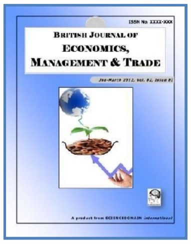


British Journal of Economics, Management & Trade
4(2): 228-263, 2014

SCIENCEDOMAIN *international*
www.sciencedomain.org

Technoeconomic Projections with Artificial Neural Networks using an Ensemble of Sparsely-Sampled Bootstrapped Data

Mohammad Dabbas[1], Perambur S. Neelakanta[2*] and Dolores De Groff[3]

[1]Department of Engineering, Broward College, Coconut Creek, Florida 33066, USA.
[2]Department of Computer and Electrical Engineering and Computer Science College of Engineering and Computer Science Florida Atlantic University Boca Raton, Florida 33431, USA.
[3]Civil, Environmental and Geomatics Engineering College of Engineering and Computer Science Florida Atlantic University Boca Raton, Florida 33431, USA.

Authors' contributions

This work was carried out in collaboration between all authors. Author PSN formulated the problem and wrote the first draft of the manuscript. Author MD developed the analytical methods and performed the computations. Author DDG managed the neural network analyses. All authors read and approved the final manuscript.

Original Research Article

Received 29th March 2013
Accepted 25th June 2013
Published 8th November 2013

ABSTRACT

Aims: The present study refers to developing an artificial neural network (ANN) that can be designed exclusively for *ex ante* forecasting in technoeconomic contexts using an ensemble set of sparse and insufficient sampled-data availed *ex post*.
Study Design: In general, the samples in a data set of technoeconomic structures would largely be limited in number due to sparse-sampling; also, availability of number of such sets is mostly inadequate for robust training of an ANN so as to obtain realistic inferences subsequently in the prediction phase. Hence, a sparsity-recovery strategy is advocated *via* a cardinality enhancement procedure (through Nyquist sampling) performed on the sparse data set in order to augment the number of samples in its sampled-data space. Further, the concept of statistical bootstrapping technique of resampling is invoked and applied on the cardinality-improved subset so as to obtain an enhanced number of data sets. This ensemble of data set is then adopted to facilitate robust training of the test

**Corresponding author: Email: neelakan@fau.edu;*

ANN.
Place and Duration of Study: The studies were conducted (2012-2013) at: Department of Computer and Electrical Engineering and Computer Science, College of Engineering & Computer Science, Florida Atlantic University, Boca Raton, Florida 33431, USA.
Methodology: The study governs technoeconomic *ex ante* projections pertinent to a wind-power generation business complex elucidated *via* ANN-based forecasting. Relevant test ANN is designed to accommodate training with an ensemble of sampled set available *ex post* but, in limited numbers. The associated scarcity is recovered by artificially enhancing the data space to an adequate extent *via* Nyquist sampling and bootstrapping techniques. Further, the test ANN designed corresponds to a multilayer perceptron (MLP) supporting backpropagation of the perceived error at the output with respect to a supervisory value. It accommodates the bootstrapped data space at its input relevant to technoeconomic details on a practical wind-power system performance reported in the literature. The training and prediction exercises on the test ANN corresponds to optimally elucidating output predictions in the context of the technoeconomics framework of the power generation considered.
Results: Using the test ANN trained with bootstrap-enhanced, scarcity-recovered sparse data on wind-power generation statistics and associated plant economics, reliable inference (in the prediction phase) is achieved on the system performance. That is, the ANN output obtained depicts forecast projections on the productivity of electric power generation in the *ex ante* regime. Simulation studies thereof and results obtained demonstrate the efficacy of the method proposed, bootstrapping algorithm developed and the use of MLP in the technoeconomic contexts.

Keywords: Technoeconomic forecasting; artificial neural network; sampled-data; bootstrapping; scarcity recovery; wind-turbine power generation.

1. INTRODUCTION

In economics, the observed (*ex post*) outcomes invariably exhibit rise and fall in their values due to various exogenous and endogenous causal factors. To assess the corresponding trend in economy across a projected *ex ante* domain, an analysis can be performed to identify the components that go through changes in the observed *ex post* values and how such changes lead to discernable patterns along the *ex ante* frame. In a general sense, relevant forecasting implies a strategy of prediction on a “statement concerning unknown, in particular future, events”; and specific to economics-related contexts forecasting refers to the “best estimate” of futuristic projections of an associated entity with reference to its growth or decline as a function of time [1,2]. That is, forecasting in economics implies predictions on futuristic details on growth or decay of a dependent variable as a function of time. Further, *in lieu* of seeking a projection in the temporal framework, forecasting may also be done as regard to any outcome as a function of the associated causative inputs. That is, forecasting entails in general, the task of quantitatively estimating the details about the likelihood of future events (or unknown outcomes/effects) based on past and current information on the observed events (or known causal factors).

