Int J Syst Assur Eng Manag
https://doi.org/10.1007/s13198-022-01760-9

ORIGINAL ARTICLE

Multi-level classification technique for diagnosing osteoporosis and osteopenia using sequential deep learning algorithm

T. Ramesh[1] · **V. Santhi**[1]

Received: 25 February 2022 / Revised: 27 May 2022 / Accepted: 2 August 2022

Abstract Sequential analysis techniques have brought new insights into a primary analysis of medical data for early detection of diseases. The presence of Sequential patterns in the datasets help to develop new drugs, define populations susceptible to certain types of illness and also to identify predictors of many medical related common diseases. At the same time, the accuracy of sequential data mining results, may depend upon techniques used, such as convolution of many attributes, feature selection method, handling of class imbalance, algorithm preference, and performance metrics. Sequential data extraction is more successful in detecting osteoporosis and osteopenia, both of which are life-threatening diseases that mostly affect women after menopause. Furthermore, its primary causes include small bone fractures, which may lead to mortality in the latter stages. As a result, the emphasis of this research work is on the development of a classifier using a deep learning technique for early prediction of osteoporosis and osteopenia diseases using health data set. In this paper, we propose, a novel sequential classifier, which is based on the deep convolution neural networks for performing classification on health care dataset relevant to osteoporosis and osteopenia, to increase classification accuracy. The proposed work distinguishes osteoporotic and osteopenia affected patients from a group of people, based on Bone Mass Density values. As per the result obtained, the proposed approach performs classification with improved accuracy and reduces the false positive rate in the detection of osteoporosis and osteopenia.

Keywords Osteoporosis · Convolution neural network · Deep neural networks · Multi-class classification · Osteopenia

1 Introduction

There are many reasons why osteoporosis, which is also called as "metabolic bone disease," can go unnoticed for a long time and happen often. Because of this, bone cells micro-architectural degeneration significantly increases the risk of bone fragility as well as bone fractures, especially in the spine and femur, but also in other parts of the body and thereby patient's height gradually decreases (Jalodia 2021; Klibanski, 2001; Watts et al. 2008). As soon as possible, all of us need to get a precise test and need to undergo therapy that would help things not get worse. If any personnel is suffered with osteoporosis, a bone mineral density (BMD) test known as DEXA is used to measure BMD in order to confirm it. The DEXA testing need to be taken in the lower back, femoral neck, and upper hip area. In the diagnostic testing, prognosis, as well as to treat osteoporosis disease, BMD tests are very important, and it will express BMD in grams of mineral per square centimetre scanned (g/cm2). The DEXA test gives BMD in two standards, they are the T-score and the Z-score, the T- score values of $-1, <-1$, and the values of 1–2.5, indicates that the personnel is affected with osteoporosis, as defined by the World Health Organization (WHO) (Deo 2015). It is used to show how much the patient's score can be above or below the normal value. The Z-score also shows the patient's BMD value in comparing with a personnel having the same age and same

✉ V. Santhi
vsanthi@vit.ac.in

T. Ramesh
t.ramesh2014@vit.ac.in

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, India

Published online: 29 August 2022

sex. This is called the patient's "Z score." The mandibular cortical width that can be measured from digital dental radiographic images can also be used to find people who have low bone mass density (BMD) values (DPRs). The osteoporosis and osteopenia terms are used to describe a loss in bone density and a greater rate of bone breakdown than bone creation, resulting in porous bones. Decisions about a patient's medical care, such as whether or not they need medical treatment, should only be made by a doctor or other licenced health care practitioner. Whenever we have a question about a medical problem, we should always see a doctor or other healthcare professional. More information may be gleaned from physical examination data using sequential structures than with traditional categorization approaches. Different physical examination characteristics may be identified using Sequential procedures, which can also be utilised to learn about each feature's contribution to osteoporosis and osteopenia diagnosis categorization.

Recently, the 4th industrial revolution has seen a flurry of interest in artificial intelligence (AI). AI relies heavily on machine learning (ML), a fundamental component of which is employed in medical research. Prediction and classification are two of the most important functions of a machine learning system. Watson, the most prominent machine learning algorithm, has been used in the pharmaceutical industry to diagnose, analyse images, and optimise treatments (Kavitha et al. 2016). Medical professionals might benefit by using the above said algorithm for prediction and categorization of their diagnostics and treatment of patients. We devised a method for estimating the lumbar spine's T-score and classifying osteoporotic and non-osteoporotic vertebrae using HU of lumbar CT based on the prediction function of MLs. In the Materials and Methods section, we've outlined the steps involved in creating the data collection. In osteoporosis, density is reduced and a change in the microarchitecture structure of the bone appears, which decreases bone tolerance and increases the risk of fracture (Fine 2006). In osteoporosis, there is a progressive loss of bone mass, which is then used to build the model. Disruption of bones microarchitecture results in a decrease in BMD, and changes in bone protein concentration and diversity. Among the most common osteoporotic fractures are those of the hip, spine, and wrist. To be considered osteoporotic, a fracture must occur in a place with low BMD and increase in frequency beyond the age of 50.

Osteoporotic fractures are a leading source of morbidity and death apart from the obvious physical consequences. In the United States, women have a 40% lifetime risk and males have a 13% lifetime chance of suffering a hip, spine, or forearm fracture after the age of 50. In Sweden, women make up 46% of the workforce, while males make up 22%. The BMD of Africans and Americans is 6% higher than that of Caucasians and Asians, putting them at greater risk. Every fifteen seconds, someone in the European Union has an osteoporotic fracture, resulting in a broken bone. As many as 75% of women with osteoporosis don't realise they have the condition.

It is possible for any individual get affected with both primary (idiopathic) and secondary osteoporosis. Primary osteoporosis, which affects women after menopause, is known as postmenopausal osteoporosis. Men might also get senile osteoporosis, which falls under this category. Anyone may develop secondary osteoporosis as a consequence of certain hormonal abnormalities and other chronic illnesses, such as glucocorticoids or other ailments that cause bone loss via multiple processes. When the condition is caused by steroids, it is known as osteoporosis produced by steroids (glucocorticoids). A fractured bone is often the first sign of osteoporosis, which is why the illness is referred to as "the silent crippler" since many individuals don't realise they have it until it's too late. However, early identification and treatment of osteoporosis may reduce a person's risk of fracture to an absolute minimum. Studies employing Artificial Intelligence algorithms have been utilised to determine whether or not a person has osteoporosis for these reasons.

In this work, we provide a novel approach for classifying data into several categories, using Sequential deep convolutional neutral networks based classifier which is used in this work for the effective classification of the osteoporosis and osteopenia medical data sets. For this purpose, we used two data sets namely lumbar spine and femoral neck datasets and performed the training and testing with the Sequential Deep convolution neural network algorithm. The proposed model has been compared with other deep learning classification algorithms namely Long Short-term memory algorithm, Convolution Neural Networks, Recurrent Neural networks, Multi-layer perceptron neural networks and Deep belief neural networks. From the experiments conducted with these existing algorithms and the Sequential deep convolution neural network based approach, it is proved that the Deep Sequential Neural network based classification provides higher classification accuracy than the other existing deep learning algorithms. The major advantages of the proposed model include the increase in classification accuracy, reduction in training and testing time and finally the reduction in false positive rate with respect to the effective identification of osteoporosis and osteopenia diseases.

The rest of the paper is arranged as follows. Section 2 summarises pertinent research and literature. Materials and procedures for this study are described in Sect. 3, including the participants and characteristics that were used, in order to test the proposed approach. The experimental results are discussed in Sect. 4. Finally, Sect. 5 concludes the proposed work.

2 Related works

The study of various deep neural networks and its important features are presented in this section. The Recursive Neural Network (RvNN), Recurrent Neural Network (RNN), CNN, and deep generative models will be covered in this section. This article does not cover all the new networks and architectures that have emerged in recent months due to the rapid growth of deep learning. These deep learning networks are summarised in Table 1, along with the most important aspects and the most important studies.

The Predictions in RvNN may be made in a hierarchical framework, and outputs can be classified by compositional vectors. Recursive Auto Associative Memory (RAAM) (Pijnappels et al. 2005), an architecture designed to handle items arranged in an arbitrary form, such as trees or graphs, was a major inspiration for the construction of a RvNN. Recursive data structures of variable size were used to construct a distributed representation with constant width. Architectures such as RvNN, RNN, and CNN. An approach known as structure (BTS) learning was used to teach the network new skills (MacRae et al. 1992). BTS is a backpropagation technique (Goller and Kuchler 1996) that can accommodate a tree-like structure in addition to the usual backpropagation approach. Auto Association is used to teach the network to replicate the input layer's pattern in the output layer. An area where RvNN has shown particular success is in natural language processing (NLP) (Runge et al. 1999). In 2011, Runge et al. (1999) suggested a RvNN architecture that can handle input from several modalities. Using RvNN to categorise natural photos and natural language words is shown in two instances in Runge et al. (1999). A phrase, on the other hand, is broken down into its individual words. To create a syntactic tree, RvNN determines the potential pair's score. RvNN calculates a merging plausibility score for each pair of units. In order to create a compositional vector, we combine the two pairs that have the greatest scores (Crepaldi et al. 2007). After each merging multiple units, a composition vector, and a class label are used in RvNN to create a bigger area. Class label for the new region, assuming both components are nouns, would be a noun phrase. The compositional vector representation of the whole area is used as the roots of the RvNN tree.

