Artificial Intelligence in Medicine 75 (2017) 51–63

Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Handling limited datasets with neural networks in medical applications: A small-data approach

Torgyn Shaikhina, Natalia A. Khovanova *

School of Engineering, University of Warwick, Coventry CV4 7A L, UK

ARTICLE INFO

Article history:
Received 12 May 2016
Received in revised form
21 November 2016
Accepted 28 December 2016

Keywords:
Predictive modelling
Small data
Regression neural networks
Osteoarthritis
Compressive strength
Trabecular bone

ABSTRACT

Motivation: Single-centre studies in medical domain are often characterised by limited samples due to the complexity and high costs of patient data collection. Machine learning methods for regression modelling of small datasets (less than 10 observations per predictor variable) remain scarce. Our work bridges this gap by developing a novel framework for application of artificial neural networks (NNs) for regression tasks involving small medical datasets.
Methods: In order to address the sporadic fluctuations and validation issues that appear in regression NNs trained on small datasets, the method of multiple runs and surrogate data analysis were proposed in this work. The approach was compared to the state-of-the-art ensemble NNs; the effect of dataset size on NN performance was also investigated.
Results: The proposed framework was applied for the prediction of compressive strength (CS) of femoral trabecular bone in patients suffering from severe osteoarthritis. The NN model was able to estimate the CS of osteoarthritic trabecular bone from its structural and biological properties with a standard error of 0.85 MPa. When evaluated on independent test samples, the NN achieved accuracy of 98.3%, outperforming an ensemble NN model by 11%. We reproduce this result on CS data of another porous solid (concrete) and demonstrate that the proposed framework allows for an NN modelled with as few as 56 samples to generalise on 300 independent test samples with 86.5% accuracy, which is comparable to the performance of an NN developed with 18 times larger dataset (1030 samples).
Conclusion: The significance of this work is two-fold: the practical application allows for non-destructive prediction of bone fracture risk, while the novel methodology extends beyond the task considered in this study and provides a general framework for application of regression NNs to medical problems characterised by limited dataset sizes.

1. Introduction

IN recent decades, a surge of interest in Machine learning within the medical research community has resulted in an array of successful data-driven applications ranging from medical image processing and the diagnosis of specific diseases, to the broader tasks of decision support and outcome prediction [1–3]. The focus of this work is on predictive modelling for applications characterised by small datasets and real-numbered continuous outputs. Such tasks are normally approached by using conventional multiple linear regression models. These are based on the assumptions of statistical independence of the input variables, linearity between dependent and independent variables, normality of the residuals, and the absence of endogenous variables [4]. However, in many applications, particularly those involving complex physiological parameters, those assumptions are often violated [5]. This necessitates more sophisticated regression models based, for instance, on Machine learning. One such approach – predictive modelling using feedforward backpropagation artificial neural networks (NNs) – is considered in this work. NN is a distributed parallel processor which resembles a biological brain in the sense that it learns by responding to the environment and stores the acquired knowledge in interneuron synapses [6]. One striking aspect of NNs is that they are universal approximators. It has been proven that a standard multilayer feedforward NN is capable of approximating any measurable function and that there are no theoretical constraints for the success of these networks [7]. Even when conventional multiple regression models fail to quantify a nonlinear relationship between causal factors and biological responses, NNs retain their

* Corresponding author.
E-mail address: n.khovanova@warwick.ac.uk (N.A. Khovanova).

http://dx.doi.org/10.1016/j.artmed.2016.12.003

capacity to find associations within high-dimensional, nonlinear and multimodal medical data [8,9].

Despite their superior performance, accuracy and versatility, NNs are generally viewed in the context of the necessity for abundant training data. This, however, is rarely feasible in medical research, where the size of datasets is constrained by the complexity and high cost of large-scale experiments. Applications of NNs for regression analysis and outcome prediction based on *small datasets* remain scarce and thus require further exploration [2,9,10]. For the purposes of this study, we define small data as a dataset with less than ten observations (samples) per predictor variable.

NNs trained with small datasets often exhibit unstable behaviour in performance, i.e. sporadic fluctuations due to the sensitivity of NNs to initial parameter values and training order [11–13]. NN initialisation and backpropagation training algorithms commonly contain deliberate degrees of randomness in order to improve convergence to the global minimum of the associated cost function [6,9,12,14]. In addition, the order with which the training data is fed to the NN can affect the level of convergence and produce erratic outcomes [12,13]. Such inter-NN volatility limits both the reproducibility of the results and the objective comparison between different NN designs for future optimisation and validation. Previous attempts [15] to resolve the stability problems in NNs demonstrated the success of *k*-fold cross-validation and ensemble methods for a medical *classification* problem; the dataset comprised 53 features and 1355 observations, which corresponds to 25 observations per predictor variable. To the best of our knowledge, effective strategies for *regression* tasks on small biomedical datasets have not been considered, thus necessitating the establishment of a framework for application of NNs to medical data analysis.

One important biomedical application of NNs in hard tissue engineering was considered in our previous work [11,16], where a NN was applied for correlation analysis of 35 trabecular bone samples from male and female specimens of various ages suffering from severe osteoarthritis (OA) [17]. OA is common degenerative joint disease associated with damaged cartilage [18]. Unlike in osteoporosis, where decreasing bone mineral density (BMD) decreases bone compressive strength (CS) and increases bone fracture risk, the BMD in OA was seen to *increase* [19,20]. There is further indication that higher BMD does not protect against bone fracture risk in OA [19,21]. The mathematical relationship between BMD and CS observed in healthy patients does not hold for patients with OA, necessitating development of a CS model for OA.

In the current work, we consider the application of NNs to osteoarthritic hip fracture prediction for non-invasive estimation of bone CS from structural and physiological parameters. For this particular application there are two commonly used computational techniques: quantitative computed tomography-based finite element analysis [22,23] and the indirect estimation of local properties of bone tissue through densitometry [24,25]. Yet, *subject-specific* models for hip fracture prediction from structural parameters of trabecular bone in patients affected by degenerative bone diseases have not been developed. An accurate patient data driven model for CS estimation based on NNs could offer a hip fracture risk stratification tool and provide valuable clinical insights for the diagnosis, prevention and potential treatment of OA [26,27].

The *aim* of this research is to develop subject-specific models for hip fracture prediction in OA and a general framework for the application of regression NNs to small datasets. In this work we introduce the *method of multiple runs* to address the inter-NN volatility problem caused by small data conditions. By generating a large set (1000+) of NNs, this method allows for consistent comparison between different NN designs. We also propose *surrogate data test* in order to account for the random effects due to small datasets. The use of surrogate data was inspired by their successful application in nonlinear physics, neural coding, and time series analysis [28–30].

The utility of the proposed framework was explored by considering a larger dataset. Due to the unavailability of a large number of bone samples, a different CS dataset, that of 1030 samples of concrete, was used [31,32]. We designed and trained regression NNs for several smaller subsets of the data and demonstrated that small-dataset (56 samples) NNs developed using our framework can achieve a performance comparable to that of the NNs developed on the entire dataset (1030 samples).

The structure of this article is as follows. Section 2 describes the data used for analysis, NN model design, and introduces the new framework. In Section 3, the role of data size on NN performance and generalisation ability is explored to demonstrate the utility of the proposed framework. In Section 4 we apply our framework for prediction of osteoarthritic trabecular bone CS and demonstrate the superiority of the approach over established ensemble NN methods in the context of small data. Section 5 discusses both the methodological significance of the proposed framework and the medical application of the NN model for prediction of hip fracture risk. Additional information on NN outcomes and datasets is provided in the Appendices.

2. Methodology

2.1. Porous solids: data

2.1.1. Compressive strength of trabecular bone

Included in this study are 35 patients who suffered from severe OA and underwent total hip arthroplasty (Table A1, Appendix A). The original dataset [17] obtained from trabecular tissue samples taken from the femoral head of the patients contained five predictor features (a 5-D input vector for the NN): patients' age and gender, tissue porosity (BV/TV), structure model index (SMI), trabecular thickness factor (tb.th), and one output variable, the CS (in MPa). The dataset was divided at random into training (60%), validation (20%) and testing (20%) subsets, i.e. 22, 6 and 7 samples, respectively.

2.1.2. Compressive strength of concrete

The dataset [31] of 1030 samples was obtained from a publically available repository [32] and contained the following variables: compressive strength (CS) of concrete samples (in MPa), the amounts of 7 components in the concrete mixture (in kg/m^3): cement, blast furnace slag, fly ash, water, superplasticizer, coarse and fine aggregates, and the duration of concrete aging (in days). The CS of concrete is a highly nonlinear function of its components and the duration of aging, yet an appropriately trained NN can effectively capture this complex relationship between the CS and the other 8 variables. A successful application of NNs to CS prediction based on 700 concrete samples has been demonstrated in an original study by Yeh [31]. For the purposes of our NN modelling, the samples were divided at random into training (60%), validation (10%) and testing (30%). Thus, out of 1030 available samples, 630 were used for NN training, 100 for validation and 300 were reserved for testing.

