Journal of Big Data

RESEARCH Open Access

Application of deep learning technique in next generation sequence experiments

Su Özgür[1*] and Mehmet Orman[1]

*Correspondence:
suozgur35@gmail.com

[1] Department of Biostatistics and Medical Informatics, Ege University Faculty of Medicine, 35100 Bornova-İzmir, Turkey

Abstract

In recent years, the widespread utilization of biological data processing technology has been driven by its cost-effectiveness. Consequently, next-generation sequencing (NGS) has become an integral component of biological research. NGS technologies enable the sequencing of billions of nucleotides in the entire genome, transcriptome, or specific target regions. This sequencing generates vast data matrices. Consequently, there is a growing demand for deep learning (DL) approaches, which employ multi-layer artificial neural networks and systems capable of extracting meaningful information from these extensive data structures. In this study, the aim was to obtain optimized parameters and assess the prediction performance of deep learning and machine learning (ML) algorithms for binary classification in real and simulated whole genome data using a cloud-based system. The ART-simulated data and paired-end NGS (whole genome) data of Ch22, which includes ethnicity information, were evaluated using XGBoost, LightGBM, and DL algorithms. When the learning rate was set to 0.01 and 0.001, and the epoch values were updated to 500, 1000, and 2000 in the deep learning model for the ART simulated dataset, the median accuracy values of the ART models were as follows: 0.6320, 0.6800, and 0.7340 for epoch 0.01; and 0.6920, 0.7220, and 0.8020 for epoch 0.001, respectively. In comparison, the median accuracy values of the XGBoost and LightGBM models were 0.6990 and 0.6250 respectively. When the same process is repeated for Chr 22, the results are as follows: the median accuracy values of the DL models were 0.5290, 0.5420 and 0.5820 for epoch 0.01; and 0.5510, 0.5830 and 0.6040 for epoch 0.001, respectively. Additionally, the median accuracy values of the XGBoost and LightGBM models were 0.5760 and 0.5250, respectively. While the best classification estimates were obtained at 2000 epochs and a learning rate (LR) value of 0.001 for both real and simulated data, the XGBoost algorithm showed higher performance when the epoch value was 500 and the LR was 0.01. When dealing with class imbalance, the DL algorithm yielded similar and high Recall and Precision values. Conclusively, this study serves as a timely resource for genomic scientists, providing guidance on why, when, and how to effectively utilize deep learning/machine learning methods for the analysis of human genomic data.

Keywords: Next generation sequencing, Deep learning, Machine learning, Variant calling format, Cloud computing

Springer Open

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Introduction

With the widespread use of biological data processing technology and the rapid advancement in high-throughput sequencing (HTS) technologies, especially Illumina systems, next-generation sequencing (NGS) technology has become an indispensable part of biological research in many areas [1]. Next-generation sequencing technologies enable the sequencing of billions of nucleotides in the entire genome, transcriptome or smaller target regions. Therefore, the growth in data volume gives rise to extremely large data matrices. Systems that detect meaningful information from very large data structures have increased the need for deep learning (DL) approach which uses multilayer artificial neural networks (ANN). This situation has led researchers to utilize advanced statistical methods instead of classical statistical approaches in their studies.

The quality of machine learning (ML) approaches depends on selecting the appropriate features [2]. Various preprocessing, dimensionality reduction, and feature selection techniques are employed to uncover these features. To reduce computation time and increase accuracy, it is essential to reduce dependence on specific features at this stage. Deep learning algorithms aim to classify and describe data by extracting features that can provide more information from individually less informative variables. Unlike traditional machine learning methods, DL methods provide a significant advantage in solving problems in high-dimensional data matrices and analyzing such data [3]. Performing hyperparameter optimization is crucial for creating an effective model and determining the optimal architecture and parameters [4–6].

Next-generation sequencing (NGS) methods have been at the center of numerous biological and medical research and have become very popular topics in recent years with deep learning algorithms [1]. Especially since feature extraction is not possible in genetic data analysis, the application of deep learning techniques in this field is important for researchers to obtain more accurate results. In the existing literature, no studies have been identified that specifically investigate the optimized parameter evaluation of algorithms in NGS data. Therefore, it is necessary to obtain optimized values of hyperparameters, such as epoch, the number of layers, learning rate, and batch size, in NGS data analysis. In the literature, there are a limited number of studies that have performed diagnosis or classification using machine learning or deep learning techniques on various types of genetic data, including exome, metagenomic, and omics data. The convolutional neural network method (CNN) was utilized for the identification of clathrin proteins, the deficiency of which in the human body leads to significant neurodegenerative diseases such as Alzheimer's [6]. Deep Neural Network (DNN) and XGBoost algorithms were used to classify variants into two classes which are somatic and germline, for a given whole exome sequencing (WES) data [7]. Performance comparisons were conducted between ML and DL algorithms to predict the effects of non-coding mutations on gene expression and DNA [8]. By utilizing TCGA data as input, a deep learning algorithm was used for the model of the association between genes and their corresponding proteins in relation to survival prognosis [9].

Deep learning techniques have recently emerged as powerful tools for various biomedical applications, notably in the realm of Next-Generation Sequencing. The exponential growth in genomic data produced by NGS platforms has presented both challenges and opportunities. Traditional bioinformatics methods often struggle to efficiently process

and interpret the vast quantities of data generated. In contrast, deep neural networks (DNNs) have shown significant promise in detecting complex patterns, predicting phenotypes, and classifying genomic variants, among other tasks (Fig. 1) [10].

The Deep Neural Network (DNN) is a subfield of machine learning algorithms that models the workings of the biological nervous system. In a DNN model, there are multiple layers, including input and output layers, as well as more than two hidden layers, each containing neurons (processing nodes). These hidden layers are crucial components of the DNN model and actively participate in the learning process. While using more hidden layers during training can enhance the model's performance, it can also introduce significant challenges such as model complexity, computational cost, and overfitting. One of the remarkable capabilities of the DNN model is its ability to automatically extract relevant features from unlabeled or unstructured datasets using standard learning procedures. Several researchers have reported that DNN models outperform traditional learning methods in various complex classification problems. Therefore, in various domains, DNN models can achieve highly accurate prediction performance, especially in classification problems involving intricate relationships [11].

Recurrent Neural Networks (RNNs) are a type of artificial neural network designed to process sequential or time series data. Unlike conventional neural networks, which assume independence between inputs and outputs, RNNs operate on sequences, performing a similar task for each element in the sequence while taking into account previous outputs. However, the widespread utilization of RNNs in DNA sequencing data, where the order of bases holds crucial significance, has been limited [10]. Maraziotis et al. pioneered the implementation of RNNs in genomics, utilizing microarray experimental data and employing a recurrent neuro-fuzzy protocol to infer complex causal relationships among genes by predicting time series gene expression patterns. Although most RNN applications in genomics are combined with other algorithms like Convolutional Neural Networks (CNNs), CNNs excel in capturing local DNA sequence patterns, whereas RNN derivatives are more adept at capturing long-range dependencies within sequence datasets [12].