As per the survey, ReLu and Sigmoid function has given better results compared to linear and bipolar Methods. The RNN is sensitive to the disappearing and discharge gradients, hence the gradients may degrade or increase exponentially as a result of the training's repeated multiplication of tiny and large derivatives. When the network ages, its sensitivity decreases, causing it to lose track of earlier inputs as new ones are introduced. The LSTM (Papaioannou et al. 2010) provides memory blocks in its recurrent connections as a way of dealing with this problem. The network's temporal states are stored in memory cells in each memory block. In addition, gated units are included to regulate the flow of information. The vanishing gradient problem may be alleviated greatly by residual connections in extremely deep networks (Bonaiuti et al. 2005).

In deep learning, CNN is a well-known and commonly used algorithm (Bonaiuti et al. 2005). For example, it has been used widely in NLP (Li et al. 2013), voice processing (Li et al. 2013), and computer vision, to mention a few applications. Neurons in animal and human brains serve as inspiration for the construction of this network. When it comes to cats' visual cortex, it's specifically designed to mimic the intricate sequence of cells (Li et al. 2013),

Table 1 Deep neural networks with its features in various applications

DL networks	Descriptive key points	Papers
RvNN	Utilizes a structure like a tree Preferable in NLP	Goller et al. (2010),Zhou et al. (2010), Socher et al. (2016) Hajimirsadeghi and Mori (2016)
RNN	Excellent for sequential data Preferable for natural language processing and voice processing	Cho et al. (2020), Ezat et al. (2020), Li et al. (2013), Everingham (2009)
CNN	Initially intended for image recognition Extensive support for natural language processing, voice processing, and computer vision	LeCunn et al. (2014), Jia et al. (2014), Krizhevsky et al. (2014), Jia et al. (2014), Kim (2014), Jia et al. (2014), Abdel-Hamid et al. (2017), Bronshtein (2017), He et al (2017)
DBN	Directed connections Unsupervised learning	Hinton (2013), Tseng et al. (2013), Hansen (1990), Hinton et al. (2013), Tseng et al. (2013), Hansen (1990), Hadsell et al (2008)
DBM	Undirected connections Unsupervised learning Composite model of RBMs	Salakhutdinov et al. (2014), Zhou et al. (2002), Salakhutdinov et al. (2013), Pulido et al. (2013), Pulido et al. (2014), Horlings et al. (2009)
GAN	Game-theoretical framework Unsupervised learning	Goodfellow et al. (2016), Sarle (1996), Basheer and Hajmeer (2000), Radford et al. (1996), Baudry et al. (2012)
VAE	Graphical model Unsupervised learning Probabilistic	Kingma et al. (2005), He et al (2016) Pijnappels et al. (2005)

(Schoor et al. 2006). Three key benefits of CNN are discussed in Itoi and Sinaki (1994), including parameter sharing, sparse interaction, and comparable representation. Local linkages and shared weights are employed in the network rather than completely connected networks to make use of the two-dimensional design of the incoming data (e.g., an image signal) (Mackey and Robinovitch 2006). As a consequence of this procedure, the network may be trained more quickly and with fewer parameters. The visual cortex cells do a similar action. A little portion of a scene is all that these cells are sensitive to. As a result, the cells function as local filters for the input, allowing them to extract any spatially local correlation that could exist.

When it comes to CNNs, convolutional layers are separated by pooling (subsampling) and then fully connected (MLP-like) layers. It is common for CNNs to use three dimensions to represent the inputs (x), which are represented by the three dimensions of the inputs ($x = \mathrm{m} \times \mathrm{m}\ \mathrm{r}$): height, width, and depth ($\mathrm{r} = 3$ for RGB images). Each convolutional layer has k filters (kernels) of size n n q in each of the convolutional layers. It's important to keep in mind that although n should be less than the input picture, q might be either smaller or equal to r. When the input is divided by the number of feature maps (hk) and each map has an average size of 1, the filters are used as a foundation for the local connections, which are convolved with the input that share the same parameter values as before (weight W k and bias bk). Weights and the layers' inputs (as described in Eq. 1) are combined to form a dot product, similar to how MLP accomplishes it, except that the inputs are just a fraction of the original volume. Convolutional outputs are subsequently subjected to output nonlinearity, represented by the vector f.

$$h_k = f(W_k * x + b_k \tag{1}$$

To decrease the number of parameters in the network, speed up training, and prevent overfitting, each feature map is then subsampled in the subsampling layers. Over a continuous area in all feature maps, the pooling operation is performed (e.g., average or maximum), like in typical neural networks, the final stage layers are often fully connected. The data is abstracted at a high level using these layers, which integrate low- and mid-level components (Yang and Pai 2007). An event's likelihood of falling into a certain category is encoded in each score generated by the final layer (such as Softmax or SVM). Deep Boltzmann Machine (DBM), Variational Autoencoder (VAE), and the DBN are all examples of deep generative networks (VAE). Undirected connections are used in the top two layers of DBN (Sinaki 2003), whereas directed connections are used in the lower levels, creating a probabilistic generative hybrid model. In the lowest visible layer, the input units' values are represented as a data vector. In an unsupervised method, a DBN learns to probabilistically recreate its inputs, with the layers serving as feature extractors. For the DBN to execute categorization tasks, a supervised training approach is required. The DBN seems to be a collection of RBMs (Huntoon et al. 2008), (Burke et al. 2012).

The various deep learning algorithms aid in the improvement of learning performance, the expansion of application scope, and the simplification of the calculation process. The incredibly lengthy training period of deep learning models, however, continues to be a serious issue for researchers. Furthermore, increasing the number of training data & model parameters may dramatically improve classification accuracy. Several innovative strategies have been developed in the literature to speed up deep learning processing. Integration of modularized deep learning algorithms, optimization approaches, distribution mechanisms, and infrastructure support are all part of deep learning frameworks (Bergland et al. 2011). They've been designed to make the implementation process easier while also boosting system-level development and research. Some of these exemplary methodologies and frameworks are presented in this section.

Unsupervised deep learning algorithms based on generative models, such as GANs and VAEs, have been more popular in recent years. To train and apply a fixed extracted features for supervised tasks, GANs are used in Huntoon et al. (2008). An unsupervised learning network based on CNNs has shown to be better in visual data processing. Another study uses a deep sparse Auto encoder to learn features from a very large-scale picture dataset (Huntoon et al. 2008) (Sinaki et al. 2002). From unlabeled input, this network generates a high-level feature extractor that may be used for unsupervised face identification. Additionally, other high-level objects including such animal faces and human bodies may be identified using the derived features. As an alternative to maximum likelihood, Crisco and Panjabi (1991) suggest a generative probabilistic technique for unsupervised learning based on Markov chain Monte Carlo transitional operators. Most individuals don't have the resources necessary to train a deep neural network from the ground up in a reasonable amount of time. As a result, big datasets are often used to pre-train a deep network (such as a CNN) (e.g., ImageNet). Using pre-trained networks as fixed feature extractors or fine-tuning the weights of the pre-trained model is a method known as "transfer learning" (Geiger et al. 2013). (Especially for large new datasets that are similar to the original one). Alternatively, the model should keep learning to optimise the weights of the deep network's high-level components in order to prevent this from happening again in the future. Because the dataset is insufficient to train a full deep network, this approach is referred to as semi-supervised learning.

In deep learnig, network topologies and architectures are often time static (defined before learning starts) or time

invariant (that is, they do not change over time). When data is broadcast online, the temporal complexity constraint presents a substantial barrier to the system (Bennell et al. 2010). However, since online deep learning was initially incorporated into mainstream research, only a tiny increase has been seen (Girshick 2015). DNNs are generally built using the Stochastic Gradient Descent (SGD) approach, in which training data are used one at a time to update model parameters using a previously defined label. Rather of doing sequential processing on each sample, the modifications must be made in batches (Leetun et al. 2004). Girshick (2015) offers a unique strategy in which the samples in each batch are considered as Independent and Identically Distributed (IID) (Geusens and Dinant 2007). Batch processing is more efficient because it equally divides computer resources & execution time. Another issue that occurs when dealing with online learning is dealing with high-velocity data with time-varying distributions. This task is inspired by banking and financial data pipeline which are very valuable in terms of monetary worth. The present assumption is that the data is sufficiently near in time to be able to securely assume piecewise stationarity, resulting in a distribution that is comparable. This assumption, as stated in Girshick et al. (2014), characterizes material as having a specific degree of correlation thus leads to the construction of models based on this assumption. Nonstationary datasets, on the other hand, do not contain IIDs and are usually longitudinal data streams. Furthermore, online learning is often memory-limited, difficult to parallelize, and requires a linear learning rate on each input data set. The introduction of algorithms capable of learning online from non-IID data would be a big step forward towards deep learning applications that use enormous volumes of data.

Training a DNN (Glorot and Bengio 2010) is an optimization process in which the objective is to determine the network parameters that minimise the loss function. A basic technique used to deep learning, the SGD method (Goodfellow et al. 2016) is an iterative process that modifies parameters depending on the gradient for each training sample, which is implemented in practise as a heuristic. It is less computationally expensive than the original gradient descent method, which evaluates the whole dataset whenever the parameters are altered. The hyper parameter learning rate is responsible for controlling the update pace throughout the learning process. After a lengthy period of time, lower learning rates will ultimately lead to an ideal state, while higher learning rates would decay the loss more quickly but may produce swings throughout the training period (Generative adversarial nets 2014). The concept of utilising momentum to control the oscillation of SGD is presented in order to regulate the oscillation of SGD. Because it is based on Newton's first law of motion, this approach provides quicker convergence as well as a correct momentum, which may enhance the optimization outcomes obtained by the SGD (Google 2016). For determining the appropriate learning rate, on the other hand, numerous strategies have been presented. The addition of weight decay & learning rate decay is mostly for the purpose of regulating the learning rate & expediting its convergence. A learning rate decay may be used to dynamically lower the learning rate of a learning algorithm, enhancing its performance (Ha et al. 2015). A weight decay works as a penalty coefficient in the cost function, avoiding overfitting. Furthermore, it has been found that adjusting the learning rate with relation to the gradients of the previous stages helps minimise volatility. Adagrad (Graves et al. 2013) is the first adaptive algorithm that has been successfully used to applications using deep learning. By accumulating the cumulative squared gradients over time, it boosts the learning rate for rarely modified parameters while lowers the learning rate for regularly updated parameters. Adagrad's learning rate may get extremely low, at which time the model would no longer be optimised, given the fact that square gradients are always positive. The osteoporosis and osteopenia is identified using BMD and if BMD value is high there is a high risk of osteoporosis. This has been analysed using the clinical femoral data set and performed classification with high accuracy in our proposed research work.