2.2. NN design for CS prediction in porous solids

Considering the size and nature of the available data, a feedforward backpropagation NN with one hidden layer, *p* input features and one output was chosen as the base for the CS model (Fig. 1). The *k* neurons in the hidden layer is characterised by a hyperbolic tangent sigmoid transfer function [33], while the output neuron relates the CS output to the input by using a simple linear transfer function (Fig. 1).

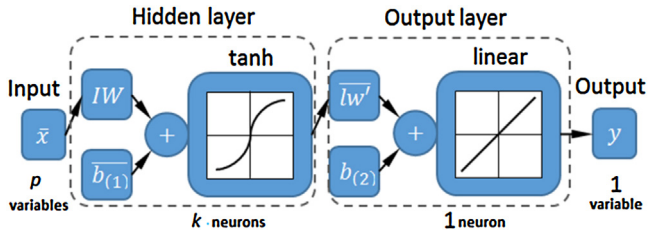

Fig. 1. Neural network model topology and layer configuration represented by a p-dimensional input, k-neuron hidden layer and 1 output variable.

The p-by-k input weights matrix IW, k-by-1 layer weights column vector $\overline{lw'}$, and the corresponding biases $\overline{b^{(1)}}$ and $b^{(2)}$ for each layer were initialised according to the Nguyen-Widrow method [34] in order to distribute the active region of each neuron in the layer evenly across the layer's input space.

The NNs were trained using the Leverberg-Marquardt backpropagation algorithm [35–37]. The cost function was defined by the mean squared error (*MSE*) between the output and actual CS values. Early stopping on an independent validation cohort was implemented in order to avoid NN overtraining and increase generalisation [38]. The validation subset was sampled at random from the model dataset for each NN, ensuring a diversity among the samples. The resulting NN model mapped the output y (in MPa) to the input vector $\bar{x}$ is:

$$y = \tanh\left[\bar{x} \cdot IW + \overline{b^{(1)}}\right] \cdot \overline{lw'} + b^{(2)} \quad (1)$$

The final values of the weights and bias parameters in (1) for the trained bone data NN are provided in Table A3 in Appendix B.

Note, parameter estimation for the optimal network structure, size, training duration, training function, neural transfer function and cost function was conducted at the preliminary stage following an established textbook practice [6,9]. Assessment and comparison of various NN designs were carried out using the multiple runs technique.

2.3. Method of multiple runs

In order to address the small dataset problem we introduce the method of multiple runs in which a large number of NNs of the same design are trained simultaneously. In other words, the performance of a given NN design is assessed not on a single NN instance, but repeatedly on a set (multiple run) of a few thousands NNs. Identical in terms of their topology and neuron functions, NNs within each such run differ due to the 3 sources of randomness deliberately embedded in the initialisation and training routines: (a) the initial values of the layer weights and biases, (b) the split between the training and validation datasets (test samples were fixed), and (c) the order with which the training and validation samples are fed into the NN. In every run, several thousand NNs with various initial conditions are generated and trained in parallel, producing a range of successful and unsuccessful NNs evaluated according to criteria set in Section 2.7. Subsequently, their performance indicators are reported as collective statistics across the whole run, thus allowing consistent comparisons of performance among runs despite the limited size of the dataset. This helps to quantify the varying effects of design parameters, such as the NN's size and the training duration during the iterative parameter estimation process. Finally, the highest performing instance of the *optimal NN design* is selected as the working model. This strategy principally differs from NN ensemble methods (as discussed below in Section 2.6) in the sense that only the output of a single best performing NN is ultimately selected as *the working (optimal) model*.

In summary, the following terminology applies throughout the paper:

- design parameters are NN size, neuron functions, training functions, etc.
- individual NN parameters are weights and biases
- optimal NN design is based on estimation of appropriate NN size, topology, training functions, etc.
- working (optimal) model is the highest performing instance selected from a run of the optimal NN design.

The choice of the number of NNs per run is influenced by the balance between the required precision of the statistical measures and computational efficiency, as larger runs require more memory and time to simulate. It was found that for the bone CS application considered in this study, 2000 NNs maintained most performance statistics, such as mean regression between NN targets and predictions, consistent to 3 decimal places, which was deemed sufficient. For inter-run consistency each 2000 NN run was repeated 10 times, yielding 20,000 NNs in total. The average simulation time for instantiating and training a run of 2000 NNs on a modern PC (Intel® Core™ i7-3770 CPU @3.40 GHz, 32 GB RAM) was 280 s.

2.4. Surrogate data test

Where a sufficient number of samples is available, the efficiency of learning by NN of the interrelationships in the data is expected to correlate with its test performance. With small datasets, however, the efficiency of learning is decreased and even poorly-designed NNs can achieve a good performance on test samples at random. In order to avoid such situation and to evaluate NN performance in the presence of random effects, *a surrogate data test* is proposed in this study. Surrogate data mimics the statistical properties of the original dataset independently for each component of the input vector. While resembling the statistical properties of the original data, the surrogates do not retain the intricate interrelationships between the various components of the real dataset. Hence, the NN trained and tested on surrogates is expected to perform poorly.

Numerous surrogate data NNs are generated using method of multiple runs described in Section 2.3. The highest performing surrogate NN instance defines the lowest performance threshold for real data models. To pass the surrogate data test, real data NNs must outperform this threshold.

The surrogate samples can be generated using a variety of methods [29,39,40]. In this study two approaches were used. For trabecular bone data, all continuous input variables were normally distributed according to the Kolmogorov-Smirnov statistical test [4]. Thus surrogates were generated from random numbers to match the truncated normal distributions, e.g. mean and standard deviation estimated from the original data, as well as the range and size of the original tissue samples (Table A2, Appendix A). For the concrete data, where vector distributions were not normal, random permutations [4] of the original vectors were applied.

2.5. Summary of the proposed framework

Combined, the method of multiple runs and surrogate data test comprise a framework for application of regression NNs to small datasets, as summarised in Fig. 2. Multiple runs enable (i) consistent comparison of various NN designs during design parameter estimation, (ii) comparison between surrogate data and real data NNs during surrogate data test, and (iii) selection of the working model among the models of optimal design.

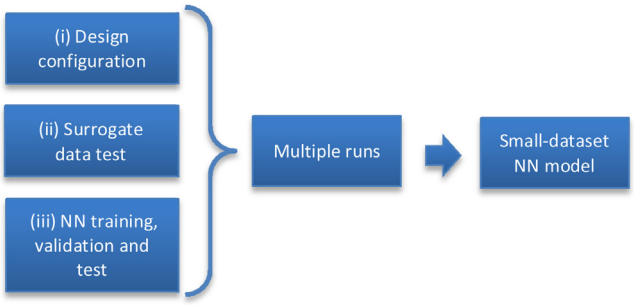

Fig. 2. Proposed framework for application of regression neural networks to small datasets.

2.6. Assessing NN generalisation

In the context of ML, *generalising performance* is a measure of how well a model predicts an outcome based on independent test data with which the NN was not previously presented. In recent decades considerable efforts in ML have been dedicated to improving the generalisation of NNs [41,42]. A data-driven predictive model has little practical value if it is not able to form accurate predictions on new data. Yet in small datasets, where such test data are scarce, the simple task of assessing generalisation becomes impractical. Indeed, reserving 20% of the bone data for independent testing leaves us with only 7 samples. The question of whether the NN model would generalise on a larger set of new samples cannot be illustrated with such limited test data. This poses a major obstacle for small medical datasets in general, thus the effect of dataset size on NN performance must be considered. We investigate the effect of the model dataset size on the generalisation ability of the NN models developed with our framework on a large dataset of concrete CS samples described in Section 2.1. The findings are presented in Section 3.4.

2.7. Performance criteria

In order to assess the performance of an *individual* NN, including the best performing, the linear regression coefficients R between the actual output (target) and predicted output were calculated. In particular, regression coefficients were calculated for the entire dataset (R_{all}), and separately for training (R_{train}), validation (R_{val}), and testing (R_{test}). R can take values between 0 and 1, where 1 corresponds to the highest model predictive performance (100% accuracy) with equal target and prediction values. R greater than 0.6 defines statistically significant performance, i.e. $R_{all} \geq 0.6$, $R_{train} \geq 0.6$, $R_{val} \geq 0.6$, and $R_{test} \geq 0.6$⟨REFNUMLINK⟩ [11].

The root mean squared error (*RMSE*) across the entire dataset was also assessed. *RMSE* presents the same information regarding model accuracy as the regression coefficient$R$, but in terms of the *absolute* difference between NN predictions and targets. *RMSE* helps to visualise the predictive error since it is expressed in the units of the output variable, i.e. in MPa for CS considered in this work.

The *collective* performance of the NNs within a multiple run was evaluated based on the following statistical characteristics:

- mean μ and standard deviation σ of R_{test} and R_{all} averaged across all NNs in the run,
- the number of NNs that are statistically significant,
- the random effect threshold $R_{sur,max}$ set by the highest performing surrogate NN, in terms of R_{all} and R_{test}.