In practice, the temporal forecast on an economics variable is done using a set of observed *ex post* data *versus* time *via* regression analysis and a trend curve of the outcome is determined as a function of time. This trend-line is then projected to forecast an estimate of possible values of the dependent variable at a specified futuristic value of time. That is, forecast estimate is obtained using extrapolation of the trend values beyond the range over which the regression is performed. However, such extrapolations based on classical strategies of simple curve-fitting may not lead to realistic forecasting because relevant efforts may often ignore the uncertainty of underlying realism pertinent to various factors that cohesively enable each data point (dependent variable) to acquire that value at a given instant of time. As a consequence, the conjectural or stochastical considerations inherent in the evolution of the entity under growth (or decline) are not *per se* carried forward into the forecast regime while envisaging a simple mathematical regression analysis. Consequently, the forecasts made may end up as unrealistic 'hockey-stick' projections [3]. Nevertheless, such pursuits are not uncommon in forecasting an economics outcome but often leading to over- and/or under-estimations. Therefore, methods are sought in practice to conceive forecasting methods in economics (as well as in other areas) that are more reliable.

In particular, considering technology-specific economics, robust forecasting is essential for near-futuristic reliable pursuits of practical, strategic engineering operations and exercises on managerial decisions using the tactical technoeconomic details in hand. Here, the term "technoeconomics" offers explicitly insight into the synergism of technological considerations in the perspectives of macro-, meso- and microscopic economics theory [1]. Relevant studies involve multi-disciplinary considerations, underpinned by a portfolio of specialized quantitative research tools like econometrics, mathematical programming, data-mining, statistical analysis, cost-benefit analysis, forecasting methods, etc. For a techno-economist, relevant econometric strategies denote the part of engineering economics applied to a specific technology being addressed; and a judicious blend of principles of economics and econometric pursuits is necessary in real-world technoeconomic efforts toward robust forecasting and reliable decision-making strategies for imminent applications.

The first level of relevant approach toward technoeconomic forecasting includes developing an evolutionary model on the associated independent *versus* dependent variables and then the model is analyzed *via* a chosen econometric approach. Such evolutionary models when ascribed to business structures of technoeconomic enterprises (like electric power utility or telecommunication service industry), they portray the growth and futuristic welfare of the industry in question. Further, relevant forecasting *vis-à-vis* growth dynamics conforms to robustly predicting the performance of the underlying infrastructure technology as well as the prevailing (competitive) market profile pertinent to the products supplied and service options rendered to the consumers. As such, in the realm of technoeconomics, the growth (and hence the forecasting) considerations of the underlying complex business structure include both market *versus* time and technology *versus* performance projected in the *ex ante* regime consistent with the associated causal factors. The temporal market growth indicates the revenue prospects and the projected technology-specific performance details are needed for prudent maintenance and operation schedules in the near- and far-sighted future. Further, in all such technical and managerial processes, decision-making is imperative and any decision being made thereof has an element of looking forward into a state of 'yet-to-be-explored' scenario pertinent to market status and engineering concerns. In both cases, the heuristics of required decisions needs logistics of forecasting reliable for use.

The present study is concerned with forecasting on the technology-related performance details *versus* causal factors. That is, in the context of technology-centric business, an exclusive scheme of forecasting efforts is needed on the system performance in implementing an adaption framework that robustly sustains the engineering operations on tactical and strategic basis. For example, in order to implement reliable maintenance and enhanced production schedules, forecast-benchmarking models [4] can be adopted in the technoeconomic framework of an engineering business (such as electric power generation and distribution service industry). Such models concern with performance variables *versus* causative factors involved. Hence, the forecasting strategies pursued rely on a set of sample observations on a dependent (outcome) variable *versus* corresponding set of independent causal entities. Such observations recorded and made available in the database are presumably known with certainty. That is, in the relevant *ex post* domain, the outcomes (dependent variable) *versus* the independent causal variable are mostly deterministic and can be checked against the existing data. Corresponding forecast exercises predict the values of the dependent variables beyond the *ex post* domain falling in the *ex ante* regime.

The efficacy of forecasting in technoeconomic exercises can be estimated in terms of the reliability of the outcome projected by the (forecasting) method pursued; and in reality, such methods are far from being simple since the business in hand itself could be a complex system. Traditionally as stated earlier, the regression coefficients of the trend curves of the dependent variables are deduced *ex post* and interpreted as a function of independent causal variables in the *ex ante* domain. However, such pursuits towards forecasting may suffer from errors, especially if the overall statistical data used is small in size rendering the included variables being insufficient to portray adequately the entropy involved. Further, the accuracy of long-term forecast is largely influenced by paradigm shifts in managerial visions and the forecasts become susceptible to errors in truly tracking the associated unpredictable states. That is, the technoeconomic data profile in general, may have a complex structure with mostly nondeterministic attributes that affect robust forecasting, (if the data profile exhibits intense non-stationarity or when it falls into chaotic states). Therefore, the progress (growth/decay) of the dependent variable decided *via* forecasting becomes limited by the restricted (short-term) details available on the track of the *ex post* chain; and the *ex post* data inherently involve chances or probability considerations that directly lead to a need for describing and dealing with the uncertainty associated with the values of the data set being processed and used while forecasting.