A model parallelism technique, on the other hand, divides the training stage over numerous graphics processing units. In a basic model-parallel technique, each GPU is responsible for just a part of the model's computation. Example: If the model includes two LSTM layers, the systems with two GPUs might use one of them to calculate one of the LSTM layers, resulting in a faster total calculation time. The model-parallel method has the advantage of allowing for the training and prediction of very large deep neural networks (Graves et al. 2013). The COTS HPC system, for example, used roughly 82 GB of memory to train a neural network with over 11 billion parameters (Graves et al. 2013). Model-parallel approaches are essential since fitting such a large model onto a single processor is difficult, hence the model must be partitioned. One downside of model parallelism is each node can only compute a fraction of the results (Gregor and LeCun 2010), requiring the usage of synchronisation to get the whole set of results (Gregor and LeCun 2010). Because each node in a model-parallel approach must synchronise gradients and attribute values on every update step, it imposes greater synchronisation loss & communication cost than a data-parallel strategy (Sinaki and Mikkelsen 1984). This indicates that model parallelism's scalability is less than ideal. Greenspan et al. (2016) In order to find the best model partitioning and placement strategy, Google has provided an automatic device positioning system based on deep reinforcement learning. If you combine processes

together based on their embedding representation, you may get a 60 percent boost in performance over using human specialists alone, according to this methodology.

Although there are several studies in the literature that are connected to the identification of osteoporosis and osteopenia, many gaps have been detected in the research as a result of their existence. The majority of the studies that were identified in the literature employed basic data sets to validate the model presented by the researchers that worked in the field of osteoporosis and osteopenia detection, and this was the case for the majority of the studies (Garofolo et al. 1993). Furthermore, when the data was not appropriately pre-processed and when the entire set of characteristics was utilised for classification, the machine learning-based algorithms employed for the identification of osteoporosis and osteopenia illnesses took longer to complete the task. Currently available work on osteoporosis and osteopenia detection using machine learning approaches has a number of disadvantages, one of which is that it is divided into two parts, namely the feature selection phase and the classification phase. The suggested deep learning-based technique, on the other hand, not only conducts automated feature extraction and classification (Han et al. 2014), but it also demonstrated improved classification accuracy when tested with both T-Scores and Z-scores, as shown in the following experiments. Additionally, the suggested technique has the advantage of reducing categorization time as well as mistake rate, which is a significant benefit. As per the investigation carried out, it is understood that the Sequential deep convolutional neural networks is efficient for the classification of osteoporosis and osteopenia. In deep convolutional neural networks, the network parameters are identified and optimized to minimise the loss function. In Sect. 3, the materials and methods are discussed in detail.

3 Materials and methods

The proposed research focuses primarily on the femoral and spine dataset initially describing the research protocol, Algorithms contribution towards research and state which statistical tests were done to analyse the data for Osteoporosis and osteopenia.

3.1 Datasets

In this proposal, we have used publically accessible datasets for testing and evaluating the same. In total, we have used two data sets, they are Femoral and Spine. In each dataset, there are 5401 entries with 10 features in each record. There are four features describes about demographic information of patients and also the remaining six include mandibular cortical and trabecular bone properties obtained from of the patients' digital data collected (Tables 2, 3).

Table 2 Medical data set description

Attribute /feature number	Attribute/feature name	Attribute/feature description
1	Age	Patient age
2	Height	Patient height
3	Weight	Patient weight
4	BMI	Body mass index
5	C.Width	Cortical bone thickness or width
6	C.FD	Fractal dimension of cortical bone
7	Tr.thick	Thickness of trabecular bone
8	Tr.FD	Fractal dimension of trabecular bone
9	Tr.Numb	Trabecular count
10	Tr.separa	Trabecular disconnection

As illustrated in Fig. 1, the model evaluation process begins with the consideration of the osteoporosis as well as osteopenia data sets, which are further divided into training and testing sets, as we have chosen a multi-layered neural network that represents forward and backward pass with an input layer, a hidden layer, and an output layer. Initial consideration is given to random weights & constant bias. The output of the hidden layer is activated using the Sigmoid and Relu functions. The output of the forward pass is used to estimate the error.

If the error difference is high, then it is back propagated using derivatives and updated the weights. Finally, classification is properly monitored and performance evaluation is performed.

3.2 Deep neural network

The neural network must constantly learn in order to do tasks more competently or even to use many approaches to get a better outcome. When it acquires new information, it learns how to respond appropriately in a new scenario. When the problems you solve get more difficult, the learning becomes more in-depth. A deep neural network is a kind of machine learning wherein the system utilises several levels of nodes to derive high-level functions from input data. This requires the facts to be transformed into a more abstract & creative component. Consider the following photograph of an ordinary guy to better comprehend the output of deep learning. Even if you've never seen this image of his face or body before, you'll always recognise it as a person and distinguish it from other species. This is an instance of how a deep neural network operates. Information is analysed and categorised creatively and analytically to guarantee that the

Table 3 Features of Osteoporosis and osteopenia

Clinical reference	Osteoporosis	Non-osteoporosis	*p* value
T Score	(T-Score ≤ −2.5)	(T-Score > −2.5)	–
Number of patients	598	535	–
Female	506 (84.6)	371 (69.3)	< 0.0001
Male	92 (15.4	164 (30.7)	< 0.0001
Mean age (SD, min–max)	82.7 (8.3, 60, 100)	77.7 (9.0, 60–98)	< 0.0001
BMI (SD, min–max)	20.1 (3.1, 13.3–29.0)	23.3 (9.0, 14.1–39.2)	< 0.0001

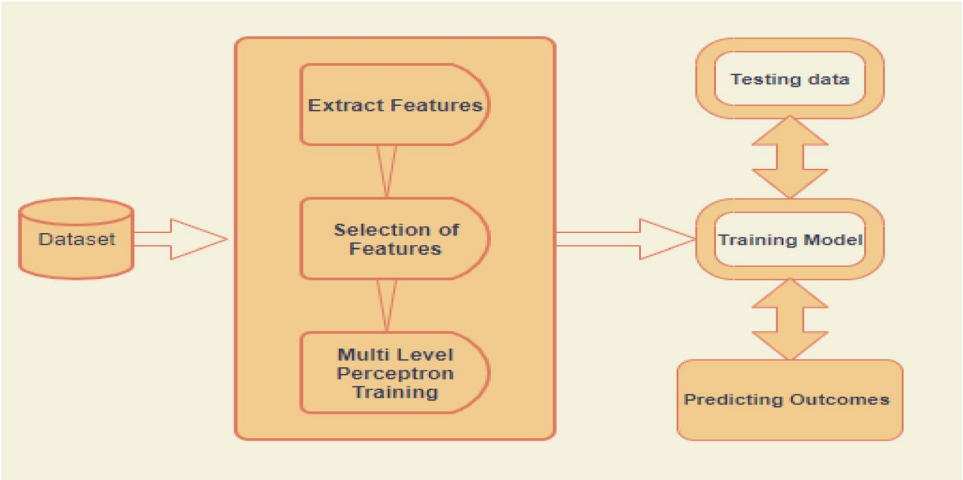


Fig. 1 Flow diagram of the typical processing of machine learning algorithm

item is appropriately recognised. Because these elements are not introduced directly into the system, the machine learning system must alter and generate them.

There are many varieties of neural networks, and the distinctions between them are in their operating principles, action scheme, and application domains. Convolutional neural networks (CNN) are primarily utilized for image identification, with audio recognition being a rare exception. It is often used with photographs since it eliminates the need to verify each pixel individually. CNN examines a picture block by block, beginning in the left upper corner & progressing pixel by pixel until a successful conclusion is reached. The outcome of each verification is then transmitted via a convolutional layer, where some data components have connections and some do not. The system can generate the results of the verifications and deduce what is in the image based on this data.

A Deep Neural Network's simplified description is a hierarchical (layered) design of neurons connected to other neuron (similar to the neurons in the brain). In result of the information they receive, these neurons send out a message or signal to other neurons, producing a complex system that learns via some sort of feedback. A 'N' layered Deep Neural Network is shown in the figure below (Fig. 2).

3.3 Sequential model

A Sequential model is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor. A Sequential model is not appropriate, if the model has multiple inputs or multiple outputs and any of the layers has multiple inputs or multiple outputs. The sequential algorithms need the text input to be transformed into a fixed-length vector and numerous processes are required for training the network model, which may be described algebraically using the matrices of input attribute values and weights. Consider an n-by-p design matrix, where n signifies the total number of observed samples and p denotes the total number of parameters evaluated across all observed samples. The model then generates a forecast, wherein the weights are changed to identify trends to improve forecast accuracy.

Consider the input space $X = R\ X$ and the output space $Y = R\ Y$, with X and Y denoting the input and output dimensions, respectively. We denote $\{(\times 1, y1), \ldots, (x', y')\}$ the set of labelled training instances such that $xi \in X$ and $yi \in Y$. $\{(x' + 1, y' + 1), \ldots, (xT, yT)\}$ will denote the set of testing examples.