In order to select the *best performing* NN in a run, we considered both R_{train} and R_{val}. Commonly the validation subset is used for model selection [9], however under small-data conditions, R_{val} is unreliable. On the other hand, although R_{tr} does not indicate the NN performance on new samples, it gives a useful estimation of the highest expected NN performance. It is expected that R_{tr} is higher than R_{val} for a trained NN. Subsequently, when selecting the best performing NN, we disregard models with $R_{val} > R_{train}$and from the remaining models we choose the one with the highest R_{val}. Note that R_{test} should not be involved in the model selection as it reflects the *generalising performance* of NN models on new data.

2.8. Alternative model: NN ensemble methods

Ensemble methods refer to powerful ML models based on combining predictions of a series of individual ML models, such as NNs, trained independently [43,44]. The principle behind a good ensemble is that its constituent models are diverse and are able to generalise over different subsets of an input space, effectively offsetting mutual errors. The resulting ensemble is often more robust than any of its constituent models and has superior generalisation accuracy [43,44]. We compared the NN ensemble performance with that of a single NN model developed within the proposed multiple runs framework for both the concrete and bone applications.

In an ensemble, the constituent predictor models can be diversified by manipulating the training subset, or by randomising their initial parameters [44]. The former comprises boosting and bagging techniques, which were disregarded as being impractical for the small datasets, as they reduced already scarce training samples. We utilised the latter ensembling strategy, where each constituent NN was initialised with random parameters and trained with the complete training set, similar to the multiple runs strategy described in Section 2.3. Optiz & Maclin showed that this ensemble approach was "surprisingly effective, often producing results as good as Bagging" [43]. The individual predictions of the constituent NNs were combined using a common linear approach of simple averaging [45].

2.9. Statistical analysis

A non-parametric Wilcoxon rank sum test, also known as the Mann–Whitney U test, for medians was utilised for comparing the performances of any two NN runs [46]. The null-hypothesis of no difference between the groups was tested at the 5% significance level and this is presented by p-values.

3. Investigations of the effect of data size on NN performance: concrete CS models

In this section, we utilise a large dataset on concrete CS, described in Section 2.1.2, to investigate the role of dataset size on NN performance and generalising ability. It is demonstrated that for a larger number of samples the optimal NN coefficients can be derived without involving the proposed framework, yet the importance of the framework increases as the data size is reduced.

3.1. Collective NN performance (per run)

First, a large-dataset NN model was developed on a complete dataset of 1030 samples, out of which 30% (300 samples) were reserved for tests. The NN was designed as in Fig. 1, with $p = 8$ inputs and $k = 10$ neurons in hidden layer. In a multiple run of 1000, all large-data NNs performed with statistically significant regression coefficients ($R > 0.6$). As expected with large data, the collective performance was highly accurate, with $\mu(R_{all}) = 0.95$ and $\mu(R_{test}) = 0.94$ when averaged across the multiple run of 1000 NNs. (Fig. 3a)

Secondly, a NN was applied to a smaller subset of the original dataset (Fig. 3b). Out of 1030 concrete samples, 100 samples were sampled at random and without replacement [4]. The proportions

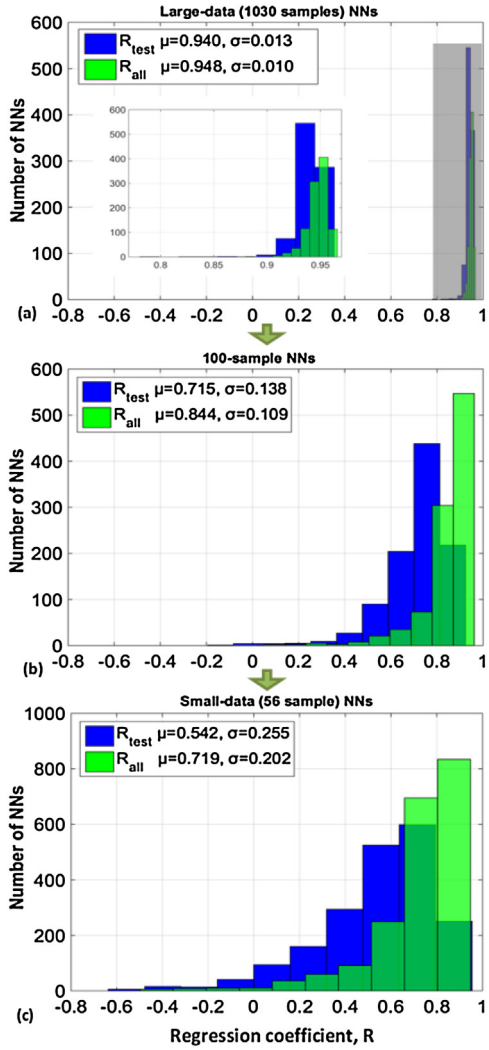

Fig. 3. Distributions of regression coefficients R_{all} and R_{test} across a run of neural networks: (a) large-dataset model (1030 samples), (b) intermediate 100 sample model, and (c) small-dataset model (56 samples). The inset shows the enlarged area highlighted in (a).

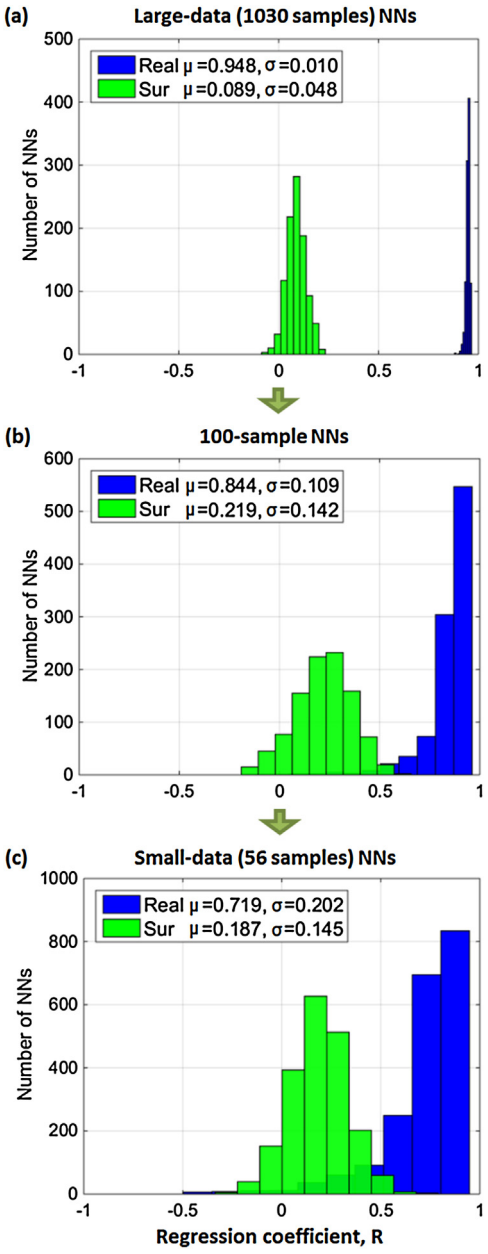

Fig. 4. Distributions of regression coefficients achieved by small-dataset neural networks for surrogates (green) and real concrete data (navy) for (a) large-dataset model (1030 samples), (b) intermediate 100 sample model, and (c) small-dataset model (56 samples). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for training, validation and testing subsets, as well as the training and initialisation routines, were analogous to those used for the large concrete dataset NN with an exception to the following adjustments:

- 2000 and not 1000 NNs were evaluated per run to ensure inter-run repeatability,
- the number of neurons in the hidden layer was reduced from 10 to 5 and the number of maximum fails for early stopping was decreased from 10 to 6 to account for a dataset size reduction.

Finally, an extreme case with even smaller subset of the data was considered (Fig. 3c). From the concrete CS dataset with 8 predictors, 56 samples were selected at random to yield the same ratio of the number of observations per predictor variable as in the bone CS dataset (35 samples and 5 predictors). The small-dataset NN based on 56 concrete samples was modelled on 41 samples and initially tested on 15 samples.

Fig. 3 illustrates the changes to the regression coefficient distributions as the size of the dataset decreased from (a) 1030 to (b) 100, and to (c) 56 samples.

In comparison to the large-dataset NNs (Fig. 3a), the distributions of the regression coefficients along x-axis for smaller dataset NNs (Fig. 3b and c) were within much wider ranges. The standard deviations σ also increased substantially for NN modes based on smaller datasets compared with the initial large-dataset model (Fig. 3a). Distributions of the regression coefficients achieved by the 2000 NN instances within the same run (Fig. 3c) demonstrate higher intra-run variance when compared to the large-dataset NNs (Fig. 3a). Over half of the NNs did not converge and only 762 NNs produced statistically significant predictions.