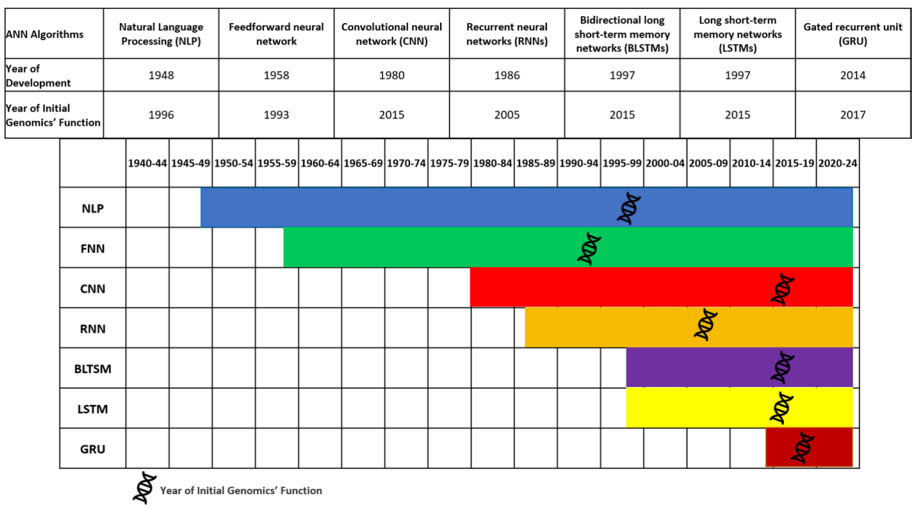

Fig. 1 Timeline of implementing deep learning algorithms in Genomics [10]

A convolutional neural network (CNN) is a deep learning algorithm characterized by a deep feedforward architecture comprising various building blocks, including convolution layers, pooling layers, and fully connected layers. It can be visualized as a fully connected network, where each node in a single layer is connected to every node in the next layer. In CNN layers, convolution units process input data from units in the previous layer, collectively contributing to making predictions. The fundamental principle behind this deep architecture is to enable extensive processing and connection features, allowing the network to capture complex nonlinear associations between inputs and outputs. Due to these features that effectively define linear relationships, CNNs have found applications in a wide range of fields, including medicine, genetics, engineering, and economics [13].

Deep Reinforcement learning (DRL) is a machine learning technique in which a computer agent learns to perform a task through repeated trial-and-error interactions with a dynamic environment. This learning approach empowers the agent to make a series of decisions aimed at maximizing a reward metric for the task, all without human intervention and without being explicitly programmed to achieve the task. Studies of RL in the field of genetics are quite limited, and the first applications seem to be aimed at solving DNA sequence alignment using the Markov decision process (MDP) [14].

As the scale of genetic data expands, there will be an increase in costs and time associated with data processing. This situation leads to an increase in demands such as data analysis and fast delivery of findings at low costs.

Furthermore, it is important to present the optimized parameters obtained from methods such as ML and DL applied to different types of genetic data to practitioners in the field. This allows for performance evaluations and ensures the maximum information can be obtained from the data.

In this study we have used GPU based model training but there are several computational environment options for deep learning applications. For instance: SPARK, High Performance Computing (HPC), Field-Programmable Gate Array (FPGA). Khan S. et al. and Xueqi L. et al. reported that there are limitations on high I/O latency, distributed compute memory maximization, optimization of configurable parameters and maintaince of the clusters [15, 16].

The goal of this study was to obtain optimized parameters and evaluate the prediction performance of deep learning and machine learning (ML) algorithms for binary classification in both real and simulated whole-genome data using a cloud-based system. In this study we explored the following question: "Is GPU infrastructure based algorithm (DL) performs better than CPU based ML algorithms in terms of accuracy, time and repeatability?".

Next-generation sequencing

The human genome (Deoxyribonucleic Acid, DNA) consists of around 3 million nucleotides. There are four nucleotides in DNA: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). Only about 2% of DNA encode proteins. These DNA fragments that encode proteins are called exons, and the combination of all exons within the genome is known as the exome. The remaining parts of DNA are expressed as intergenic regions (introns) that do not encode protein.

Damage to DNA can result in various consequences such as malformations, cancer, aging, genetic alterations, and cell death [17]. Therefore, the early detection of DNA damage plays anincreasingly important role in diagnosis, treatment, and the quality of life for patients.

It has been determined that most of the mutations that lead to the formation of diseases occur in the exon regions of DNA [18].

Next-generation sequencing (NGS) is a method that is based on the simultaneous and parallel processing of each part of a DNA molecule obtained from a single sample, which is divided into millions of parts. In other words, NGS is the process of determining the order of nucleotide bases in an individual's DNA molecule. NGS technology can detect genetic variants in an individual's DNA that may be associated with a disease. However, technical limitations may cause false negative results as they affect the diagnostic process of diseases [19]. In addition, the lack of sequence depth also changes the reliability of the detected variants. Although whole exome sequencing is a powerful method for diagnosis, it should not be considered the best approach for all clinical indications. However, it is the most important step in establishing the necessary associations for the detection of clinical findings and the resulting phenotype variants [20].

ART: a next-generation sequencing read simulator

The ART simulator is a group of methods that can generate data exactly the same as Illumina technology, including erroneous reads that may occur in real genomes. ART software was primarily developed for simulation studies helping to design data collection modalities for the 1000 Genomes Project. ART simulates sequencing reads by mimicking real sequencing processes with empirical error models or quality profiles summarized from large recalibrated sequencing data. Moreover, ART can simulate reads using the user's own read error model or quality profiles [21, 22].

Whole genome data of human chromosome 22

In this research, the second dataset used was real data (whole genome) of chromosome 22, which includes ethnicity information. This dataset was prepared by the Microsoft Genomics team and made publicly available for use.

The individuals in this dataset consist of five different populations, which are as follows: British from England and Scotland (91 individuals), Finnish from Finland (99 individuals), Colombian from Medellin (94 individuals), Chinese (103 individuals), and individuals with African ancestry from the Southwest USA (61 individuals). The data from these countries were categorized into 190 individuals of European ancestry and 258 individuals of non-European ancestry by expert geneticists. The dataset consists of 448 FASTQ files, with each file containing individual variants on chromosome 22 of the human genome. VCF data was generated based on the Human Genome 38 (GRCh38) reference genome from the raw FASTQ data [23].

Cloud computing

Studies based on large sequencing datasets are growing rapidly, and public archives for raw sequencing data are periodically doubled. Researchers need to use large-scale computational resources to use this data. Cloud computing, a model where users rent

computers and storage from large data centers, is an attractive solution for genome research. Particularly in genetic research, conducting analyses directly on the stored data not only saves time but also reduces costs associated with data transfer across platforms [24]. We have implemented our pipeline on Microsoft Azure cloud VMs and Jupyter notebooks.

Methods

In this section, we introduce the proposed best practice pipeline for the classification of Next Generation Sequencing data. Firstly, we constructed a dataset to simulate the entire Human Genome.

Secondly, we obtained the VCF data by aligning the real Chr 22 whole genome FASTQ data shared by Microsoft Genomics Team with the reference genome using the BWA-GATK tool, which the Broad Institute defines as the best practice.

Datasets

ART simulation data set

The distributions of the number of variants for two different continental groups (European and Others), as reported by the 1000 Genomes Project, were used to generate the variant types in the simulation. This approach ensured that the simulated data closely resembled real individuals.