The DAG structure is included in the DSNN model:

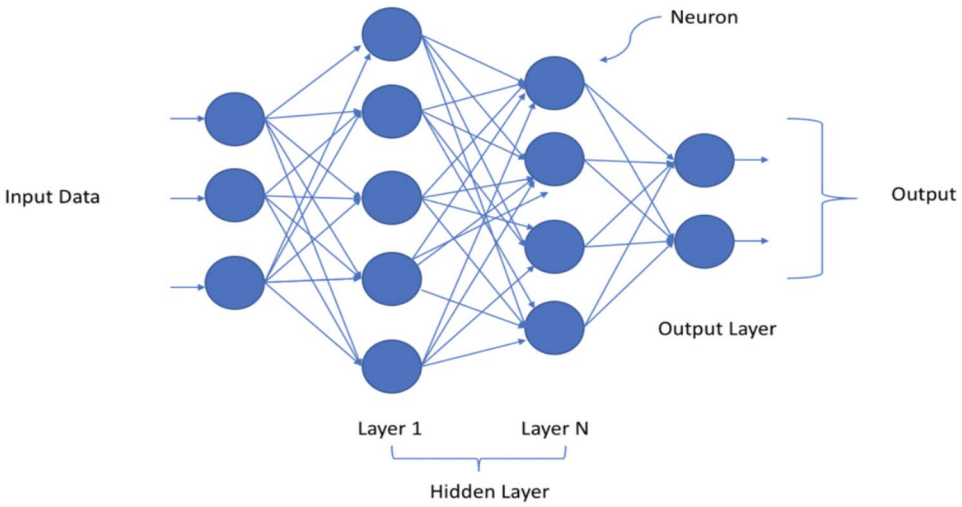


Fig. 2 The basic structure of deep neural network

- Each node n belongs to the list n1,.. nN, where N is the total number of nodes in the DAG.
- The node n1is the root of a tree as it has no parents.
- Because cn, i denotes the i-th child of the provided node n and #n denotes the node's total number of children, I must be an integer between 1 and #n.

  We describe a model in which each node has three children. Using pn1 and pn2, the sequence of chosen nodes is calculated for a given input (n1, n2, n6, n8). Not that the figure represents just pn1 and pn2, but that each node has a p function associated with it. The final prediction, given x, is thus fn6, n8 (fn2, n6 (fn1, n2 (x))).

  leaf(n) is true if and only if node n is a leaf of the DAG, – i.e., a node with no children.
- Each node is associated with a unique representation space. R dim(n), where dim(n) denotes this space's dimension. Nodes perform the same job as layers in traditional neural networks.—dim(n1) = X, i.e. the dimension of the root node equals the dimension of the model's input. – Dim(n) = Y for each node n if leaf(n) is true, that is, the measurement of the leaf nodes equals the dimensions of the output space.
- We examine the mapping functions fn,n0: R dim(n) → R dim(n 0), which are associated with edges (n, n0). fn,n0 returns a new representation in node n0 of the input x depending on the representation in node n. Similar to a neural network, the model's output is a sequence of f-transformations performed to the input.
- Additionally, each node is associated with a selection function denoted by the symbol pn: R dim(n) → R #n is capable of generating a score for each of node n's children given an input in R dim (n). This function returns a probability distribution over the child nodes of n given a vector.

$$P\left(C_{n,i}|Z\right) = \frac{e^{P_n^i(z)}}{\sum_{j=1}^{\#n} e^{P_n^j(z)}} \tag{2}$$

Selection functions are used to determine which f-functions to utilise by traversing the DAG from root to leaf nodes (Google 2016)

$$a^{(t)} = b + W_h^{(t-1)} + U_x^{(t)} \tag{3}$$

$$h^{(t)} = \tanh\left(a^{(t)}\right) \tag{4}$$

$$o^{(t)} = c + V_h^{(t)} \tag{5}$$

$$y^{\wedge(t)} = soft\max\left(o^{(t)}\right) \tag{6}$$

In Sequential networks, like this one, translate input sequences to outputs of equal length. Thus, it would simply add the losses over all time steps to get the total loss for a particular sequence of x and y values. Softmax function is used to produce the vector of probabilities over output based on the outputs o(t). L is the negative logarithm of the probability of achieving the actual objective y(t) with the input so far, and we suppose that this reflects the loss. (Google 2016). Multimodal deep learning is another popular direction in recent deep learning research because of redundancy in the basic model approach.

Following a forward propagation run across the graph illustrated above, a reverse propagation pass is performed going right to left. This is the computation of the gradient. Parallelization is not a possibility due to the inherent sequential nature of the forward propagation graph; each time step may be computed only after the previous one. Memory consumption is

thus O(), as forward-pass calculations must be kept until they can be utilised in the backward pass. On an unrolled graph with an O() cost, a technique known as back-propagation over time may be employed (BPTT). All time steps in the network share the parameters; therefore each output's gradient relies on both the current time step's computations and the prior time step's calculations. The reduction of time in performance measurement is one of the criteria where in the proposed reduction of working time is made by the time taken by the algorithm to the class. Our three weight matrices U, V, and W have gradients, or bias terms b, c., as well as the learning rate, must be calculated based on our loss function L. The gradient offers us a feel of how the loss changes in relation to each weight parameter, just like standard back-propagation. In order to minimise loss, the following equation is used to update the weights W:

$$\mathrm{W} \leftarrow \mathrm{W} - \alpha \frac{\partial L}{\partial W} \tag{7}$$

The same procedure applies to the other values U, V, b, and c.

The gradient is calculated with reference to the output o(t), assuming that o(t) is sent as an argument to the softmax function, which generates a vectors of probabilities over the output. Furthermore, we assume that the loss is proportional to the negative logarithm of the actual target y. (t) (Graves et al. 2013)

$$\left(\nabla_{o^{(t)}} L\right)_i = \frac{\partial L}{\partial o_i^{(t)}} = \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial o_i^{(t)}} = y_i^{\wedge(t)} \; 1_{i=y^{(t)}}. \tag{8}$$

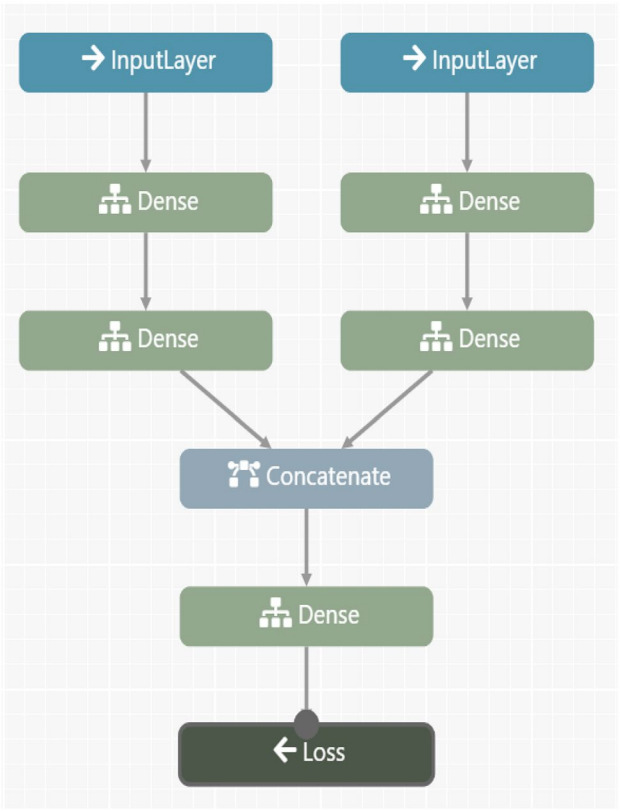


Fig. 4 Sequential model layers

The sequential model design is described in the above Fig. 3 which is going to show the procedure from data set selection, Feature Extraction, Pre-processing, classification

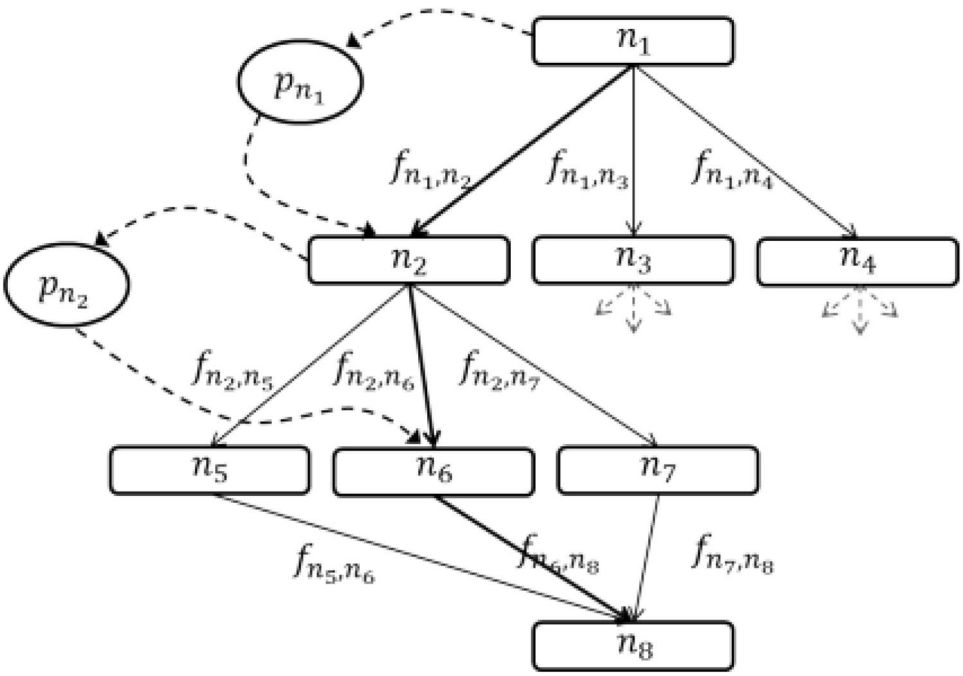


Fig. 3 Architecture of deep sequential neural networks

based on training and testing data followed by our Proposed model implementation Sequential model and finally Predicting outcome. Detailed Sequential model implementation with Layers is illustrated in Fig. 4

The sequential algorithms workflow is described as per the femoral dataset is mentioned below,

Step 1: Specify the Input shape
The model must be aware of the form of the input data it will be receiving. The very first layer in a linear model (only the first layer, since subsequent layers are capable of doing automated shape inference) must thus receive information on the shape of the input shape. There are a variety of options for doing this: The first layer should be given an input shape parameter. There are two types of items in this form tuple: integers and None entries, with None indicating that any positive integer may indeed be anticipated. The batching dimension is not included in form of the supplied data.