The mean regression coefficients across the run decreased to $\mu(R_{all}) = 0.719$, and $\mu(R_{test}) = 0.542$ (Fig. 3c). When considering only statistically significant NNs ($R > 0.6$), the mean performance of all samples was $\mu(R_{all,signif}) = 0.839$ and individually for tests $\mu(R_{test,signif}) = 0.736$. Despite higher volatility, an undesirable distribution spread and lower mean performance, the maximal R values

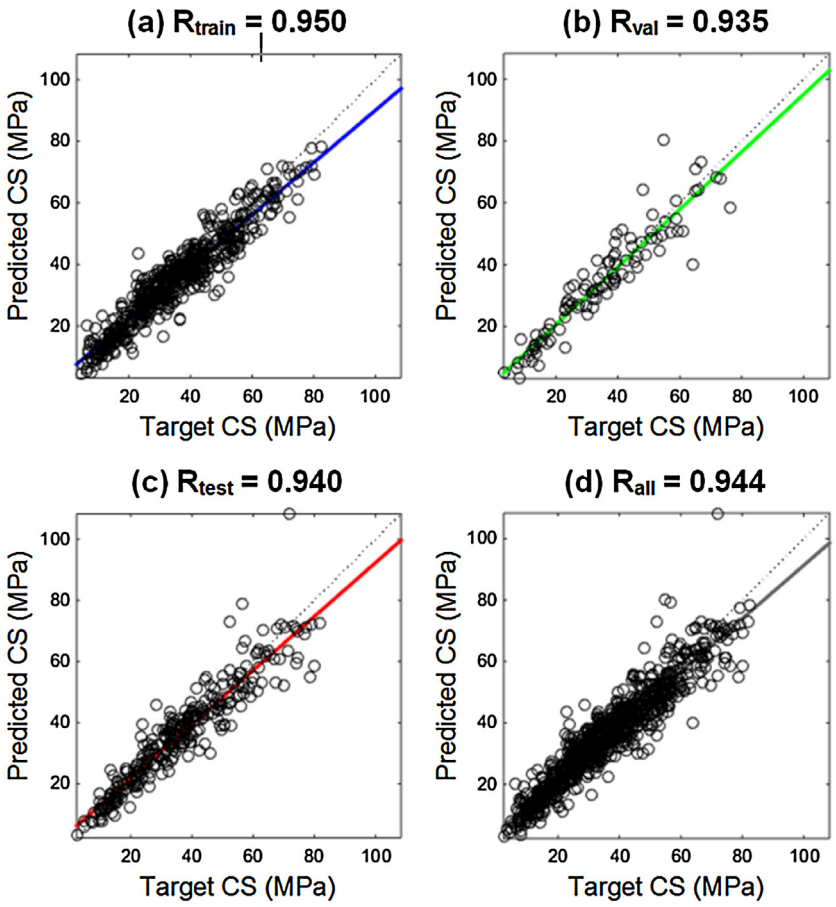

Fig. 5. Linear regression between target and predicted compressive strength achieved by the specimen large-data (1030 samples) concrete neural network model. Values are reported individually for (a) training (blue), (b) validation (green), (c) testing (red), and (d) the entire dataset (black). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

for the small-dataset NNs were comparable with those for the large-dataset NNs.

3.2. Surrogate data test: interpretation for various dataset sizes

As expected, NNs trained on the real concrete data consistently outperformed surrogate NNs. Fig. 4 demonstrates how the difference in performance between the real and surrogate NNs increased with the dataset size.

For the large-dataset NN developed with 1030 samples (Fig. 4a), the surrogate and real-data NN distributions did not overlap. In fact, the surrogate NNs in this instance achieved approximately zero mean performance, which signifies that random effects would not have an impact on NN learning with a dataset of this size.

The 100-sample and 56-sample surrogate NNs had a non-zero mean performance of $\mu(R_{all,sur,100}) = 0.219$ (Fig. 4b) and $\mu(R_{all,sur,56}) = 0.187$ (Fig. 4c), respectively. They were also characterised by a higher standard deviation of $\delta = 0.142$ and $\delta = 0.145$ compared to large-dataset NNs ($\delta = 0.048$). The non-zero mean performance of NNs suggests that random effects cannot be disregarded with small datasets and require quantification offered by the proposed surrogate data test.

For 56-sample datasets (Fig. 4c), the surrogate NNs performed with an average regression of $\mu(R_{all,sur,56})$=0.187, as opposed to $\mu(R_{all,real,56}) = 0.715$ for real-data NNs. None of the 2000 surrogate small-dataset NNs achieved a statistically significant performance ($R \geq 0.6$) across all regressioin coefficients, i.e. R_{train}, R_{val} and R_{test}. The surrogate threshold for the 56-sample NN was considered: the highest performing surrogate NN achieved $R_{sur,max,56} = 0.791$ on training dataset. This was largely due to overtraining, as its corresponding performance on test samples was poor ($R_{test} = 0.515$).

3.3. Individual NN performance

This subsection compares performance of individual NNs: a large-dataset NN (1030 samples) and a small-dataset NN (56 samples) developed using the proposed framework. As shown in Fig. 3a, all large-data NNs performed with high accuracy and small variance, thus one of them could be selected as a working model without the need for multiple runs. The performance of one of 1000 large-data NN from the run in Fig. 3a is demonstrated in Fig. 5. This NN achieved (R_{all}) = 0.944 and generalised with (R_{test}) = 0.94 on 300 independent test samples (Fig. 5d). This large-dataset model provides an indication of NN performance achieved with abundant training samples.

For small datasets, we are now concerned with NNs that perform above the surrogate data threshold of $R_{sur,max,56} = 0.791$ established in Section 3.2. Among the 2000 small-dataset (56-sample) NNs, the best-performing NN was selected using the performance criteria in Section 2.7. This model achieved regression coefficients of (R_{all}) = 0.92 on the entire dataset, and separately: (R_{train}) = 0.96, (R_{val}) = 0.92 and (R_{test}) = 0.90 on 15-sample test (Fig. 6a–d). In comparison, the large-dataset NN developed with 1030 samples performed only 2.12% higher. The *R*values were well above the

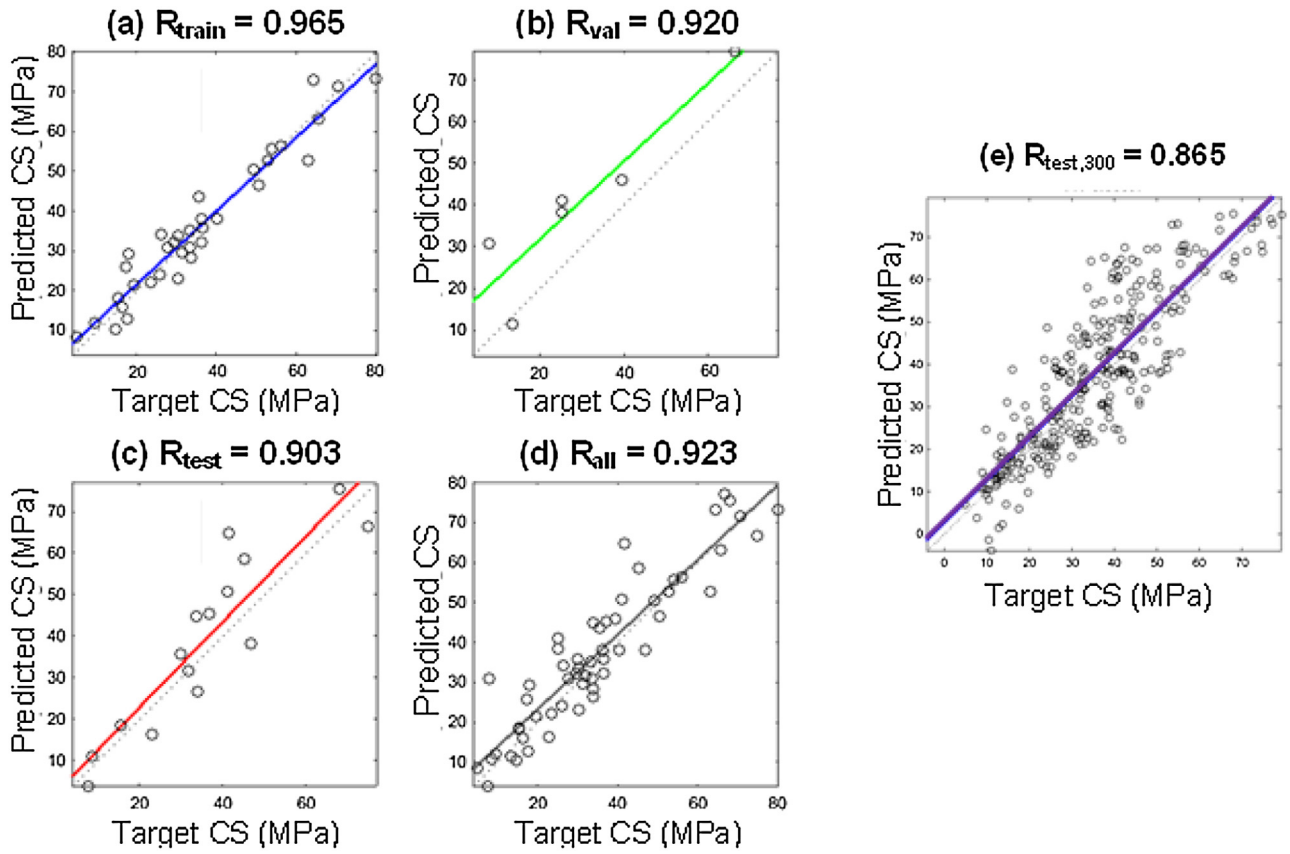

Fig. 6. Linear regression between target and predicted compressive strength achieved by the small-dataset (56 samples) optimised concrete neural network. Values are reported individually for (a) training (blue), (b) validation (green) and (c) testing (red), (d) the entire dataset (black), and (e) for 300 independent test samples (purple). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

surrogate threshold, indicating that high performance of the small-data NN was not due to luck. This result was confirmed when the small-data NN was subjected to the generalisation assessment on new test samples.

