan, L., Hu, J., Xiong, X., *et al.* (2018). The role of DNAmethylation in coronary artery disease. *Gene*, *646*, 91–97.
[121] Omura, N., & Goggins, M. (2009). Epigenetics and epigenetic alterations in pancreatic cancer. *International Journal of Clinical and Experimental Pathology*, *2*, 310–326.
[122] Lao, V. V., & Grady, W. M. (2011). Epigenetics and colorectal cancer. *Nature Reviews Gastroenterology & Hepatology*, *8*, 686–700.
[123] Liu, Y., Aryee, M. J., Padyukov, L., *et al.* (2013). Epigenome-wide association data implicate DNA methylation as an intermediary of genetic risk in rheumatoid arthritis. *Nature Biotechnology*, *31*, 142–147.
[124] Li, D., Xie, Z., Pape, M. L., *et al.* (2015). An evaluation of statistical methods for DNA methylation microarray data analysis. BMC Bioinformatics, 16.
[125] Zhang, Q., Zhao, Y., Zhang, R., *et al.* (2016). A comparative study of five association tests based on CpG set for epigenome-wide association studies. *PLoS One*, *11*.
[126] Jaffe, A. E., Murakami, P., Lee, H., *et al.* (2012). Bump hunting to identify differentially methylated regions in epigenetic epidemiology studies. *International Journal of Epidemiology*, *41*, 200–209.
[127] Pedersen, B. S., Schwartz, D. A., Yang, I. V., *et al.* (2012). Comb-p: Software for combining, analyzing, grouping and correcting spatially correlated P-values. *Bioinformatics*, *28*, 2986–2989.
[128] Butcher, L. M., & Beck, S. (2015). Probe Lasso: A novel method to rope in differentially methylated regions with 450K DNA methylation data. *Methods*, *72*, 21–28.
[129] Du, P., Zhang, X., Huang, C. C., *et al*. (2010). Comparison of Beta-value and Mvalue methods for quantifying methylation levels by microarray analysis. *BMC Bioinformatics*, *11*, 587.
[130] Satterthwaite, F. (1946). An approximate distribution of estimates of variance components. *Biometrics*, *2*, 110–114.
[131] Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing, *Journal of the Royal Statistical Society*, *57*, 289–300.
[132] Riley, J. W., Stouffer, S. A., & Suchman, E. A. (1949). *The American soldier: Adjustment during army life*. Princeton.
[133] Kechris, K. J., Biehs, B., Kornberg, T. B. (2010). Generalizing moving averages for tiling arrays using combined p-value statistic. *Statistical Applications in Genetics and Molecular Biology*, *9*.
[134] Wang, D., Yan, L., Hu, Q., *et al.* (2012). IMA: An R package for high-throughput analysis of Illumina's 450K Infinium methylation data. *Epigenetics Chromatin*, *28*, 729–730.
[135] Marsit, C. J., *et al.* (2011). DNA methylation array analysis identifies profiles of blood derived DNA methylation associated with bladder cancer. *Journal of Clinical Oncology*, *29*, 1133–1139.
[136] Rocke, D. M. (1993). On the beta transformation family. Technometrics, 35, 72–81.
[137] Kuan, P. F., *et al.* (2010). A statistical framework for Illumina DNA methylation arrays. *Bioinformatics*, *26*, 2849–2855.
[138] Teschendorff, A. E., *et al.* (2009). An epigenetic signature in peripheral blood predicts active ovarian cancer. *PLoS One*, *4*.
[139] Aryee, M. J., *et al.* (2011). Accurate genome-scale percentage DNA methylation estimates from microarray data. *Biostatistics*, *12*, 197–210.
[140] Kohl, M., *et al.* (2010). Preprocessing of gene expression data by optimally robust estimators. *BMC Bioinformatics*, *11*, 583.
[141] Belsley, D. A., *et al.* (1980). *Regression diagnostics: Identifying influential data and sources of collinearity*. Wiley.
[142] Mosteller, F., *et al.* (1977). *Exploratory data analysis and regression*. Addison Wesley.
[143] Sofer, T., Schifano, E. D., Hoppin, J. A., *et al.* (2013). A-clustering: A novel method for the detection of co-regulated methylation regions, and regions associated with exposure. *Bioinformatics*, *29*, 2884–2891.
[144] Liang, K. Y., *et al.* (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, *73*, 13–22.
[145] Izenman, A. (2008). Modern multivariate statistical techniques: Regression, classification, and manifold learning. Springer Publishing Company.
[146] Johnson, W. E., *et al.* (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics, 8, 118–127.
[147] Kolde, R., *et al.* (2016). seqlm: An MDL based method for identifying differentially methylated regions in high density methylation array data. *Bioinformatics*, *32*, 2604–2610.
[148] Lokk, K., *et al.* (2014). DNA methylome profiling of human tissues identifies global and tissue-specific methylation patterns. *Genome Biology*, *15*.

[149] Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, *14*, 465–471.

[150] Gomez, L., *et al.* (2019). coMethDMR: accurate identification of co-methylated and differentially methylated regions in epigenome-wide association studies with continuous phenotypes. Nucleic Acids Research, 47.

[151] Houseman, E. A., *et al.* (2008). Model-based clustering of DNA methylation array data: a recursive-partitioning algorithm for high-dimensional data arising as a mixture of beta distributions. *BMC Bioinformatics*, *9*, 365.

[152] Laird, P. W., *et al.* (2010). Principles and challenges of genomewide DNA methylation analysis. *Nature Reviews Genetics*, *11*, 191–203.

[153] Siegmund, K. D., *et al.* (2011). Statistical approaches for the analysis of DNA methylation microarray data. *Human Genetics*, *129*, 585–595.

How to Cite: Mukherji, A., Mondal, A., Banerjee, R., & Mallik, S. (2023). Recent Landscape of Deep Learning Intervention and Consecutive Clustering on Biomedical Diagnosis. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA2202480