In this study, NGS reads based on synthetic human genomes were derived using one of the most commonly used methods in genetic data simulation: a next-generation sequencing read simulator (ART) [21]. As a result, this study produced "500 data for group 0" and "500 data for group 1". The distinction between the groups was achieved by changing the f and m parameters. The average simulation time for generating a whole genome FASTQ paired-end data took approximately 4 h and 12 min using the virtual computer configurations employed in this study. The simulations were made in batches of 100 on 10 different virtual machines [25]. The following are the codes used to generate the simulated data:

- art_illumina.exe -ss HS25 -i./testSeq.fa -o./paired_end_com -l 150 -f 5 -p -m 250 -s 10 (for group 0)
- art_illumina.exe -ss HS25 -i./testSeq.fa -o./paired_end_com -l 150 -f 10 -p -m 500 -s 10 (for group 1)

Chromosome 22 WGS data set

In the study, NC24, one of the NC series virtual machines supported by NVIDIA Tesla K80 Card and Intel Xeon E5-2690 V3 processor, was used. The analyses were conducted using Python programming language. The paired-end Next Generation Sequencing (NGS) data of Chromosome 22 (Ch22) in FastQ format was obtained using Illumina NextSeq 500. After performing quality control, the data was aligned to the reference genome (GRCh38) using the Burrows-Wheeler Aligner (BWA) method, which is part of the Broad Institute's best practices analysis pipeline. By applying the Genome Analyzer Tool Kit (GATK) method, which is the most frequently used pipeline for Variant

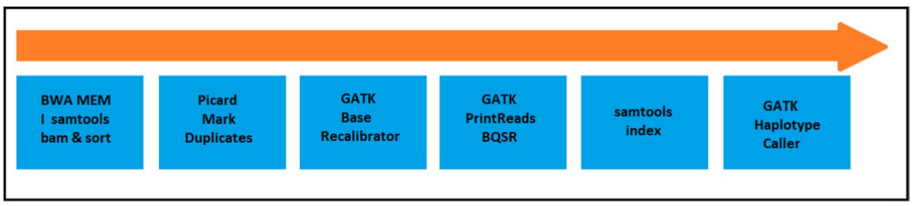

Fig. 2 Standard BWA/GATK flow chart

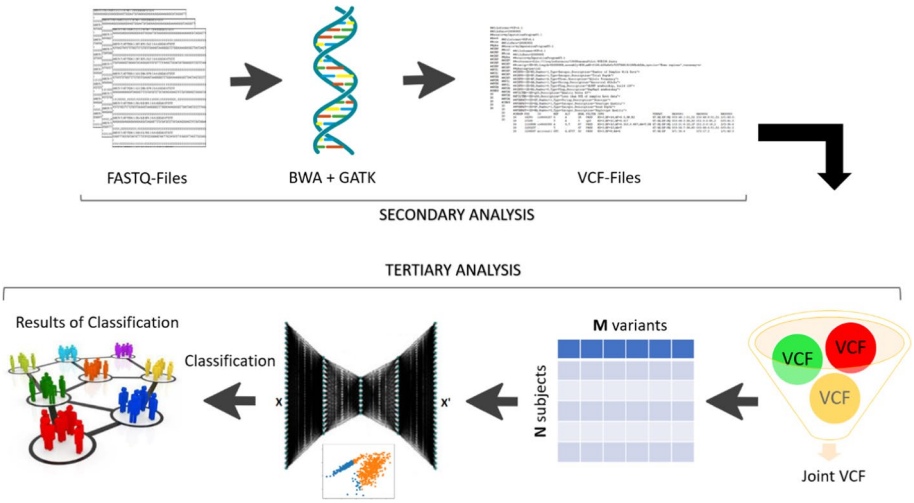

Fig. 3 Pipeline [28]

Calling, to the aligned data, Variant Calling Format (VCF) data describing the variants were obtained (Fig. 2, Additional file 1: Table S1) [26].

Secondary analysis

This section involves that the process where the produced reads of the individual's exome or genome are aligned to the reference genome and variant calls are generated. The first of the limitations at this stage is the lack of available human reference genomes and the lack of consensus on which optimal reference genome to use. Several software has been developed to realize this reading process. Various platforms such as BWA, Novalign, Stampy, SOAP2, LifeScope, and Bowtie are frequently used. As a result of this process, a BAM file is created as output (Fig. 3).

FASTQ

FASTQ is a text-based file format that contains nucleotide sequence reads and quality scores for each nucleotide read [27]. A typical FASTQ file contains 4 lines: The first line starts with the '@' character and specifies the identity of the sequence. The second line contains the raw sequence data, represented by a font. The third line starts with symbol plus "+" and can be optionally blank, or optionally followed by the sequence identifier which in the first line is written. In the fourth line, the quality value of the sequence is displayed in ASCII format. The quality value shows the probability of the sequence misreading during reading. Higher quality scores indicate a smaller probability of error

(p_{error}). The phred-scaled quality score (Q) is converted to probability with the formula as $Q = -10\log_{10}{}^{*}p_{error}$.

Variant calling&GATK

Variant calling stage entails identifying single nucleotide polymorphisms (SNPs), multiple nucleotide polymorphisms (MNPs), small insertions and deletions (InDels, they are usually less than 50 bp) from next generation sequencing data [29]. In this process, between 20,000 and 100,000, variants are discovered per exome, and approximately 3–4 million variants for whole genome sequencing.

The Variant Call Format (VCF) is a text file that contains information about the variants found between the reference genome and the sample genome. The VCF format was developed for the 1000 Genomes Project. A VCF file consists of 8 fixed and mandatory columns, which are as follows: # chromosome (CHROM), a 1-based position of the start of the variant (POS), unique identifiers of the variant (ID), the reference allele (REF), a comma-separated list of alternate non-reference alleles (ALT), a phred-scaled quality score (QUAL), site filtering information (FILTER), and a semicolon-separated list of additional, user-extensible annotations (INFO) [30].

Experience has shown that software developed based on Bayesian statistical probability methods, such as SAMtools and the Genome Analysis Toolkit (GATK) (https://gatk.broadinstitute.org/hc/en-us), are frequently preferred for their ability to reduce sequencing errors [31]. In this study, GATK, which was developed by the Broad Institute, was used for variant discovery following alignment with BWA.

Burrows-Wheeler Aligner, BWA

Alignment tool (Burrows-Wheler Aligner, BWA) is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM. BWA-backtrack algorithm is designed for Illumina sequence reads up to 100 bp, while BWA-MEM and BWA-SW for longer sequences ranged from 70 bp to 1Mbp. BWA-MEM and BWA-SW have similar characteristics such as long-read support and split alignment. However, the BWA-MEM algorithm is faster and it provides more accurate results for high-quality queries. BWA-MEM also has a better algorithm than BWA-backtrack for 70–100 bp Illumina reads. In this study, the BWA-MEM algorithm was used for alignment [32].

Tertiary analysis

This is the third and final step of the NGS analysis workflow. After merging the VCF data of individuals (Joint VCF), a matrix is created with individuals in rows and variants in columns. At the final stage, techniques such as machine learning, deep learning, and clustering are applied to this VCF matrix. Generally, this step includes the annotation of genes, mutations and transcripts. But it is focused to obtain the prediction performance and optimized parameters of deep learning and machine learning algorithms for the “Binary Classification” in real and simulated whole genome data using a cloud-based system in this study. Because the most important problems in genetic data are the storage, organization and modeling of this data. Therefore, it does not include a process related to "annotation" (Fig. 3).