3.4. Generalising performance of the small-dataset NN

In order to assess generalisation, 300 new test samples were randomly selected from the available dataset of 1030–56 = 974 samples not previously seen by the NN. Modelled with only 41 samples, the NN was able to predict CS on 300 new test samples with $R_{test,300}$ = 0.865 (Fig. 6e); the corresponding *RMSE* was 9.5 MPa. This constitutes a 7.5% decrease in generalising performance compared to the specimen large-dataset NN tested with the same number of independent samples (Fig. 5c).

In other words, using the proposed framework we were able to develop an 86.5% accurate NN model with an 18 times smaller dataset than the original one, which demonstrates superiority of the suggested methodology and its applicability to the problems characterised by restricted dataset sizes.

3.5. Comparison of the small-dataset NN with the ensemble model for the concrete CS data

Firstly, an NN ensemble was designed by combining the outputs of 1000 NNs trained with the complete dataset of concrete samples (analogous to the large-dataset NNs described in Section 3.1 and presented in Fig. 3a). As anticipated, this NN ensemble was able to achieve a superior generalisation accuracy of R_{test} = 0.96 when tested on 300 independent samples.

The second NN ensemble was designed by combining the 2000 56-sample NNs (analogous to the small-dataset NNs in Section 3.1 and Fig. 3c). This ensemble achieved R_{test} = 0.81 on 15 independent test samples. In comparison, our small-dataset concrete NN model developed with the multiple runs technique achieved R_{test} = 0.903 on the same test samples. Subsequently the generalising ability of this ensemble was assessed on 300 additional concrete samples. The ensemble was able to retain its generalising ability with the accuracy of $R_{test,300}$ = 0.81, proving its robustness, irrespective of the test sample size.

Despite such striking consistency, the accuracy of the ensemble model was decreased by over 8% when compared with the generalising performance of the single NN model, developed using method of multiple runs ($R_{test,300}$ = 0.865, Section 3.3). These results demonstrate that a NN ensemble can achieve a remarkable performance on predictive tasks with sufficient data, but is unable to perform as well as the multiple runs model on small datasets.

4. Results: bone CS model

4.1. NN design configuration

The NN design described in Section 2.2 for bone CS data comprised 5 input parameters. The heterogeneous 1×5 input vector, $\bar{x}$, was stacked in the following order: x_1 = morphology (SMI), x_2 = level of interconnectivity (tb.th), x_3 = porosity (BV/TV), x_4 = age and x_5 = gender. Following a standard parameter estimation routine, but with the help of multiple runs, the NN design was configured to 4 neurons in the hidden layer (Appendix B). The number of permissible consecutive validation iterations during which

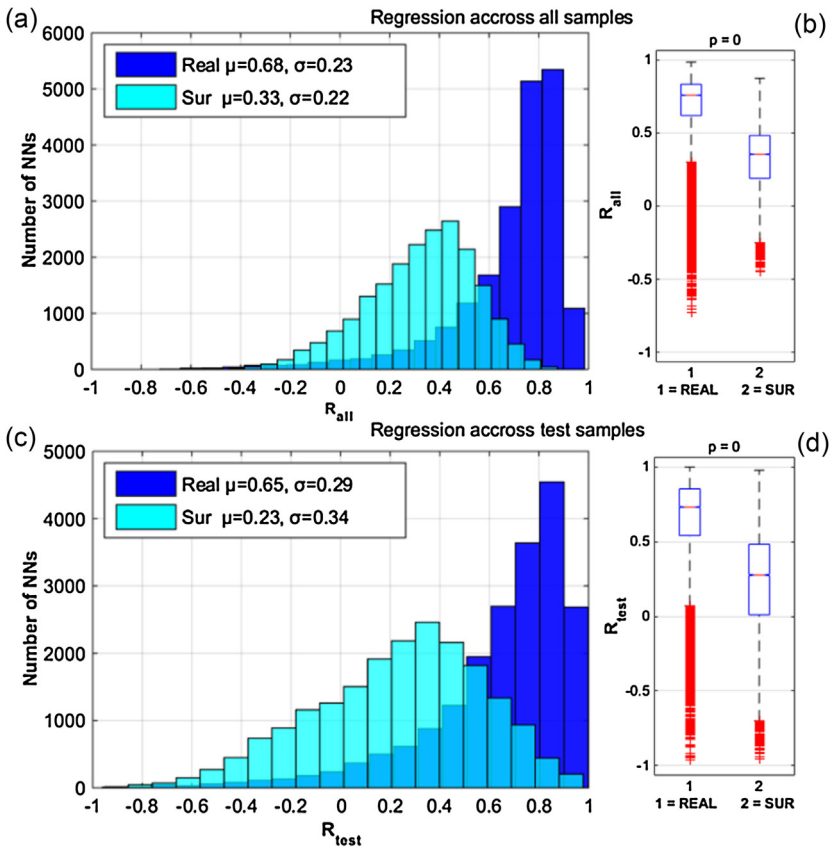

Fig. 7. Distributions (a) of regression coefficients achieved by neural networks for surrogates (light blue) and real bone data (navy) and (b) Wilcoxon rank sum test for medians across *all* samples. Distributions and Wilcoxon test results across *test* samples are reported in (c) and (d). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the NN performance fails to improve and which directly influences duration of NN training, was set to 9 (Appendix B).

4.2. Surrogate data test

Performances of the NNs trained with real and surrogate data were compared by assessing 10 runs of 2000 NNs, i.e. a total of 20,000 NNs. The real dataset NNs consistently outperformed the surrogate NNs with, on average, a 35% performance increase (Fig. 7a).

Wilcoxon rank sum tests for median R_{all} and R_{test} across 20,000 NNs revealed significant statistical difference ($p = 0$) between the groups, with median $R_{all,sur} = 0.38$ for surrogates versus median $R_{all,real} = 0.78$ for the real dataset (Fig. 7b). Similar differences in the distributions of $R_{test,real}$ and $R_{test,sur}$ were observed for tests samples (Fig. 7c–d). The surrogate threshold was $R_{sur,max} = 0.87$ which indicated the lower performance threshold for the real dataset NN. Overall, the surrogate test signified that the accurate results yielded by the bone NN model are not due to random effects.

4.3. Optimal bone CS model

Among the run of 2000 NNs of optimal design, the best-performing NN was capable of predicting trabecular tissue CS with *RMSE* = 0.85 MPa on the test samples. The linear regression coefficients between targets and predictions achieved by the NN were: individually for $R_{tr} = 0.999$, $R_{val} = 0.991$, $R_{test} = 0.983$ and $R_{all} = 0.993$ (Fig. 8a–d). This indicates a very high accuracy of predictions despite the limited dataset of 35 samples. The final values of weights and biases of this fully-trained network are provided in Appendix C.

4.4. Comparison with ensemble NN

The NN ensemble achieved $R_{test} = 0.882$, which is 11% lower than the accuracy of the proposed multiple run NN model ($R_{test} = 0.983$) and only marginally higher than the surrogate threshold$R_{sur,max} = 0.87$ established in Section 4.2 for the bone dataset. This result further confirms that the NN ensembles, when tasked with small-dataset applications, were unable to realise their full predictive potential and were inferior to NNs designed within a multiple runs framework.