(a) The first layer should be given an input shape parameter. There are two types of items in this form tuple: integers and None entries, with None indicating that any positive integer may be anticipated. The batching dimension isn't included in input shape function.
(b) Some 2D layers, such as Dense, allow the specification of their input form via the use of the input dim argument, while some 3D temporal layers allow the specification of their input length through the use of the input dim and input length parameters.
(c) You may supply a batch size parameter to a layer if you need to set a fixed batch size for your input data (this is important for stateful recurrent networks, for example). Layers will demand every set of inputs to get the batch shape if you feed them both batch size = 32 and input shape = (6, 8) to them (32, 6, 8).

Step 2: Compilation

(iv) Before you can begin training a model, you must first setup the learning process, which is accomplished via the compile procedure. It is presented with three arguments:
(v) A programme that optimises. This might be a string identification for an existing optimizer (such as rmsprop and adagrad), or an implementation of the Optimizer class, depending on the context. Optimizers are a good example of this.
(vi) A loss function is used. This is the goal that the model should attempt to achieve as efficiently as possible. Either a loss function (including such categorical crossentropy or mse) or an objective function may be specified using the string identifier. Losses are one example of this.
(vii) A list of metrics is provided. You will also want to put this to metrics = ['accuracy'] for every classification challenge you are working on. This might be the string identification of a current metric or the function callback for an undocumented measure.
(viii) We will separately go over optimizers, loss functions, and metrics in a later lesson.

Step 3: Training
On Numpy arrays of data input and labels, Keras models are trained. Three functions are available for training a model:

(a) The fit function, this is the most basic
(b) The fit_generator. This is a bit more complicated as it takes in a generator instead of a numpy array. Often used for large datasets.
(c) The train_on_batch function which allows you to do a single gradient update over one batch of samples.

Step 4: Evaluation
All evaluation methods have the same extra methods: X, X_on_batch, and X_generator, As a result, I shall leave to the reader's exploration of them. I will show off all the Xs below:

- Evaluate/test (test_on_batch)
- Predict/predict_classes/predict_proba (only predict for generator etc.

As we mentioned regarding implementation procedure in the materials and methods section the analysis, applying the model and extracting the result will be discussed in the Result and Discussion section.

4 Results and discussions

The data collection was read and converted to comma-separated value files CSV on the local workstation using Python scripting language. The resulting CSV files include the picture numbers, image classes, and image labels. The technique includes the following steps:

Configuration of the environment: import libraries & read data Analyse Data: Gain an understanding of the variables' meaning and predictive ability. The processed to extract features from raw data is called feature engineering.

Pre-processing steps include data segmentation, handling missing values, encoding categorical variables, and scaling. Selecting Features: retaining just the most essential characteristics Model development: training, fine-tuning hyper parameters, validation, and testing Evaluation of performance: read the measurements and describe the capability:—comprehend how the model generates results. A very large quantities of data to get better results than other techniques. It is extremely expensive to train due to complex data modeling.

Figure 5 indicates the data set of the Osteoporosis and osteopenia dataset. Each attribute is responsible for multi-level classification. Figure 6 illustrates the statistics of the training data, where each class are demonstrated in the form of count, standard deviation, minimum value, mean, & maximum value for the aggregation of training data.In machine learning, data normalisation is used to make learning algorithm less sensitive to the size of features. This enables our system to converge to more correct weights, resulting in a more accurate estimate. Normalization improves the consistency of the characteristics, which enables the model for predicting outputs more correctly. Statistics of the training data is shown in Fig. 6 and Normalized data is shown in Fig. 7.

Because we cannot input integer sequences to the neural network, we will vectorize and transform each sequence to tensors. This is accomplished using One-Hot Encoding each sequence. Our training / testing data include unique components. Vectorizing our input data produces two two-dimensional tensors; a training data tensor of shape as well as a test input tensor of shape. Figure 8 illustrates the seaborn axis grid.

4.1 Performance metrics

The Deep neural sequential models used in the modality-specific transfer learning with fine-tuning and ensemble learning were evaluated in terms of the following performance metrics:(6) accuracy, (7) precision (positive predictive value; ppv), (8) recall (sensitivity), (9) specificity, (10) negative predictive value (npv), (11) F1 score.

	Weight	Standing Height	Body Mass Index	Gender	Race/Ethnicity - Recode	Age in years	Femoral neck BMD	Femoral neck BMC	Femoral neck area	T-score	class
1682	118.9	180.6	36.45	1.0	3.0	73.0	0.932	5.67	6.08	-0.1	0.0
2814	76.1	158.6	30.25	2.0	1.0	54.0	0.872	4.25	4.88	-0.1	0.0
4055	78.2	172.2	26.37	2.0	3.0	63.0	0.691	3.58	5.17	-1.7	1.0
42	77.3	154.1	32.55	1.0	1.0	51.0	0.554	2.67	4.82	-3.2	2.0
3118	49.3	147.2	22.75	2.0	1.0	71.0	0.672	3.05	4.54	-1.9	1.0
...	...	...	...	...	...	...	...	...	...	...	...
4106	80.7	160.6	31.29	2.0	3.0	64.0	0.752	3.52	4.67	-1.2	1.0
2706	77.6	170.3	26.76	1.0	5.0	76.0	0.795	4.23	5.32	-1.2	1.0
2007	77.4	166.0	28.10	1.0	4.0	51.0	0.781	4.03	5.17	-1.3	1.0
3358	72.7	159.3	28.60	2.0	2.0	67.0	0.652	3.03	4.65	-2.1	1.0
1824	122.1	182.8	36.54	1.0	3.0	77.0	0.665	4.07	6.11	-2.3	1.0

5392 rows × 11 columns

Fig. 5 Dataset

	count	mean	std	min	25%	50%	75%	max
Weight (kg)	4316.0	78.485264	17.459987	32.30000	66.30000	77.000000	89.000000	160.500000
Standing Height (cm)	4317.0	166.464587	10.002593	137.30000	159.20000	166.100000	173.600000	199.300000
Body Mass Index (kg/m2)**	4314.0	28.238053	5.401936	14.59000	24.50000	27.740000	31.347500	52.700000
Gender	4319.0	1.501968	0.500054	1.00000	1.00000	2.000000	2.000000	2.000000
Race/Ethnicity - Recode	4319.0	2.911785	1.112624	1.00000	2.00000	3.000000	4.000000	5.000000
Age in years	4319.0	62.616578	8.699276	50.00000	55.00000	62.000000	69.000000	85.000000
Femoral neck BMD	4319.0	0.776358	0.141770	0.34100	0.67800	0.768000	0.861000	1.406000
Femoral neck BMC	4319.0	4.077543	0.912625	1.70000	3.42000	4.020000	4.640000	10.080000
Femoral neck area	4319.0	5.239843	0.565106	3.08000	4.85000	5.230000	5.640000	8.030000
T-score	4319.0	-1.182642	1.160184	-4.80531	-1.99115	-1.258065	-0.489331	4.415929

Fig. 6 Statistics of the training data

	Weight	Standing Height	Body Mass Index	Gender	Race/Ethnicity - Recode	Age in years	Femoral neck BMD	Femoral neck BMC	Femoral neck area	T-score
182	106.2	178.3	33.41	1.0	1.0	57.0	1.366	6.54	4.78	3.4
3438	77.7	158.3	31.01	2.0	3.0	50.0	0.716	3.55	4.95	-1.5
4076	45.5	153.4	19.30	2.0	3.0	63.0	0.560	2.67	4.76	-2.9
1081	83.5	175.0	27.27	1.0	3.0	57.0	0.868	4.81	5.54	-0.6
2201	75.3	178.9	23.50	1.0	4.0	59.0	0.559	3.24	5.80	-3.1
...	...	...	...	...	...	...	...	...	...	...
4552	72.5	162.6	27.40	2.0	3.0	76.0	0.548	2.53	4.62	-3.0
2868	75.3	154.7	31.46	2.0	1.0	58.0	0.676	3.61	5.34	-1.8
4461	80.2	161.6	30.71	2.0	3.0	73.0	0.906	4.72	5.20	0.2
2636	112.4	176.8	35.96	1.0	5.0	62.0	1.119	5.89	5.26	1.4
3356	78.1	160.2	30.40	2.0	2.0	67.0	0.738	4.02	5.45	-1.3