Machine learning methods

XGBoost

XGBoost (eXtreme Gradient Boosting) algorithm is a high-performance version of the Gradient Boosting algorithm optimized with various arrangements. It was introduced by Tianqi Chen and Carlos Guestrin in the article "XGBoost: A Scalable Tree Boosting System" published in 2016. The most important characteristics of the algorithm are its high predictive power, preventing over-learning, handling missing data and at the same time performing these operations quickly. According to Tianqi, XGBoost runs 10 times faster than other popular algorithms. It is shown as the best of the decision tree-based algorithms (Table 1) [33].

LightGBM

LightGBM is a high-performance gradient boosting algorithm using a tree-based learning algorithm designed by Microsoft Research Asia in the Distributed Machine Learning Toolkit (DMTK) project in 2017 (https://lightgbm.readthedocs.io/en/latest). This algorithm has some advantages over boosting algorithms. These advantages are; solving prediction problems related to big data more effectively, using fewer resources (RAM), high prediction performance, and parallel learning [33]. It is very fast, therefore it is defined by the expression "Light". In the article (A Highly Efficient Gradient Boosting Decision Tree), LightGBM was found to be 20 times faster than other algorithms [34].

In the LightGBM algorithm, optimizing the *learning rate*, *max dept*, *num leaves*, *min data* in leaf parameters to prevent overlearning and feature fraction, *bagging fraction* and *num iteration* parameters to accelerate the learning time increases the performance of the model (Table 1, Fig. 4).

XGBoost utilizes a level-wise tree construction strategy, building the tree in a level-by-level manner. In contrast, LightGBM adopts a leaf-wise tree construction strategy, where the tree is grown by continuously splitting the leaf with the highest gain. This leaf-wise strategy in LightGBM often results in faster training times. It is noteworthy that although XGBoost and LightGBM share similar concepts and objectives as gradient boosting frameworks, the variations in their implementations contribute to differences in performance, speed, and memory efficiency between the two algorithms [35].

Deep learning

Deep learning is a subset of artificial intelligence and machine learning that uses multi-layer artificial neural networks to make predictions with high sensitivity and accuracy in areas such as image processing, object detection, and natural language processing. With the widespread use of biological data processing technology, NGS technology has become an indispensable part of biological research in many fields. It has been reported that there will be 100 million NGS data in the estimates made for 2025 (Fig. 5) [36].

It is not possible to extract features from these structures with classical approaches. For this reason, systems that evaluate many layers at the same time and detect meaningful information from large data structures have increased the need for a deep learning approach using multi-layer artificial neural networks.

Deep learning requires the use of many hidden neurons and layers with new training models. The use of large numbers of neurons allows for a comprehensive representation of the raw data available. Adding more hidden layers to the neural network allows the hidden layers to capture nonlinear relationships. Thus, when the neural network is optimally weighted, high-level representations of the obtained raw data or images are provided [37].

In the tertiary analysis phase, Convolutional Neural Networks (CNN), one of the deep learning architectures, were used. CNNs are applied in various fields such as image recognition, video recognition, natural language processing, and computational biology. CNN is a variant of multi-layer perceptron (MLP) (Fig. 6).

Deep learning in next-generation sequencing

Genomics is advancing towards a data-driven scientific approach. With the emergence of high-throughput data generation technologies in human genomics, we are confronted with vast amounts of genomic data. Multiple genomic disciplines, such as variant calling and annotation, disease variant prediction, gene expression and regulation, epigenomics, and pharmacogenomics, benefit from the generation of high-throughput data and the utilization of deep learning algorithms to enable sophisticated predictions. Deep learning utilizes a wide range of parameters, which can be optimized through training on labeled data, particularly in the context of genetic datasets. Deep learning has the advantage of effectively modeling a large number of differentially expressed genes. There are still a limited number of studies in the literature evaluating NGS data with the deep learning method [38].

Using TCGA data as input, Wong et al. utilized deep learning to model the relationship between genes and their corresponding proteins in relation to survival prognosis [9]. They presented a model which identifies different genes associated with glioblastoma survival, glioblastoma cancer cell migration, or glioblastoma stem cells. In another study, Young et al. used a deep learning algorithm to classify glioblastomas into six subtypes in patient survival [39].

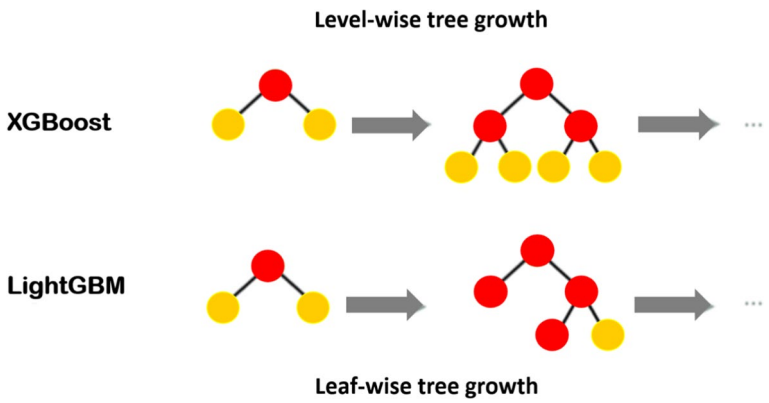

Fig. 4 LightBGM and XGBoost

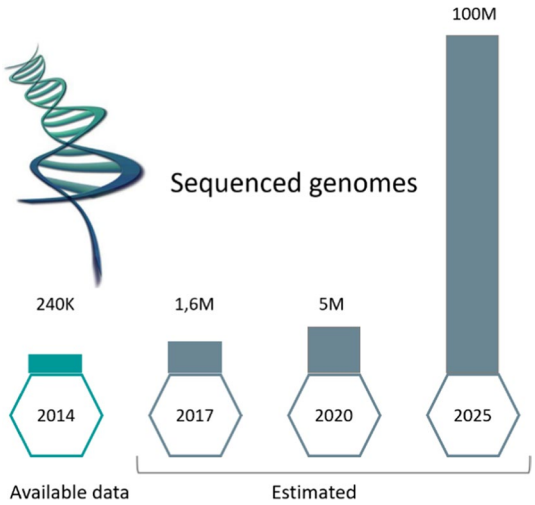

Fig. 5 Sequenced and estimated genomes

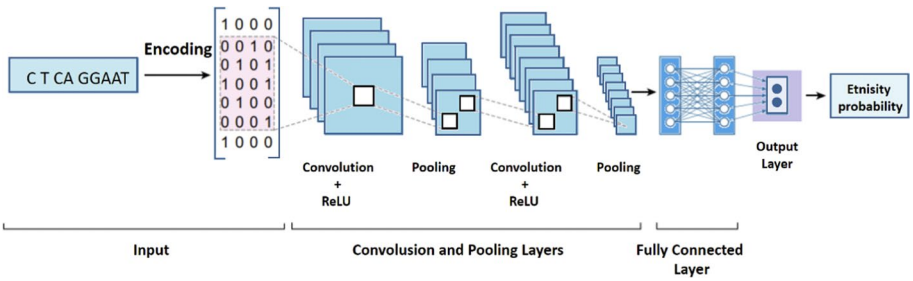

Fig. 6 Convolutional neural network structure [1]

When considering such examples, new possibilities may arise for the early diagnosis of diseases as the use of deep learning in complex data structures, such as Next-Generation Sequencing (NGS), increases.