5. Discussion

5.1. Significance of the proposed methodology

A framework for the application of regression NNs to medical datasets has been developed in order to mitigate the small dataset problem. NNs trained with small datasets exhibit sporadic fluctuations in the performance due to degrees of randomness inherent in the NN initialisation and training routines. This raises the problem of consistent comparisons between various NN models. Another problem is the evaluation of NN performance in the presence of random effects when the test data are scarce. The limitations of small datasets have been overcome in this work by using a novel framework comprising: (1) a multiple runs strategy for monitoring the performance measures collectively across a large set of NNs, and (2) surrogate data analysis for model validation. The proposed surrogate data approach provided a mechanism for NN model validation

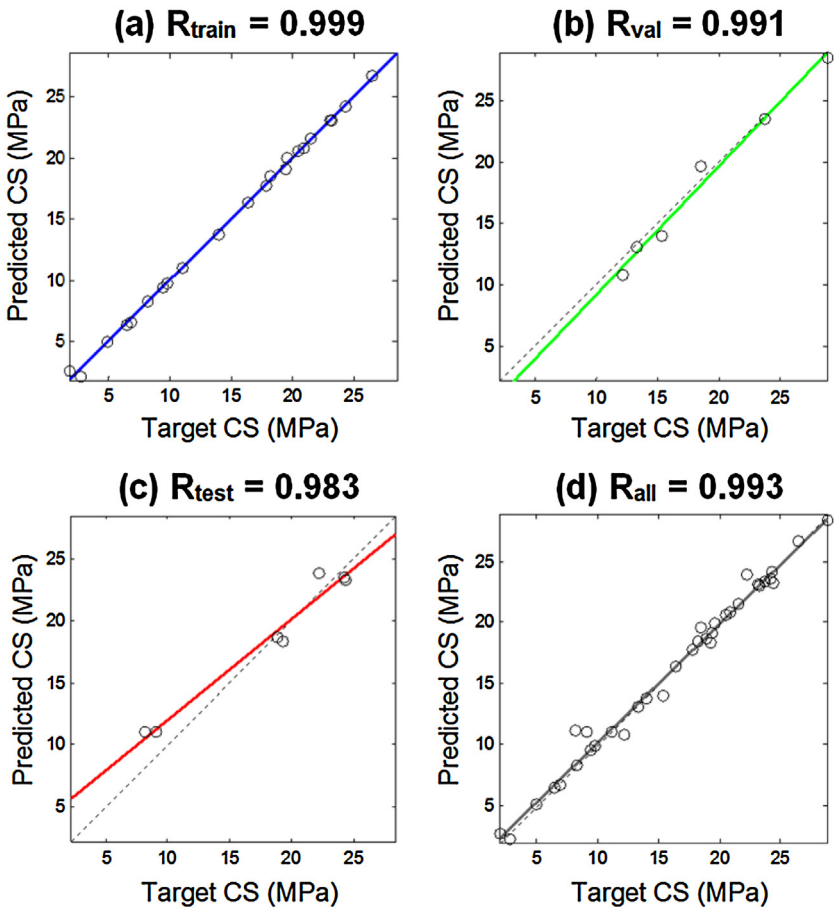

Fig. 8. Linear regression between the target and predicted compressive strength (in MPa) achieved by the bone neural network. Values were reported individually for a) training (blue), b) validation (green) and c) testing (red), and d) the entire dataset (black). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where no additional test samples were available. A large-scale study involving 20,000 NNs confirmed that NNs trained on real bone data significantly outperform the NNs trained on surrogate data.

The framework has been evaluated via a comparative study that predicted concrete CS using both large (1030 samples) and small (56 samples) datasets. Using the proposed framework it was possible to develop a small-dataset NN with performance $R_{all} = 0.923$ comparable with that of a large-dataset NN ($R_{all} = 0.944$). This demonstrates that a drastic 18 times reduction in the required dataset size corresponds to only a small decrease in accuracy of 2.12% – a compromise to be considered in single-center studies where datasets are often limited.

When applied to 35 osteoarthritic specimens, our methodology yielded a reliable predictive NN tool for non-destructive estimation of bone compressive strength. The optimised NN achieved a high generalising accuracy of 98.3%. Additionally, by quantifying random effects specific to the dataset, the surrogate data approach allowed us to define a performance threshold of $R_{sur,max} = 0.87$ for successful NNs. The successful application of the proposed methodology confirms that the size of datasets does not necessarily limit the utility of NNs in the medical domain.

5.2. Practical significance of the bone CS model

In cellular solids, CS is an exponential function of the apparent density, BV/TV, raised to the power of 3/25 [15,25,47]. Although such an exact relationship has not been established specifically for osteoarthritic trabecular tissue, this power model, with a bivariate regression coefficient $R_{powermodel} = 0.906$ is the best existing fit to the data [17]. The generalising NN performance $R_{test} = 0.983$ achieved in our study exceeded $R_{powermodel}$ by 8.5%. The proposed NN model yields substantially more accurate predictions by considering variable interrelations within multi-dimensional medical datasets and successfully capturing the complex physiological phenomena in patients suffering from severe OA.

The high accuracy of the proposed CS model enables prediction of bone fracture risk based on the structural and physiological parameters that can be derived without invasive tests on the patient. Hence, by predicting how CS correlates with the bone volume fraction, trabecular thickness and structure model index for patients of various age and gender groups, the NN model can provide a decision support tool for hard tissue engineers and clinicians alike [26]. To our best knowledge, the NN presented in this work is the only existing patient-specific model for prediction of CS in trabecular bone affected by OA.

The potential practical applications include: the estimation of bone fracture risk in osteoarthritic patients from CT-scans and basic physiological data, load modelling of synthetic bioscaffolds that mimic natural trabecular bone damaged by OA, and the tailoring of bioscaffold designs for an individual patient to match the damaged trabecular tissue at the site of implantation.

The predictive NN model can be adapted to larger datasets and to other degenerative bone disorders, such as osteoporosis and metastatic cancer, with marginal increase in design effort and cost [8,9]. Such scalability is inherent in the underlying ML algorithms,

which enable NNs to learn and improve their performance with new data [10,14,48,49].

Acknowledgement

This work has been supported by EPSRC UK (EP/K02504X/1).

Appendices

Appendix A. Trabecular bone data: real vs surrogate samples

See Tables A1 and A2

Table A1
Real bone data.

Sample no.	SMI	tb.th	BV/TV	Age (years)	Gender (F = 1)	CS (MPa)
1	0.06	243	32.5	41.8	1	20.9
2	1.42	224	21.5	52.0	1	6.91
3	0.48	239	26.6	57.0	1	18.2
4	−0.82	212	43.5	63.9	1	9.46
5	1.22	419	17.9	64.0	1	23.1
6	0.64	223	27.6	67.1	1	19.4
7	2.10	197	9.82	68.1	1	2.76
8	0.38	367	26.9	71.5	1	18.9
9	0.80	218	15.4	74.9	1	6.49
10	0.54	314	25.0	76.0	1	17.8
11	0.30	326	32.4	87.0	1	24.2
12	−0.17	287	30.4	41.7	0	21.5
13	−0.31	284	37.0	47.9	0	16.4
14	0.04	265	38.7	49.8	0	11.1
15	0.82	241	22.7	49.8	0	26.5
16	−0.23	303	37.6	65.8	0	28.8
17	1.77	219	25.3	68.0	0	4.91
18	1.33	261	17.4	72.9	0	9.81
19	0.04	307	29.7	73.9	0	23.7
20	0.36	271	31.6	81.8	0	24.4
21	0.31	252	33.8	60.9	1	20.5
22	0.70	283	22.5	62.9	1	12.2
23	1.59	247	13.7	72.6	1	1.93
24	0.45	257	27.4	45.7	0	19.6
25	0.44	266	27.5	62.9	0	18.5
26	0.15	270	32.1	77.8	0	22.2
27	1.08	193	19.4	87.0	0	9.12
28	1.93	154	9.68	49.0	1	8.22
29	0.92	263	25.3	66.0	1	15.4
30	−0.43	299	39.7	69.9	1	23.2
31	1.04	239	21.0	73.9	1	8.15
32	−0.05	288	35.6	46.8	0	24.3
33	0.39	246	26.6	64.9	0	19.3
34	0.71	178	12.2	68.0	0	14.0
35	0.70	234	21.8	84.9	0	13.3

Bone data were extracted from the original study from [17] using a Plot Digitiser tool.

Table A2
Surrogates.

Sample no.	SMI	tb.th	BV/TV	Age (years)	Gender (F = 1)	CS (MPa)
1	1.00	260	32.3	66.8	1	17.43
2	0.58	217	38.0	54.0	0	16.21
3	0.73	260	40.5	82.7	1	6.95
4	0.13	209	19.3	57.4	0	19.89
5	0.53	185	17.6	80.4	1	28.51
6	1.72	314	30.5	55.9	0	13.48
7	0.67	269	16.0	60.9	1	26.99
8	0.63	336	26.8	49.6	1	13.33
9	0.12	287	26.9	68.9	0	23.77
10	0.58	271	35.0	54.2	0	15.24
11	0.80	306	26.0	46.6	1	14.52
12	0.90	320	24.1	71.6	1	10.76
13	0.42	376	29.9	60.1	0	8.40
14	0.37	155	31.5	69.6	1	12.63
15	1.93	317	26.7	61.3	0	20.02
16	−0.49	275	23.1	68.5	1	21.19
17	1.38	378	18.0	44.5	1	19.40
18	1.47	264	28.1	79.5	0	2.93
19	1.14	258	21.1	74.4	1	22.59
20	−0.23	304	13.5	72.4	1	24.92
21	0.18	224	31.9	74.9	1	20.93
22	0.32	261	20.6	61.2	1	5.17
23	0.90	326	25.1	68.2	1	13.90
24	−0.11	270	30.3	66.9	0	19.60
25	0.98	312	23.3	65.6	0	20.09
26	−0.20	293	31.4	57.8	0	10.90
27	0.86	272	24.1	56.8	1	11.85
28	0.59	227	30.2	63.3	1	19.02
29	1.10	283	30.6	56.1	1	15.62
30	0.97	194	25.1	74.6	0	18.13
31	1.44	277	11.7	85.3	1	11.17
32	1.39	282	22.7	45.3	0	9.80
33	0.32	292	24.7	65.8	0	22.25
34	0.94	323	21.3	49.6	0	20.85
35	0.04	367	19.4	74.5	1	25.36

Surrogate data were synthesised as a random normal distribution with the mean and standard deviation of the real bone data within the same range.