1079 rows × 10 columns

Fig. 7 Normalized dataset

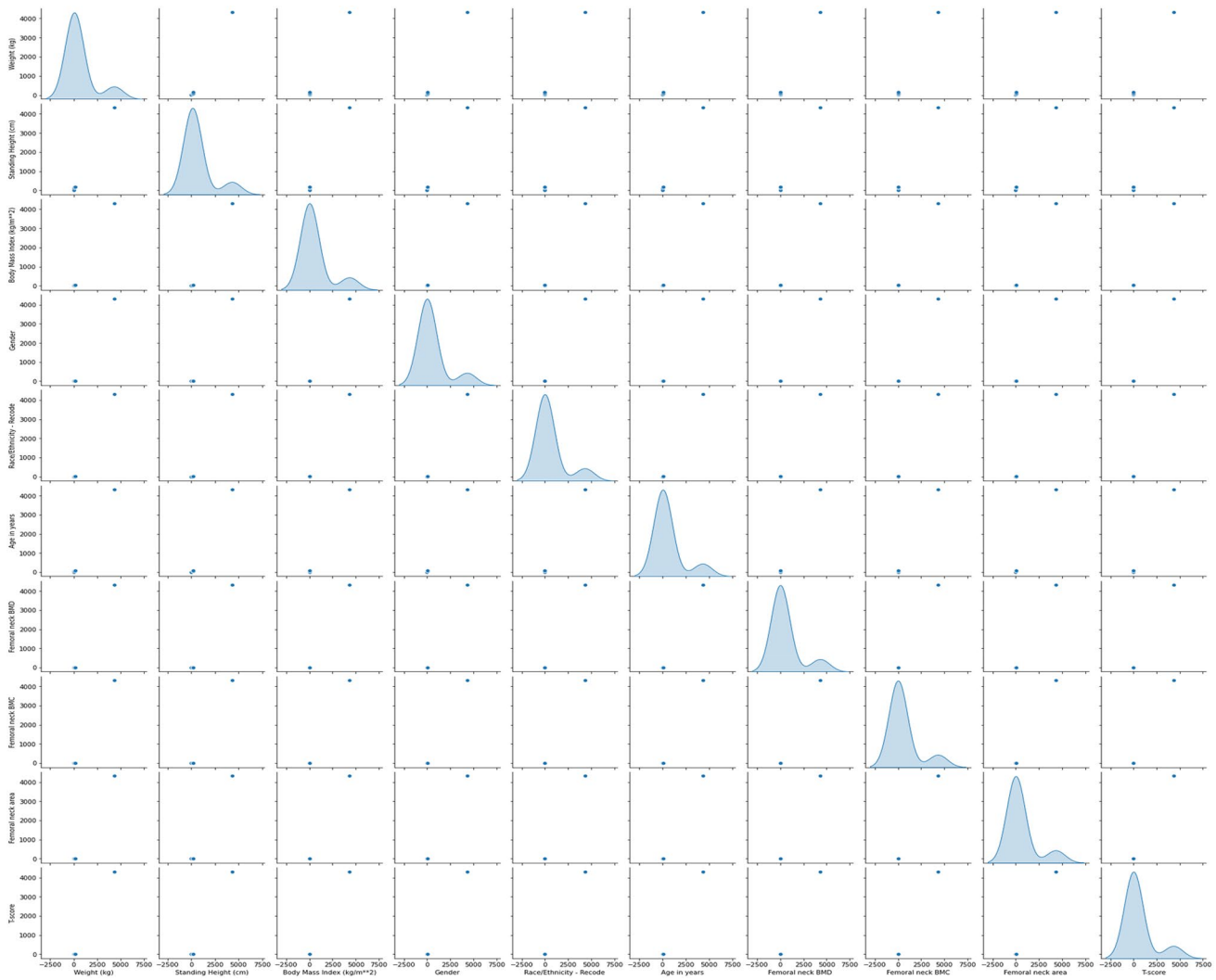


Fig. 8 Seaborn.axisgrid PairGrid

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{9}$$

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{TN})} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Specificity} = \frac{\text{T}N}{(\text{FN} + \text{FP})} \tag{12}$$

$$\text{Negative Predictive Value} = \frac{TN}{TN + FN} \tag{13}$$

$$\text{F1 score} = 2 \times \text{precision} \times \text{recall precision} + \text{recall} \tag{14}$$

The values TN and TP denote the number of genuine negative and positive outcomes, respectively, and FN and FP denote the number of negative results, and false positive respectively. Furthermore, we computed the receiver operating characteristic (ROC) curve and calculated the area under the curve (AUC) (Fig. 9).

Nine epochs were used to train the model at a learning rate of.05. Thus, we got a rather low loss with training sample and displayed the loss versus the number of epochs as well as the accuracy vs. the number of epochs.

The multi label confusion matrix function creates class- or sample-wise multi—label confusion matrix, and then in multi—class classification tasks, labels are binarized one-to-one; while confusion matrix returns a single confusion matrix for each pair of classes, as seen in Fig. 10.

The purpose of this research is to conduct a performance analysis of the categorization of osteoporosis and osteopenia data using a variety of soft and hard computing-based classification methods. From the experimental data in the preceding comparison. Sequential Deep Neural Network Model outperforms traditional or hard classification algorithms

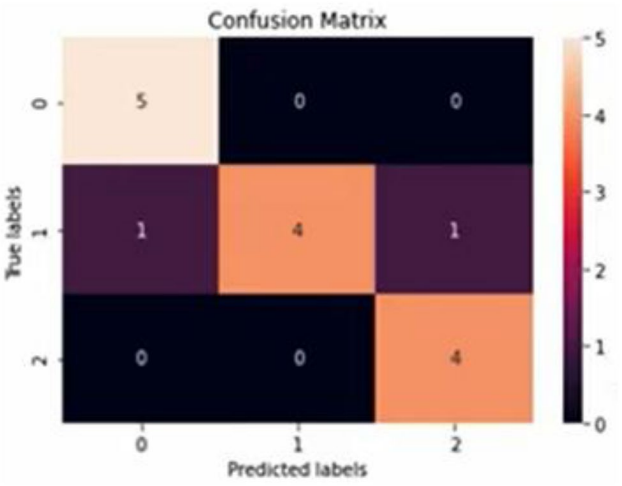


Fig. 10 Confusion matrix

on a variety of classification performance criteria such as accuracy, sensitivity, specificity, and precision. Sequential Model beat practically all other algorithms employed in soft computing-based categorization. Traditional classification algorithms outperformed their soft computing counterparts on the majority of performance metrics. Additionally, the findings differ amongst datasets. The classification model's efficiency increases with the size of the dataset it was trained on. Figure 11 illustrates the Performance Metrics.

Case 1. Comparison of algorithms with respect to Accuracy.

Accuracy is a performance indicator that indicates how well a model will perform across all classes. It is advantageous when all classes are treated equally as shown in Fig. 12. It is computed as the ratio of accurate forecasts to total predictions.

Case 2. Comparison of algorithms with respect to precision.

The precision is computed as the ratio of Positive samples accurately identified to all Positive samples classified (either