Deep learning hyperparameters

Batch size

Processing big data sets at once takes a long time and leads to memory problems. The data set is divided into small samples to prevent wasting time and memory problems, and the learning process is performed from these small pieces. The batch size defines the number of samples that will be propagated through the network [40].

Learning rate

Learning rate (LR) or step size is defined as the amount that the weights are updated during training. This learning structure can be realized in different ways. The LR parameter used during this process can be selected as a fixed value or as an incremental value. For example, it can be done by taking 0.001 until a certain learning step of the algorithm and taking 0.01 after this step. If this parameter is selected too small, the learning rate will also be slow. The larger the value of the parameter, the greater the impact of the

data on the algorithm. For this reason, it is recommended to keep this value high at the beginning of the process and to decrease it after a certain epoch [41].

Epoch

Deep learning is used to make predictions in big data structures. Due to the large size of the matrices, data is divided into smaller parts and processed in parts rather than training the entire dataset at once. The number of epochs is a hyperparameter that determines the number of times the learning algorithm will iterate over the entire training dataset (*If you have a training dataset with 1000 examples and set a batch size of 10, it will take 100 iterations to complete one epoch*.). This comprises one instance of a forward pass and backpropagation. As the number of epochs increases, the network's accuracy increases. Performance improvements in terms of accuracy tend to diminish or plateau after a certain number of epochs. When the training reaches the desired level (the error value, the point where the accuracy value is optimal), it can be terminated [42].

Number of layers

One of the most important features that distinguishes the deep learning algorithm from other artificial neural network algorithms is the number of layers, which enables it to successfully handle complex problems (Fig. 7). Increasing the number of layers improves the learning performance of the model. Thus, during the process of weight updates

Table 1 Parameters used in ML and DL for ART simulation data and Chr 22 WGS data sets

Parameters	ART simulation data	Chr 22 WGS data
Drop-out	0.2 (800 train set/200 test set)	0.2 (360 train set/88 test set)
Deep learning		
Batch size	160	60
Epoch	500; 1000; 2000	500; 1000; 2000
Number of iterations for each epoch in the model	5	6
Iteration	10,000	12,000
Learning rate (LR)	0.01; 0.001	0.01; 0.001
LightGBM		
Min data in leaf	100	100
Max depth	7	5
Num leaves	128	32
Num iterations	100	100
Learning rate (LR)	0.01	0.001
Bagging fraction	0.5	0.5
XGBoost		
Eta	0.01	0.015
Min child weight	1.4	1
Max depth	5	3
Gamma	0.1	0.1
Alpha	0.001	0.001
Lambda	1	1
Subsample	0.8	0.8

through backpropagation, the effect of these updates on the first layers will be reduced [43].

The parameters used in the algorithm are presented in Table 1.

Performance evaluation

The evaluation criteria used to measure the predictive performance of models; recall, accuracy, precision, AUC-ROC, F criteria [37].

- Precision = TP/(TP + FP)
- Recall = TP/(TP + FN)
- F-Measure = (2 × Precision × Recall)/(Precision + Recall)
- Accuracy = (TN + TP) (TN + FP + TP + FN)

TP: True Positive, FP: False Positive, FN: False Negative, TN: True Negative.

Results

The study presents the findings of the ART simulated data, which consider the distributions of the variant numbers for two different continental groups (European and Non-European) as reported by the 1000 Genomes Project. These findings are summarized below.

When the learning rate was set to 0.01 and the epoch was updated with values of 500, 1000, and 2000 in the deep learning model for the ART simulated data, the mean accuracy values of the models were 0.6319 ± 0.0065, 0.6804 ± 0.0090, and 0.7333 ± 0.0167, respectively. The median accuracy values of the models were 0.6320 [0.6210–0.6430], 0.6800 [0.6650–0.6960], and 0.7340 [0.7060–0.7630], respectively. As the epoch value increased, the average accuracy value of the model also increased (Table 2, Additional file 1: Table S2, and Fig. 7) (Table 2, Additional file 1: Table S2, Fig. S1).

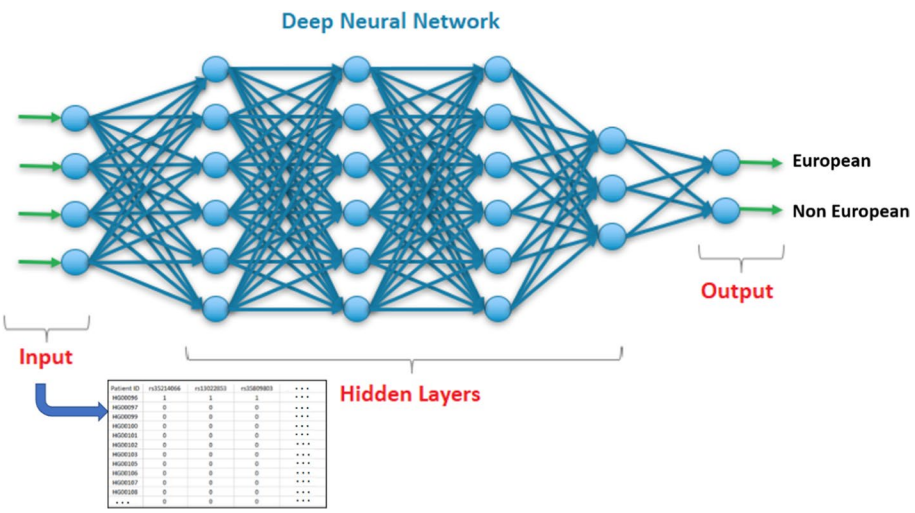

Fig. 7 Deep neural network

Table 2 Performance comparison of deep learning and machine learning algorithms on ART simulation data set

	XGBoost					LightGBM					Deep learning				
	Accuracy	**AUC**	**Recall**	**Precision**	**F-Score**	**Accuracy**	**AUC**	**Recall**	**Precision**	**F-Score**	**Accuracy**	**AUC**	**Recall**	**Precision**	**F-Score**
$\overline{X}$	0.6987	0.707	0.6777	0.6957	0.6846	0.6258	0.651	0.6473	0.6265	0.6289	0.8014	0.8067	0.7846	0.7946	0.7866
$\widetilde{X}$	0.699	0.708	0.678	0.695	0.684	0.625	0.65	0.647	0.627	0.629	0.8020	0.8060	0.7840	0.7940	0.7860
Std Dev	0.0081	0.0141	0.0093	0.0119	0.013	0.0096	0.0145	0.0093	0.0087	0.002	0.0386	0.0455	0.0468	0.0468	0.0468
Min	0.685	0.682	0.662	0.676	0.663	0.61	0.627	0.632	0.611	0.626	0.7360	0.7280	0.7040	0.7140	0.7060
Max	0.712	0.731	0.693	0.717	0.708	0.643	0.677	0.664	0.641	0.633	0.8690	0.8840	0.8650	0.8750	0.8670

$\overline{X}$, mean; $\widetilde{X}$, median; Std Dev, standard deviation; Min, minimum; Max, maximum

Secondly, the learning rate was decreased to 0.001, and the effect of epoch value on the model was investigated. When the epoch values were updated with 500, 1000, and 2000 in the deep learning models, the mean accuracy values of the models were 0.6922 ± 0.0168, 0.7214 ± 0.0182, and 0.8014 ± 0.0386, respectively. The median accuracy values of the models were 0.6920 [0.6640–0.7210], 0.7220 [0.6910–0.7530], and 0.8020 [0.7360–0.8690], respectively. As the epoch value increased, the accuracy of the model increased. It was found that the accuracy performance of the deep learning model increased as the learning rate decreased and the epoch value increased (Table 2, Additional file 1: Table S2 and Fig. S1).