Appendix B. NN design parameter estimation for bone CS data

Effects of the number of neurons in hidden layer

Limited availability of the training samples necessitates careful selection of the size of the hidden layer in order to achieve well-generalising NNs. The effect of increasing number of neurons in the hidden layer from 1 to 13 was investigated in the series of experiments that involved 10 runs of 2000 NNs for each neuron, i.e. 260,000 NNs in total were analysed for enhanced repeatability.

Reported in Fig. A1 is the number of statistically significant NNs, i.e. NNs that exhibited performance of $R_{all} \geq 0.6$ across the entire dataset, as well as individually for the training, validation and test datasets. Despite the inter-run volatility in the results, on average the highest performing NNs had 2, 3, 4, and 5 neurons in hidden layer with 890, 878, 873 and 851 statistically significant NNs per run, respectively.

For statistically significant NNs the distributions of R_{all} and R_{val} were compared for various neuron configurations. The highest R_{val} was achieved in NN designs with 3 and 4 neurons. The Wilkinson rank sum test was used to assess the inter-run volatility for the two candidate designs. Based on comparison of the 50 pairwise p-values at 5% confidence level, NNs with 4 neurons were established to be more stable than those with 3 neurons. Following careful evaluation of the largest number of statistically significant NNs produced, the highest R_{all} and R_{val} performance, and adequate inter-run stability, NN with 4 neurons in a hidden layer was chosen as the final NN design for the next stage in parameter estimation.

Another way to identify optimal NN size is by integrating a parameter regularisation into a training process. A weight decay procedure penalises large weights forcing the NN parameters to shrink. Larger networks have more parameters to start with, but regularisation prevents some of this 'excessive capacity' from being trained unnecessarily. The effective number of parameters in a NN trained with regularisation can serve as an indication of how well the NN utilises its capacity. We investigated the number of effective parameters for NNs of varying hidden layer size (from 1 to 20 neurons) trained by Bayesian regularisation backpropagation (Fig. A2). The number of effective parameters rose in NN configurations with 1–4 neurons and fell in configurations with 5 neurons and above, indicating that the NN with 4 neurons was most effective. This

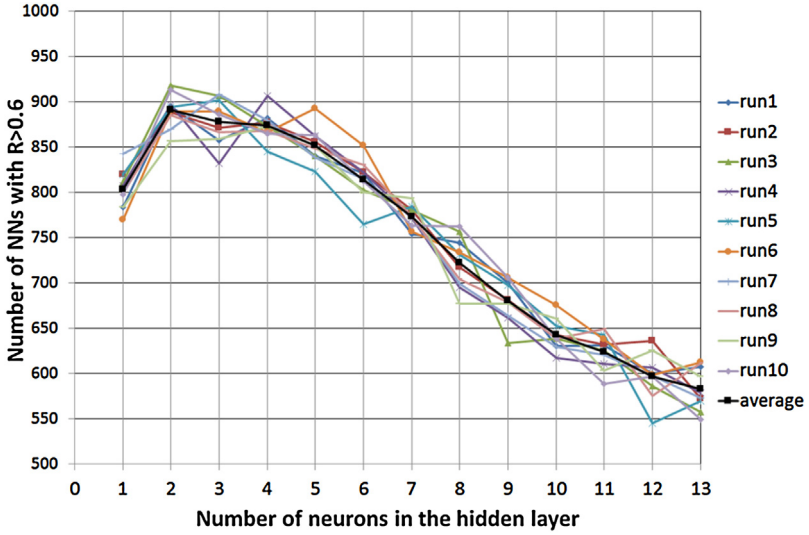

Fig. A1. Number of statistically significant NNs per run for various number of neurons in the hidden layer.

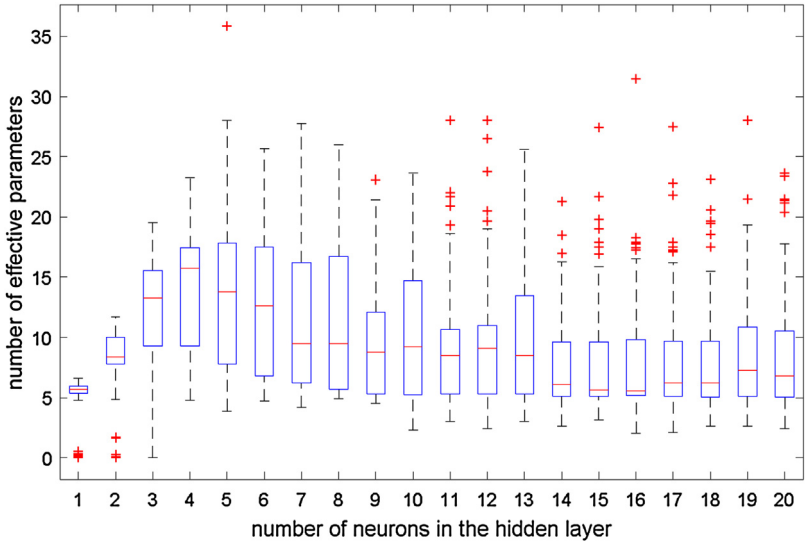

Fig. A2. Distributions of the effective number of parameters in regularised neural networks for various number of neurons in the hidden layer.

was further confirmed by considering validation performance R_{val} across 20 runs, which was highest for the NNs with 4 neurons.

Effects of the training duration

Training duration stipulates the balance between the NN training performance and generalisation. Although extended training can lead to exceptional performance on the training dataset, it often results in poor generalisation on the test data that the NNs had not seen before. Early stopping helps to avoid NN over-fitting upon reaching the maximum number of validation checks. The number, n, of consecutive validation iterations during which the NN performance fails to decrease plays key role in controlling the quality of NN training. It also affects computational efficiency of the training algorithm, which deteriorates with the increasing n.

When investigated on 20 runs of 2000 NNs, corresponding to n from 1 to 10 in the increments of 1 and 10–100 in the increments of 10, the effect of$n$ on the NN performance was marginal. No statistical difference was established between the distributions of R (neither R_{val} nor R_{all}) for various n in any possible pair of Wilkinson rank sum comparisons at 5% significance level. Thus, any configuration that yielded the highest values of R_{all} and $R_{val}$was a suitable candidate for the final NN. Based on the above considerations, the n value of 9 allowed for maximum performance across all samples while maintaining adequate simulation efficiency.

Appendix C. Values of weights and biases of the final NN model for trabecular bone data

The small-dataset bone CS NN was trained using the Levenberg Marquardt backpropagation algorithm [37]. During each iteration (epoch), the performance of the NN on training, validation and test samples was monitored in terms of its cost function expressed by *MSE*.

Fig. A3 shows how the NN error on the training set was monotonically decreasing with each epoch. The errors on the validation and test samples were sporadic until the 14th epoch. At the 31st epoch the validation error failed to decrease for 9 consecutive iterations and the early stopping criterion was reached. The weights and biases were then reverted by 9 epochs to the state at which the validation error was least, i.e. the final state of the trained NN weights

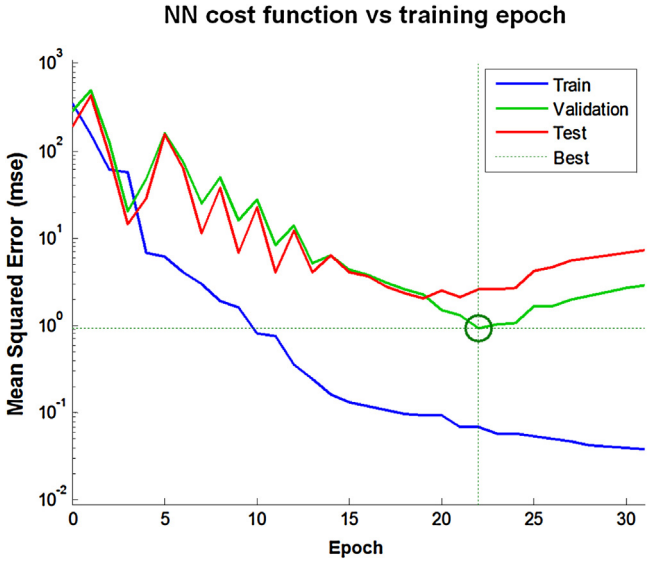

Fig. A3. Neural network cost function dynamics during the 30 epoch of training (blue), validation (green) and testing (red). Upon reaching the 9th validation check at 22nd epoch (green circle), the neural network training process was completed.

Table A3
Weights and biases.