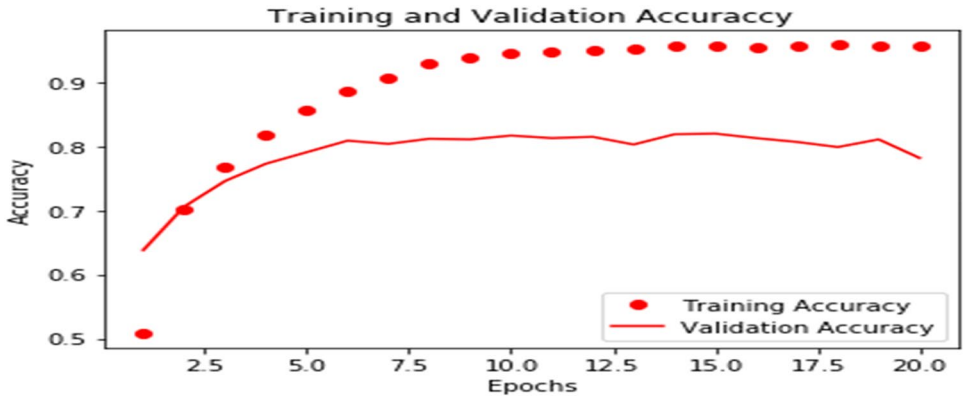


Fig. 9 Training accuracy and validation

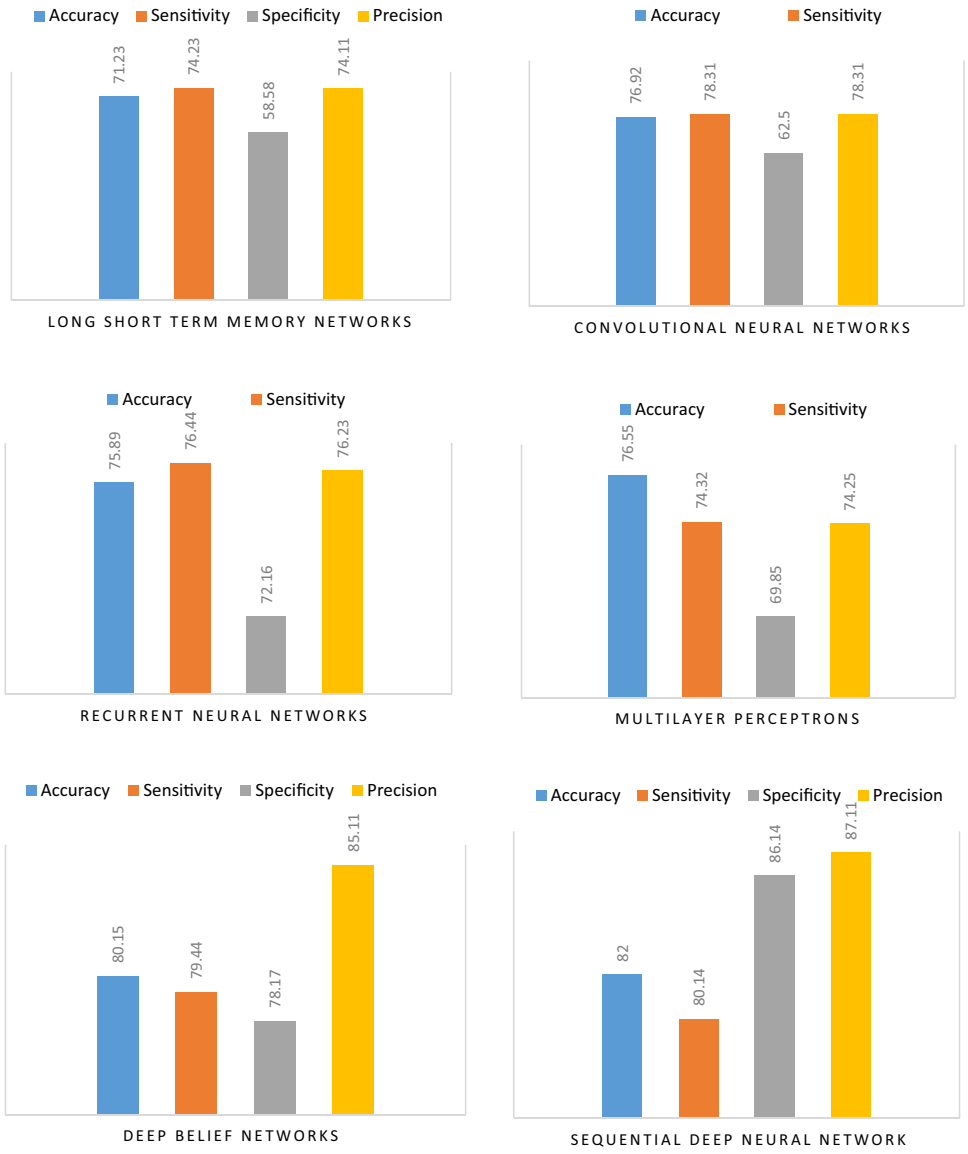


Fig. 11 Comparison of deep learning algorithms

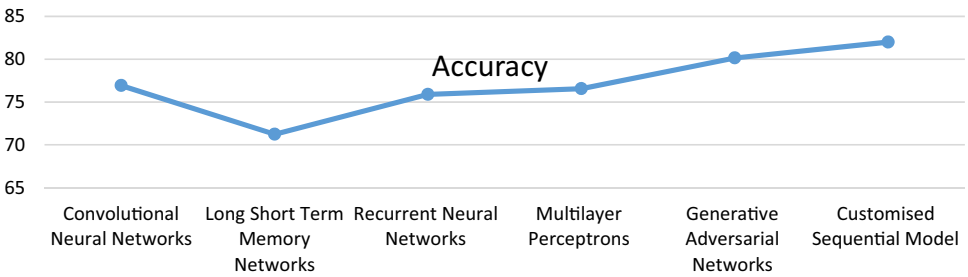


Fig. 12 Accuracy

correctly or incorrectly) as shown in Fig. 13. Precision is a metric that indicates a model's accuracy rate a sample as positive.

Case 3. Comparison of algorithms with respect to Specificity.

The specificity of a classifier is defined as the ratio of instances where the classifier properly categorised something as negative to instances where the classifier incorrectly classified something as negative as shown in Fig. 14.

Case 4. Comparison of algorithms with respect to Sensitivity.

To determine a model's sensitivity, we look at how well it can predict true positives in each of the categories that are accessible to us. As the name implies, specificity

Fig. 13 Precision

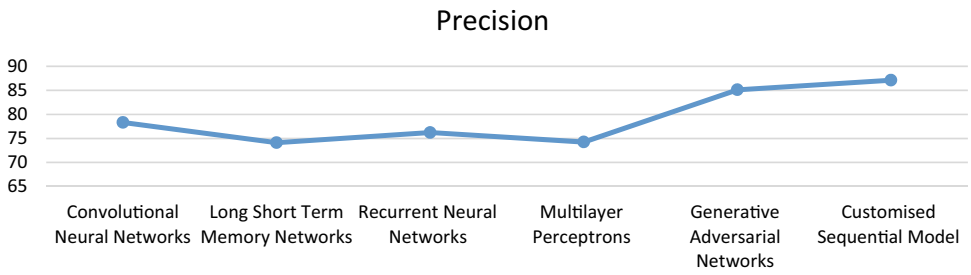


Fig. 14 Specificity

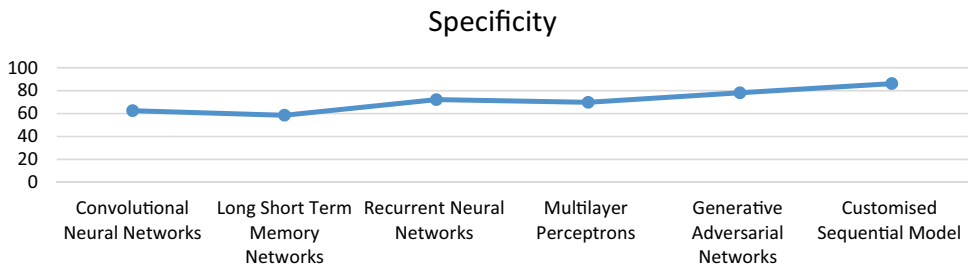


Fig. 15 Sensitivity

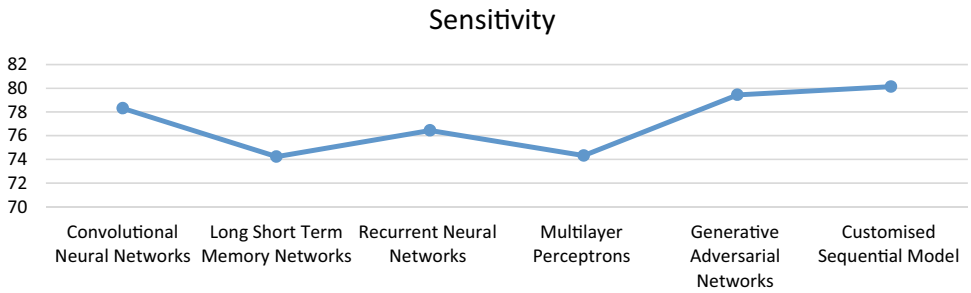
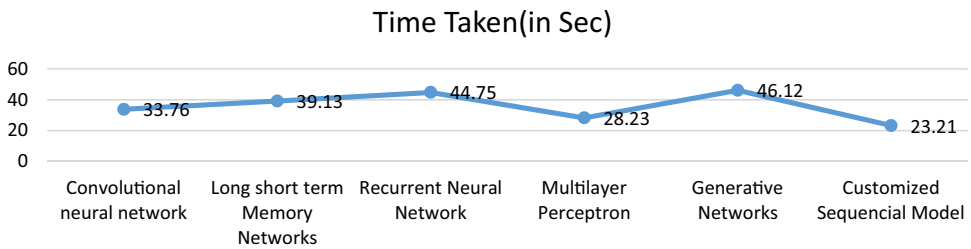


Fig. 16 Time Taken



measures how well models can predict true negatives in each of the possible categories as shown in Fig. 15. Any category model may use these measures.

Case 5. Comparison of algorithms with respect to time taken.

To estimate the time taken by the algorithms is clearly mentioned in the Fig. 16 where customized model is taking less time as per performance measure taken during implementation.

The findings and conclusions in this work are those of the researcher and do not necessarily represent the views of the Research Data Center, the National Center for Health Statistics (Centers for Disease Control and Prevention (Cdc) 2005), or the Centers for Disease Control and Prevention.

5 Conclusion

This study's findings support the idea that the Sequential model of deep neural network learning is a straightforward and effective approach to the problem of multi-class categorization. The Algorithm of Sequences The sequential model can adapt broad category of data-set requirements. There is now a basic database for training data that has been created and transferred to a powerful computer for use

during training. The Sequential model has been trained and the weight of the model has been fine-tuned, as previously stated. The model has been reworked in light of the new data. With only a few training data, the model's accuracy has improved significantly. As a result, deep learning is a "black-box" solution for many applications, despite its ability to store a large quantity of relevant data and information. Deep learning's interpretability will need to be looked at in the future.

The simultaneous modelling of numerous complex data modalities remains a challenge for deep learning. In recent deep learning research, multimodality has been a hot topic. If the datasets provided are limited, or if the data must be handled in real time, this challenge becomes much more daunting. In recent years, researchers have looked at zero-shot & one-shot learning in an effort to relieve this issue. Although the majority of deep learning implementations are supervised, machine learning is increasingly using unsupervised & semi-supervised learning to handle real world data without humans labelling. Further multimodal can be optimized using combination of suitable kernel functions.