The table presents the performance of XGBoost and LightGBM model parameters. The average accuracy value of the XGBoost model was 0.6987 ± 0.0081, and the median was 0.6990 [0.6850–0.7120]. On the other hand, the average accuracy value of the LightGBM model was 0.6258 ± 0.0096, and the median was 0.6250 [0.6100–0.6430]. It can be observed that XGBoost has a higher accuracy compared to LightGBM (Table 2, Additional file 1: Table S2 and Fig. 7).

Increasing the epoch values resulted in higher accuracy performances when using DL algorithms with low LR values, compared to machine learning algorithms. The performance of DL algorithms improved with high LR and epoch values, while lower accuracy values were observed with DL algorithms with low epoch values, in comparison to machine learning algorithms. In particular, results with the XGBoost algorithm showed a performance close to the performances obtained with the DL algorithm at high epoch and LR values (Table 3, Additional file 1: Table S3 & Fig. S2).

When the LR was set to 0.01 and the epoch was updated with values of 500, 1000, and 2000 in the deep learning model for Chr 22 WGS data, the mean accuracy values of the models were 0.5289 ± 0.0106, 0.5421 ± 0.0059, and 0.5820 ± 0.0089, respectively. The median accuracy values of the models were 0.5290 [0.5110–0.5470], 0.5420 [0.5320–0.5520], and 0.5820 [0.5670–0.5980], respectively (Table 3, Additional file 1: Table S3 and Fig. S2).

On the other hand, when the learning rate of the deep learning model was set to 0.001 and the epoch value was updated to 500, 1000, and 2000, the mean accuracy values of the models were 0.5508 ± 0.0074, 0.5841 ± 0.0079, and 0.6045 ± 0.0044, respectively. The median accuracy values were 0.5510 [0.5380–0.5640], 0.5830 [0.5710–0.5980], and 0.6040 [0.5970–0.6120], respectively (Table 3, Additional file 1: Table S3 and Fig. S2).

When evaluating the performances of machine learning algorithms for Chr 22 WGS data, the mean and median accuracy values of the XGBoost model were determined as 0.5760 ± 0.0081 and 0.5760 [0.5730–0.5790], respectively. On the other hand, the mean and median accuracy values of the LightGBM algorithm were 0.5250 ± 0.0029 and 0.5250 [0.5200–0.5300], respectively. The performance of the XGBoost algorithm was higher than that of the LightGBM algorithm (Table 3, Additional file 1: Table S3 and Fig. S2).

Discussion

In this study, the prediction performance of deep learning and machine learning algorithms was demonstrated for ART simulation data and Chr 22 whole genome data, specifically focusing on "bivariate classification." The experiments were conducted using a

Table 3 Performance comparison of deep learning and machine learning algorithms on Chr 22 WGS data set

	XGBoost					LightGBM					Deep learning				
	Accuracy	**AUC**	**Recall**	**Precision**	**F-Score**	**Accuracy**	**AUC**	**Recall**	**Precision**	**F-Score**	**Accuracy**	**AUC**	**Recall**	**Precision**	**F-Score**
$\overline{X}$	0.5760	0.5743	0.5613	0.5695	0.5635	0.5250	0.5403	0.5441	0.5780	0.5680	0.6045	0.5979	0.6096	0.6168	0.6141
$\widetilde{X}$	0.5760	0.5740	0.5610	0.5695	0.5630	0.5250	0.5410	0.5440	0.5780	0.5682	0.6040	0.5980	0.6090	0.6170	0.6140
Std Dev	0.0018	0.0092	0.0035	0.0015	0.0021	0.0029	0.0095	0.0061	0.0067	0.0007	0.0044	0.0064	0.0060	0.0177	0.0104
Min	0.5730	0.5590	0.5550	0.5670	0.5600	0.5200	0.5230	0.5340	0.5670	0.5670	0.5970	0.5870	0.6000	0.5870	0.5970
Max	0.5790	0.5900	0.5670	0.5720	0.5670	0.5300	0.5560	0.5550	0.5900	0.5690	0.6120	0.6090	0.6200	0.6480	0.6320

$\overline{X}$, mean; $\widetilde{X}$, median; Std Dev, standard deviation; Min, minimum; Max, maximum

cloud-based system, and optimized parameters were obtained. The storage, organization, and modeling of genetic data are among the most critical problems. The use of cloud systems accelerates researchers in these stages. The research demonstrated the impact of hyperparameter changes in deep learning models. Furthermore, the performance of deep learning models was compared with popular machine learning algorithms such as XGBoost and LightGBM. Additionally, this study represents an innovative approach in terms of parameter optimization and performance evaluation on whole genome data using a cloud-based system.

Le et al. [6] utilized the deep learning method to identify clathrin proteins, the deficiency of which in the human body leads to significant neurodegenerative diseases like Alzheimer's. They employed the convolutional neural network method (CNN) and selected hyperparameters as follows: epoch = 80, LR = 0.001, batch size = 10, dropout = 0.2. The model's performance was evaluated using both machine learning and deep learning methods. The model exhibited a sensitivity of 92.2%, specificity of 91.2%, accuracy of 91.8%, and Matthews's correlation coefficient of 0.83 on the independent dataset. While our study yielded similar findings to Le et al., we additionally presented model performances at different epoch values (500–2000) and LR values (0.01–0.001). Consequently, we demonstrated that the deep learning method can achieve significantly higher performance levels than machine learning algorithms, particularly at higher epoch values [6].

Akker et al. [42] have developed a machine learning model that determines the accuracy of variant calls in captured-based next-generation sequencing. The model was tuned to eliminate false positives, which are variants identified by NGS but not confirmed by Sanger sequencing. They achieved an exceptionally high accuracy rate of 99.4%. In this study, it has been shown that NGS data has relevant properties to distinguish variables with low and high confidence using a machine learning-based model. Researchers did not focus on hyperparameter optimization in this study. Moreover, providing high discrimination in low coverage NGS data, which is smaller than the whole genome sequencing data, by using a machine learning algorithm is aligned with the findings of our study [42].

Marceddu et al. used a dataset of 7976 NGS calls validated as true or false positive by Sanger sequencing to train and test different ML approaches. While gradient boosting classifier (GBC), random forest (RF), and decision tree (DT) algorithms were less affected by the imbalance in the dataset, the prediction performance of linear support vector machine (LSVM), nearest neighbor (NN), and linear regression (LR) were significantly more affected. It has also been shown that for medium-small datasets, the best algorithms that can be used from ML methods were DT, GBC, and RF. This demonstrates the potential to reduce diagnosis time and costs when integrating machine learning with NGS data. The high performance of the boosting algorithm, which is one of the popular algorithms of the last period, even in the case of data imbalance, is similar to that of our study [43].