IW	0.887	2.382	−0.888	−3.584
	1.301	−1.586	0.904	−3.841
	−3.268	0.632	−1.342	−0.144
	−1.216	−2.153	−1.380	−3.000
	−0.620	1.592	−0.379	−1.169
$\overline{lw'}$	−0.698			
	−0.151			
	2.349			
	−1.501			
$\overline{b^{(1)}}$	0.268	−0.006	−1.224	−4.972
$b^{(2)}$	0.623			

and biases corresponded to the 22nd epoch. Notably, this is not the state that minimises cost function for the *test* samples, as these independent test samples were not involved in the model training; their corresponding cost function is provided for illustrative purposes.

Table A3 shows the final weight and bias parameters for the trained bone NN: the input weights matrix IW, the layer weights column vector $\overline{lw'}$, and the corresponding biases $\overline{b^{(1)}}$ and $b^{(2)}$.

References

[1] Campbell C. Machine learning methodology in bioinformatics. In: Kasabov N, editor. Springer handbook of bio-/neuroinformatics. Berlin, Heidelberg: Springer; 2014. p. 185–206.
[2] Forman G, Cohen I. Learning from little: comparison of classifiers given little training. Proc. PKDD 2004;19:161–72.
[3] Inza B, Armañanzas E, Larrañaga P, Lozano J. Machine learning: an indispensable tool in bioinformatics. In: Matthiesen R, editor. Bioinformatics methods in clinical research, vol. 593. 2010: Humana Press; 2017. p. 25–48.
[4] Johnson JL. Probability and Statistics for Computer Science. New York: Whiley; 2011.
[5] Woolson RF, Clarke WR. Statistical methods for the analysis of biomedical data. 2nd ed. New York: Wiley-Interscience; 2002.
[6] Haykin S. Neural networks: a comprehensive foundation. 2nd ed. Prentice Hall; 1999.
[7] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2:359–66.
[8] Amato F, López A, Peña-Méndez EM, Vaňhara P, Hampl A, Havel J. Artificial neural networks in medical diagnosis. J Appl Biomed 2013;11:47–58.
[9] Hudson DL, Cohen ME. Neural networks and artificial intelligence for biomedical engineering. New York: IEEE; 2000.
[10] Grossi E. In: Suzuki K, editor. Artificial neural networks and predictive medicine: a revolutionary paradigm shift. InTech; 2011. p. 139–50.
[11] Khovanova NA, Mallick KK, Shaikhina T. Neural networks for analysis of trabecular bone in osteoarthritis. Bioinspir Biomim Nanobiomater 2015;4(no. 1):90–100.
[12] LeBaron B, Weigend AS. A bootstrap evaluation of the effect of data splitting on financial time series. IEEE Trans Neural Networks 1998;9:213–20.
[13] Bowden GJ. Optimal division of data for neural network models in water resources applications. Water Resour Res 2002;38:1–11.
[14] Wasserman PD. Neural computing: theory and practice. New York: Van Nostrand-Reinhold; 1989.
[15] Cunningham P, Carney J, Jacob S. Stability problems with artificial neural networks and the ensemble solution. Artif Intell Med 2000;20(no. 3):217–25.
[16] Shaikhina T, Khovanova N, Mallick K. Artificial neural networks in hard tissue engineering: another look at age-dependence of trabecular bone properties in osteoarthritis. In: IEEE EMBS Int. Conf. Biomed. Heal. Informatics. 2014. p. 622–5.
[17] Perilli E, Baleani M, Ohman C, Baruffaldi F, Viceconti M. Structural parameters and mechanical strength of cancellous bone in the femoral head in osteoarthritis do not depend on age. Bone 2007;41:760–8.
[18] Sinusas K. Osteoarthritis: diagnosis and treatment. Am Fam Physician 2012;1(86):49–56.
[19] Stewart A, Black AJ. Bone mineral density in osteoarthritis. Curr Opin Rheumatol 2000;12(no. 5):464–7.
[20] Živković V, Stamenković B, Nedović J. Bone mineral density in osteoarthritis. Acta Fac Med Naiss 2010;27(3):135–41.
[21] Chan MY, Center JR, Eisman JA, Nguyen TV. Bone mineral density and association of osteoarthritis with fracture risk. Osteoarthritis Cartilage 2014;22(9):1251–8.
[22] Bessho M, Ohnishi I, Okazaki H, Sato W, Kominami H, Matsunaga S, et al. Prediction of the strength and fracture location of the femoral neck by

CT-based finite-element method: a preliminary study on patients with hip fracture. J Orthop Sci 2004;9:545–50.
[23] Keyak JH, Rossi SA, Jones KA, Skinner HB. Prediction of femoral fracture load using automated finite element modeling. J Biomech 1997;31:125–33.
[24] Carter DR, Hayes WC. Bone compressive strength: the influence of density and strain rate. Science 1976;194:1174–6.
[25] Helgason B, Perilli E, Schileo E, Taddei F, Brynjolfsson S, Viceconti M. Mathematical relationships between bone density and mechanical properties: a literature review. Clin Biomech 2008;23:135–46.
[26] Geris L. Computational modeling in tissue engineering. Berlin: Springer-Verlag; 2013.
[27] Sinusas K. Osteoarthritis: diagnosis and treatment. Am Fam Phys 2012;85(no. 1):49–56.
[28] Hirata Y, Katori Y, Shimokawa H, Suzuki H, Blenkinsop TA, Lang EJ, et al. Testing a neural coding hypothesis using surrogate data. J Neurosci Methods 2008;172:312–22.
[29] Schreiber T, Schmitz A. Improved surrogate data for nonlinearity tests. Phys Rev Lett 1996;77(no. 4):635–8.
[30] Theiler J, Eubank S, Longtin A, Galdrikian B, Doyne Farmer J. Testing for nonlinearity in time series: the method of surrogate data. Physica D 1992;58(no. 1–4):77–94.
[31] Yeh I-C. Modeling of strength of high-performance concrete using artificial neural networks. Cem Concr Res 1998;28(no. 12):1797–808.
[32] Yeh I-C. UCI machine learning repository: concrete compressive strength data set. Machine Learning Repository, University of California Irvine, Center of Machine Learning and Intelligent Systems; 2007. Available: https://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength (Accessed 14 January 2015).
[33] Yonaba H, Anctil F, Fortin V. Comparing sigmoid transfer functions for neural network. J Hydrol Eng 2010;15(no. 4):275–83.
[34] Nguyen D, Widrow B. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. IEEE Int. Jt. Conf. Neural Networks, 3. 1990. p. 21–6.
[35] Levenberg K. A method for the solution of certain non-linear problems in least-squares. Q Appl Math 1944;2:164–8.
[36] Marquardt DW. An algorithm for least-Squares estimation of nonlinear parameters. J Soc Ind Appl Math 1963;11:431–41.
[37] More JJ. The Levenberg-Marquardt algorithm: implementation and theory. Lecture Notes Mat 1978;630:105–16.
[38] Fushiki T. Estimation of prediction error by using K-fold cross-validation. Stat Comput 2009;21:137–46.
[39] Timmer J. Power of surrogate data testing with respect to nonstationarity. Phys Rev E 1998;58(October (no. 4)):5153–6.
[40] Li D-C, Wu C-S, Tsai T-I, Lina Y-S. Using mega-trend-diffusion and artificial samples in small data set learning for early flexible manufacturing system scheduling knowledge. Comput Oper Res 2007;34(no. 4):966–82.
[41] Gomez I, Cannas SA, Osenda O, Jerez JM, Franco L. The generalization complexity measure for continuous input data. Sci World J 2014;2014(no. 815156):9.
[42] Zhang S, Liu H-X, Gao D-T, Wang W. Surveying the methods of improving ANN generalization capability. Proc. 2003 Int. Conf. Mach. Learn. Cybern., 2. 2003. p. 1259–63.
[43] Opitz D, Maclin R. Popular ensemble methods: an empirical study. J Artif Intell Res 1999;11:169–98.
[44] Dietterich TG. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. Mach Learn 2000;40(no. 2):139–57.
[45] Ahmad Z, Zhang J. A comparison of different methods for combining multiple neural networks models. Proc. of the 2002 Int. Joint Conf. Neural Networks, 1. 2002. p. 828–33.
[46] Hollander M, Wolfe DA. Nonparametric statistical methods, vol. 2, 2nd ed. New York: Wiley; 1999.
[47] Gibson LJ, Ashby MF, Harley BA. Cellular materials in nature and medicine. Cambridge: University Press; 2010.
[48] Eller-Vainicher C, Zhukouskaya VV, Tolkachev YV, Koritko SS, Cairoli E, Grossi E, et al. Low bone mineral density and its predictors in type 1 diabetic patients evaluated by the classic statistics and artificial neural network analysis. Diabetes Care 2011;34(October (10)):2186–91.
[49] Peteiro-Barral D, Bolon-Canedo V, Alonso-Betanzos A, Guijarro-Berdinas B, Sanchez-Marono N. Toward the scalability of neural networks through feature selection. Expert Syst Appl 2013;40:2807–16.