References

Basheer IA, Hajmeer M (2000) Artificial neural networks: fundamentals, computing, design, and application. J Microbiol Methods 43(1):3–31

Baudry S, Lecoeuvre G, Duchateau J (2012) Age-related changes in the behavior of the muscle-tendon unit of the gastrocnemius medialis during upright stance. J Appl Physiol 112(2):296–304

Bennell KL, Matthews B, Greig A, Briggs A, Kelly A, Sherburn M, Wark J (2010) Effects of an exercise and manual therapy program on physical impairments, function and quality-of-life in people with osteoporotic vertebral fracture: a randomised, single-blind controlled pilot trial. BMC Musculoskelet Disord 11(1):1–11

Bergland A, Thorsen H, Kåresen R (2011) Effect of exercise on mobility, balance, and health-related quality of life in osteoporotic women with a history of vertebral fracture: a randomized, controlled trial. Osteoporos Int 22(6):1863–1871

Bonaiuti D, Arioli G, Diana G, Franchignoni F, Giustini A, Monticone M, Maini M (2005) SIMFER Rehabilitation treatment guidelines in postmenopausal and senile osteoporosis

Bronshtein A (2017) Train/test split and cross validation in python. Understanding Machine Learning

Burke TN, França FJR, Meneses SRFD, Pereira RMR, Marques AP (2012) Postural control in elderly women with osteoporosis: comparison of balance, strengthening and stretching exercises. Randomized Controll Trial Clin Rehabil 26(11):1021–1031

Centers for Disease Control and Prevention (Cdc), National Center for Health Statistics (NCHS). National Health and Nutrition Examination Survey Data, U.S. Department of Health and Human Services, Centers for Disease Control and Prevention, Hyattsville, MD, 2005–2013. https://wwwn.cdc.gov/Nchs/Nhanes.

Crepaldi G, Romanato G, Tonin P, Maggi S (2007) Osteoporosis and body composition. J Endocrinol Invest 30(6 Suppl):42–47

Crisco JJ 3rd, Panjabi MM (1991) The intersegmental and multisegmental muscles of the lumbar spine. A biomechanical model comparing lateral stabilizing potential. Spine 16(7):793–799

Deo RC (2015) Machine learning in medicine. Circulation 132(20):1920–1930

Everingham M (2009) The PASCAL visual object classes challenge 2007. In http://www.pascal-network. org/challenges/VOC/voc2007/workshop/index.html

Ezat WA, Dessouky MM, Ismail NA (2020) Multi-class image classification using Sequencial algorithm. In Journal of Physics: Conference Series (Vol 1447, No 1, p 012021). IOP Publishing

Fine TL (2006) Feedforward neural network methodology. Springer Science & Business Media

Garofolo JS, Lamel LF, Fisher WM, Fiscus JG, Pallett DS (1993) DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1–1.1. NASA STI/Recon Technical Report 93

Geiger A, Lenz P, Stiller C, Urtasus R (2013) Vision meets robotics: the KITTI dataset. Int J Robot Res 32(11):1231–1237

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In Advances in Neural Information Processing Systems. Curran Associates, pp 2672–2680

Geusens P, Dinant G (2007) Integrating a gender dimension into osteoporosis and fracture risk research. Gend Med 4:S147–S161

Girshick R (2015) Fast R-CNN. In IEEE international conference on computer vision. IEEE, pp 1440–1448

Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In IEEE conference on computer vision and pattern recognition. IEEE, pp 580–587

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: The 13th international conference on artificial intelligence and statistics, Vol. 9. JMLR. org, pp 249–256

Goller C, Kuchler A (1996) Learning task-dependent distributed representations by backpropagation through structure. In: IEEE International conference on neural networks, Vol. 1. IEEE, pp 347–352

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning, vol 1. MIT Press

Google (2016). Alphago. Retrieved from https://deepmind.com/research/alphago. Accessed April 18, 2017

Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing. IEEE, pp 6645–6649

Greenspan H, van Ginneken B, Summers RM (2016) Guest editorial deep learning in medical imaging: overview and future promise of an exciting new technique. IEEE Trans Med Imag 35(5):1153–1159

Gregor K, LeCun Y (2010) Learning fast approximations of sparse coding. In The 27th International Conference on Machine Learning. Omnipress, pp 399–406

Ha HY, Yang Y, Pouyanfar S, Tian H, Chen SC (2015) Correlation-based deep learning for multimedia semantic concept detection. In international conference on web information systems engineering. Springer, pp 473–487

Hadsell R, Erkan A, Sermanet P, Scoffier M, Muller U, LeCun Y (2008) Deep belief net learning in a long-range vision system for autonomous off-road driving. In IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 628–633

Hajimirsadeghi H, Mori G (2016) Multi-instance classification by max-margin training of cardinality-based markov networks. IEEE Trans Pattern Anal Mach Intell 39(9):1839–1852

Han K, Yu D, Tashev I (2014) Speech emotion recognition using deep neural network and extreme learning machine. In Interspeech. ISCA, pp 223–227

Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal Mach Intell 12(10):993–1001

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In IEEE conference on computer vision and pattern recognition. IEEE Computer Society, pp 770–778

He K, Gkioxari G, Dollár P, Girshick R (2017) Mask R-CNN. In IEEE international conference on computer vision. IEEE, pp 2980–2988

Horlings CGC, Küng UM, Van Engelen BGM, Voermans NC, Hengstman GJD, Van Der Kooi AJ, Allum JH (2009) Balance control in patients with distal versus proximal muscle weakness. Neuroscience 164(4):1876–1886

Huntoon EA, Schmidt CK, Sinaki M (2008). Significantly fewer refractures after vertebroplasty in patients who engage in back-extensor-strengthening exercises. In Mayo Clinic Proceedings (Vol 83, No 1, pp 54–57). Elsevier

Itoi E, Sinaki M (1994) Effect of back-strengthening exercise on posture in healthy women 49 to 65 years of age. In Mayo Clinic Proceedings (Vol 69, No 11, pp 1054–1059). Elsevier

Jalodia N, Taneja M, Davy A (2021) A deep neural network based multi-label classifier for SLA Violation prediction in a latency sensitive NFV application. IEEE Open J Commun Soc. https://doi.org/10.1001/jama.285.6.785

Jia Y, Shelhamer E, Donahue J (2014) Caffe: convolutional architecture for fast feature embedding. ACM Multimedia

Kavitha MS, Ganesh Kumar P, Park SY, Huh KH, Heo MS, Kurita T, Chien SI (2016) Automatic detection of osteoporosis based on hybrid genetic swarm fuzzy classifier approaches. Dentomaxillofacial Radiol 45(7):20160076

Klibanski A, Adams-Campbell L, Bassford T, Blair SN, Boden SD, Dickersin K, Russell WE (2001) Osteoporosis prevention, diagnosis, and therapy. J Am Med Assoc 285(6):785–795

Leetun DT, Ireland ML, Willson JD, Ballantyne BT, Davis IM (2004) Core stability measures as risk factors for lower extremity injury in athletes. Med Sci Sports Exerc 36(6):926–934

Li WC, Chen YC, Yang RS, Kuo KN, Chen CY, Tsauo JY (2013) Taiwanese Chinese translation and validation of the Quality of Life Questionnaire of the European Foundation for Osteoporosis 31 (QUALEFFO-31). J Formos Med Assoc 112(10):621–629

Mackey DC, Robinovitch SN (2006) Mechanisms underlying age-related differences in ability to recover balance with the ankle strategy. Gait Posture 23(1):59–68

MacRae PG, Lacourse M, Moldavon R (1992) Physical performance measures that predict faller status in community-dwelling older adults. J Orthop Sports Phys Ther 16(3):123–128

Møller MF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6(4):525–533

Papaioannou A, Morin S, Cheung AM, Atkinson S, Brown JP, Feldman S, Leslie WD (2010) 2010 clinical practice guidelines for the diagnosis and management of osteoporosis in Canada: summary. CMAJ 182(17):1864–1873

Pijnappels M, Bobbert MF, van Dieën JH (2005) Control of support limb muscles in recovery after tripping in young and older subjects. Exp Brain Res 160(3):326–333

Pulido M, Melin P, Castillo O (2014) Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. Inf Sci 280:188–204

Pulido M, Melin P, Castillo O (2013) Optimization of type-2 fuzzy integration in ensemble neural networks for predicting the US Dolar/MX pesos time series. In 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS) (pp 1508–1512). IEEE

Runge CF, Shupert CL, Horak FB, Zajac FE (1999) Ankle and hip postural strategies defined by joint torques. Gait Posture 10(2):161–170

Sarle WS (1996) Stopped training and other remedies for overfitting. Comput Sci Statist, 352–360

Sinaki M (2003) Critical appraisal of physical rehabilitation measures after osteoporotic vertebral fracture. Osteoporos Int 14(9):773–779

Sinaki M, Mikkelsen BA (1984) Postmenopausal spinal osteoporosis: flexion versus extension exercises. Arch Phys Med Rehabil 65(10):593–596

Sinaki M, Itoi E, Wahner HW, Wollan P, Gelzcer R, Mullan BP, Hodgson SF (2002) Stronger back muscles reduce the incidence of vertebral fractures: a prospective 10 year follow-up of postmenopausal women. Bone 30(6):836–841

Tseng WJ, Hung LW, Shieh JS, Abbod MF, Lin J (2013) Hip fracture risk assessment: artificial neural network outperforms conditional logistic regression in an age-and sex-matched case control study. BMC Musculoskelet Disord 14(1):1–11

Van Schoor NM, Knol DL, Glas CAW, Ostelo RWJG, Leplege A, Cooper C, Lips PTAM (2006) Development of the Qualeffo–31, an osteoporosis-specific quality-of-life questionnaire. Osteoporos Int 17(4):543–551

Watts NB, Lewiecki EM, Miller PD, Baim S (2008) National Osteoporosis Foundation 2008 Clinician's Guide to Prevention and Treatment of Osteoporosis and the World Health Organization Fracture Risk Assessment Tool (FRAX): what they mean to the bone densitometrist and bone technologist. J Clinic Densitometry Offi J Int Soc Clinic Densitometry 11(4):473–477

Yang F, Pai YC (2007) Correction of the inertial effect resulting from a plate moving under low-friction conditions. J Biomech 40(12):2723–2730

Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137(1–2):239–263

Zhou X, Yu K, Zhang T, Huang TS (2010) Image classification using super-vector coding of local image descriptors. In: European conference on computer vision (pp 141–154). Springer, Berlin, Heidelberg