Sun et al. proposed the genome deep learning (GDL) method to examine the relationship between genomic variations and traits based on deep neural networks. They analyzed WES mutation data from 6083 samples from 12 cancer types from The Cancer Genome Atlas (TCGA) and WES data from 1991 healthy samples from the 1000

Genomes project. They created 12 different models to distinguish specific cancer types from healthy tissues, a general model that can identify healthy and cancerous tissues, and a mixed model to differentiate all 12 cancer types based on GDL. The accuracy of the different, mixed and total models was found to be 97.47%, 70.08% and 94.70% for cancer diagnosis, respectively. Thus, they reported that an effective method based on genomic information was developed in the diagnosis of cancer. While the accuracy value of the mixed model was determined at the performance level of the models in our study, in the models where high performances were obtained, it was observed that no information about the model parameters was presented. Although very high performance values were obtained in the study, parameter optimization was not mentioned [44].

Maruf FA et al. [7] designed a novel ensemble model using Deep Neural Network (DNN) and XGBoost to classify variants into two classes: somatic and germline, for a given Whole Exome Sequencing (WES) data. The XGBoost algorithm was used to extract features from the results of variant callers, and these features were then fed into the DNN model as input. They noted that the DNN-Boost classification model outperformed the benchmark method in classifying somatic mutations from paired tumor-normal exome data and tumor-only exome data. Although very high performance values were obtained in the study, parameter optimization was not mentioned [7].

Miotto et al. [8] reported that deep learning outperforms machine learning methods in predicting the effects of non-coding mutations in gene expression and DNA similar to our study [8].

The performance of the machine learning models obtained from the studies was found to be similar to the deep and machine learning performance of the real dataset in our study. Additionally, higher performances were achieved in our simulated data compared to the summarized studies. The findings of the study show that when genetic data is evaluated with appropriate models, the outputs are important in terms of time and supporting clinicians.

The recall and precision results of "0 and 1" or "European/non-European ethnicity" predicted in our study were found to be close to each other. This means that both groups achieve a balanced prediction performance in sample class (label) prediction for both deep and machine learning. This result is important in terms of obtaining acceptable models, especially in population-based or rare disease studies. Furthermore, studies in the literature have shown that the deep learning method also shows high performance in imbalanced classification. From this perspective, it can be concluded that the deep learning method's performance in diagnosis-specific models yields reliable results in both detecting patients and distinguishing healthy individuals. The analysis systems (cloud-based & local) in which secondary and tertiary analysis will be performed and the machine features used directly affect the performance of DL models. Especially when modeling big data matrices such as the whole genome, the availability of such infrastructure allows for iterative processes and enables the attainment of maximum performance from the model through hyperparameter optimization.

Conclusion

Scope of this study, the problem of data storage in big data was eliminated by using the cloud system and it became easier to focus on the modeling of the data.

We identified optimized parameters for deep learning and machine learning models.

In this regard, we have provided researchers with a comprehensive guide that utilizes the entirety of genetic information to enable them to obtain fast and highly accurate results.

The data stored and edited in cloud systems are modeled using GPU computing tools within the same environment. In this respect, the study has revealed the advantages of cloud systems throughout the entire research process. Researchers will be able to observe the effect of cloud systems for the highest benefit that can be obtained from genetic data, especially in population-oriented public health studies. A researcher who well-defined his/her hypotheses in the research can achieve both high performance and reliable results in data analysis by saving labor and time by taking into account the parameter optimizations and secondary and tertiary analysis processes specified in our study.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s40537-023-00838-w.

Additional file 1. **Supplementary Table 1**. A slice from the real aligned Chr 22 WGS data set. **Supplementary Table 2**. Performance comparison of Deep Learning algorithm on ART Simulation data set. **Supplementary Table 3**. Performance comparison of Deep Learning algorithm on Chr 22 WGS data set.

Acknowledgements
This project was supported by 100/2000 the Council of Higher Education (YÖK) Ph.D. Scholarships in the field of 100/2000 Bioinformatics/Biostatistics and all the research problem&design belongs to the Su ÖZGÜR's Ph.D. thesis.

Author contributions
SÖ Conceptualization, Methodology, Data Analysis, Writing-original draft preparation. MO Conceptualization, Writing, Review&Editing, All authors read and approved the final manuscript.

Funding
This study was supported by Ege University Office of Scientific Research Projects (BAP) (Project ID: TDK-2020-21725).

Availability of data and materials
Two data sets were used in this study. Sharing ART simulation data with researchers is only possible with contact of the corresponding author. Chr 22 Data has been shared publicly by the Microsoft Research team. It can be accessed via the link below: https://msropendata.com/datasets/0d473c7f-6ddf-4881-aa6d-5ef048e7eaf5

Declarations

Ethics approval and consent to participate
Ethics approval: This study is based on real data (whole genome) of chromosome 22 (containing ethnicity information) which was prepared by the Microsoft Genomics team and available to the public and there-fore ethics issues do not apply.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no conflict of interest.

Received: 18 September 2022 Accepted: 3 October 2023
Published online: 17 October 2023

References

1. Schmidt B, Hildebrandt A. Deep learning in next-generation sequencing. Drug Discov Today. 2021;26(1):173–80. https://doi.org/10.1016/j.drudis.2020.10.002.
2. Blum AL, Langley P. Selection of relevant features and examples in machine learning. Artif Intell. 1997;97(1–2):245–71. https://doi.org/10.1016/s0004-3702(97)00063-5.
3. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44. https://doi.org/10.1038/nature14539.

4. Le NQ, Ho QT, Ou YY. Incorporating deep learning with convolutional neural networks and position specific scoring matrices for identifying electron transport proteins. J Comput Chem. 2017;38(23):2000–6. https://doi.org/10.1002/jcc.24842.
5. Le NQ, Ho QT, Ou YY. Classifying the molecular functions of Rab GTPases in membrane trafficking using deep convolutional neural networks. Anal Biochem. 2018;555:33–41. https://doi.org/10.1016/j.ab.2018.06.011.
6. Le NQK, Huynh TT, Yapp EKY, Yeh HY. Identification of clathrin proteins by incorporating hyperparameter optimization in deep learning and PSSM profiles. Comput Methods Programs Biomed. 2019;177:81–8. https://doi.org/10.1016/j.cmpb.2019.05.016.
7. Maruf FA, Pratama R, Song G. DNN-Boost: somatic mutation identification of tumor-only whole-exome sequencing data using deep neural network and XGBoost. J Bioinform Comput Biol. 2021;19(6):2140017. https://doi.org/10.1142/S0219720021400175.
8. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. Brief Bioinform. 2018;19(6):1236–46. https://doi.org/10.1093/bib/bbx044.
9. Wong KK, Rostomily R, Wong STC. Prognostic gene discovery in glioblastoma patients using deep learning. Cancers. 2019;11:53. https://doi.org/10.3390/cancers11010053.
10. Alharbi WS, Rashid M. A review of deep learning applications in human genomics using next-generation sequencing data. Hum Genomics. 2022;16:26. https://doi.org/10.1186/s40246-022-00396-x.
11. Khan S, Khan M, Iqbal N, Khan SA, Chou K-C. Prediction of piRNAs and their function based on discriminative intelligent model using hybrid features into Chou's PseKNC. Chemometr Intell Lab Syst. 2020;203:104056. https://doi.org/10.1016/j.chemolab.2020.104056.
12. Maraziotis I, Dragomir A, Bezerianos A. Gene networks inference from expression data using a recurrent neuro-fuzzy approach. In: 2005 IEEE engineering in medicine and biology 27th annual conference. IEEE; 2005. p. 4834–7
13. Yamashita R, Nishio M, Do RKG, et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9:611–29. https://doi.org/10.1007/s13244-018-0639-9.
14. Lall A, Tallur S. Deep reinforcement learning-based pairwise DNA sequence alignment method compatible with embedded edge devices. Sci Rep. 2023;13:2773. https://doi.org/10.1038/s41598-023-29277-6.
15. Khan S, Khan MA, Khan M, Iqbal N, AlQahtani SA, Al-Rakhami MS, Khan DM. Optimized feature learning for anti-inflammatory peptide prediction using parallel distributed computing. Appl Sci. 2023;13:7059. https://doi.org/10.3390/app13127059.
16. Li X, Tan G, Zhang C, Li X, Zhang Z, Sun N. Accelerating large-scale genomic analysis with Spark. In: 2016 IEEE international conference on bioinformatics and biomedicine (BIBM), Shenzhen, 2016, p. 747–51. https://doi.org/10.1109/BIBM.2016.7822614
17. Wiesmüller L, Ford JM, Schiestl RH. DNA damage, repair, and diseases. J Biomed Biotechnol. 2002;2(2):45. https://doi.org/10.1155/S1110724302001985.
18. http://www.tibbigenetik.org.tr/upload/2018581083.pdf. Accessed 24 June 2022.
19. Harding KE, Robertson NP. Applications of next-generation whole exome sequencing. J Neurol. 2014;261(6):1244–6. https://doi.org/10.1007/s00415-014-7372-1.
20. Tetreault M, Bareke E, Nadaf J, Alirezaie N, Majewski J. Whole-exome sequencing as a diagnostic tool: current challenges and future opportunities. Expert Rev Mol Diagn. 2015;15(6):749–60. https://doi.org/10.1586/14737159.2015.1039516.
21. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. Bioinformatics. 2012;28(4):593–4. https://doi.org/10.1093/bioinformatics/btr708.
22. 1000 Genomes Project Consortium, Abecasis GR, Altshuler D, et al. A map of human genome variation from population-scale sequencing [published correction appears in Nature. 2011 May 26;473(7348):544. Xue, Yali [added]; Cartwright, Reed A [added]; Altshuler, David L [corrected to Altshuler, David]; Kebbel, Andrew [corrected to Keebler, Jonathan]; Koko-Gonzales, Paula [corrected to Kokko-Gonzales, Paula]; Nickerson, Debbie A [corrected to Nickerson, Debo]. *Nature*. 2010;467(7319):1061–1073. https://doi.org/10.1038/nature09534
23. https://msropendata.com/datasets/0d473c7f-6ddf-4881-aa6d-5ef048e7eaf5 Accessed 17 May 2022.
24. Langmead B, Nellore A. Cloud computing for genomic data analysis and collaboration [published correction appears in Nat Rev Genet. 2018 Feb 12]. Nat Rev Genet. 2018;19(4):208–19. https://doi.org/10.1038/nrg.2017.113.
25. Su ÖZGÜR, Implementation of deep learning technique on next generation sequence data experiments, June 2021, PhD Thesis, ID: 686642. https://tez.yok.gov.tr/UlusalTezMerkezi/tezSorguSonucYeni.jsp. Accessed 10 Sept 2022
26. Cosgun E, Oh M. Exploring the consistency of the quality scores with machine learning for next-generation sequencing experiments. Biomed Res Int. 2020;2020:8531502. https://doi.org/10.1155/2020/8531502.
27. https://support.illumina.com/content/dam/illumina-support/help/BaseSpaceHelp_v2/Content/Vault/Informatics/Sequencing_Analysis/BS/swSEQ_mBS_FASTQFiles.htm. Accessed 12 July 2022.
28. https://www.microsoft.com/en-us/research/publication/exploring-the-consistency-of-the-quality-scores-with-machine-learning-for-next-generation-sequencing-experiments-2018/. Accessed 12 July 2022.
29. Kadalayil L, Rafiq S, Rose-Zerilli MJ, et al. Exome sequence read depth methods for identifying copy number changes. Brief Bioinform. 2015;16(3):380–92. https://doi.org/10.1093/bib/bbu027.
30. Danecek P, Auton A, Abecasis G, et al. The variant call format and VCFtools. Bioinformatics. 2011;27(15):2156–8. https://doi.org/10.1093/bioinformatics/btr330.
31. Stitziel NO, Kiezun A, Sunyaev S. Computational and statistical approaches to analyzing variants identified by exome sequencing. Genome Biol. 2011;12(9):227. https://doi.org/10.1186/gb-2011-12-9-227.
32. Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics. 2009;25:1754–60.
33. Liang W, Luo S, Zhao G, Wu H. Predicting hard rock pillar stability using GBDT, XGBoost, and LightGBM algorithms. Mathematics. 2020;8(5):765. https://doi.org/10.3390/math8050765.
34. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W et al. LightGBM: a highly efficient gradient boosting decision tree. In: 31st Conference on neural information processing systems (NIPS 2017), Long Beach, CA, USA.

35. Ma X, Sha J, Wang D, Yuanbo Yu, Yang Q, Niu X. Study on a prediction of P2P network loan default based on the machine learning LightGBM and XGboost algorithms according to different high dimensional data cleaning. Electron Commer Res Appl. 2018;31:24–39. https://doi.org/10.1016/j.elerap.2018.08.002.
36. Stephens ZD, Lee SY, Faghri F, et al. Big data: astronomical or genomical? PLoS Biol. 2015;13(7):e1002195. https://doi.org/10.1371/journal.pbio.1002195.
37. Köse T, Özgür S, Coşgun E, Keskinoğlu A, Keskinoğlu P. Effect of missing data ımputation on deep learning prediction performance for vesicoureteral reflux and recurrent urinary tract ınfection clinical study. Biomed Res Int. 2020;2020:1895076. https://doi.org/10.1155/2020/1895076.
38. Danilevsky A, Shomron N. Deep learning applied on next generation sequencing data analysis. Methods Mol Biol. 2021;2243:169–82. https://doi.org/10.1007/978-1-0716-1103-6_9.
39. Young JD, Cai C, Lu X. Unsupervised deep learning reveals prognostically relevant subtypes of glioblastoma. BMC Bioinform. 2017;18(Suppl. 11):381. https://doi.org/10.1186/s12859-017-1798-2.
40. https://machinelearningmastery.com/how-to-stop-training-deep-neural-networks-at-the-right-time-using-early-stopping/. Accessed 12 July 2022.
41. Montesinos López OA, Montesinos López A, Crossa J. Fundamentals of artificial neural networks and deep learning. In: Multivariate statistical machine learning methods for genomic prediction. Cham: Springer, 2022. https://doi.org/10.1007/978-3-030-89010-0_10
42. van den Akker J, Mishne G, Zimmer AD, Zhou AY. A machine learning model to determine the accuracy of variant calls in capture-based next generation sequencing. BMC Genomics. 2018;19(1):263. https://doi.org/10.1186/s12864-018-4659-0.
43. Marceddu G, Dallavilla T, Guerri G, Zulian A, Marinelli C, Bertelli M. Analysis of machine learning algorithms as integrative tools for validation of next generation sequencing data. Eur Rev Med Pharmacol Sci. 2019;23(18):8139–47. https://doi.org/10.26355/eurrev_201909_19034.
44. Sun Y, Zhu S, Ma K, et al. Identification of 12 cancer types through genome deep learning. Sci Rep. 2019;9(1):17256. https://doi.org/10.1038/s41598-019-53989-3.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com