Characterizing and dealing with such uncertainty in forecast modeling is not, simple. But, since uncertainty is present implicitly in all decision-theoretic reasoning, at least an ‘expected value that includes the uncertainty’ associated with the data set should be a part of any forecasting exercise [5]. In all, considering various constraining variables on forecasting being partially deterministic and mostly stochastic, the technoeconomic forecasts cannot *per se* be made just by using any simple analytical methods based on stochastic simulations.

In such cases forecast projections can however, be attempted *via* a ‘black-box approach’ using artificial neural networks (ANN) trained with ex post details containing the diverse and uncertain variables of the technoeconomic infrastructure. Hence, rational intuitions on forecast projections can be derived in the *ex ante* regime as a prediction phase effort using the trained ANN in question. Yet, such *ex ante* forecasts could still be susceptible to errors in truly or exactly tracking the unpredictable states of the *ex post* regime into the regime of forecast. As such, any forecast done (either *via* ANN or otherwise) could be valid within a progressive error-bar set between an upper- and lower-bound [3].

In the present study, the ANN-based approach pursued is more appropriate in technoeconomic contexts, since the ANN can be trained to assimilate the *ex post* vagaries in economics and performance shifts in engineering variables. The trained ANN then has a pattern of input-output relations mapped in it reflecting the stochastics of the *ex post* contexts. When a new data set is addressed at its input, it can classify it robustly *vis-à-vis* the trained pattern stored in it. In short, by training an ANN with an ensemble of data known *a priori* and using the trained network, an output can be sought in a prediction phase when subjected to an input of unknown data set. Relevant (converged) output observed is then analyzed to infer forecast details on *a posteriori* basis pertinent to the data space [6-19].

The underlying efforts conform to a best-effort strategy so that the outcomes projected and forecast made are rendered fairly robust and reliable consistent with observed details ex ante. Such assurance is however feasible, only if the *ex post* data adopted and applied on the models or tools are available to an adequate extent so that the stochastical features of the epochs of the past are projected effectively into the *ex ante* domain and tracked reliably. However, it is not uncommon in practice that, mostly available prior data for use in the aforesaid forecasting efforts on business performance evaluations are limited in number and are sparsely indicated with details incompletely described. Such insufficient and scarce set of information would however, hamper making realistic and robust forecast inferences say, for example using the ANN strategy or otherwise.

Therefore, deducing forecasts even with inadequately captured *ex post* data is sought and emphasized in the present study. A relevant suite is proposed thereof to apply models and tools that conform to an ANN-based approach. Hence, by duly considering the inadequacy or scarcity of available (sampled) data, the cardinality of data space is first (artificially) enhanced *via* Nyquist sampling and Whittaker-Kotelnik-Shannon (WKS) heuristics of interpolation theory; and, statistical resampling *via* bootstrapping is then exercised on the cardinality-improved data set to obtain its multiple pseudo replicates. The ensemble of such surrogate sets is used for training the test ANN multiple times with scarcity-recovered details (availed through cardinality-improvement and post-bootstrapping exercises). Thus the ANN is rendered thereof to yield a prediction performance on the output to a reliable extent.

Thus, the scope of the present study is to address the notions of technoeconomic forecasting using the proposed method using ANN. Relevantly, the consequences of applying inadequate ensemble of sample data as the ANN input (during training phase) is considered; and, the need to enhance the sampled data space, that is, exercising a scarcity recovery procedure so as to improve the ANN performance toward robust output predictions is indicated. Hence, a cardinality-enhancement scheme in a sampled subset (*via* Nyquist sampling principle and WKS interpolation theoretics) as well as statistical resampling to obtain multiple surrogates of the cardinality-enhanced subset (using bootstrapping algorithm) are developed *vis-à-vis* scarcity removal in the sampled-data space availed *ex post* pertinent to typical technoeconomic contexts. The test ANN trained with sparsity recovered data is then used for subsequent prediction exercise to obtain meaningful forecasts in the *ex ante* regime. The test ANN designed is based on a multilayer perceptron (MLP) architecture using backpropagation (BP) of the error schedule [6].

The (limited) sampled-data set considered here refers to a practical technoeconomic context of wind-turbine based power generation complex [20-22]; however, without any loss of generality, the methodology outlined can assess the technoeconomic performance of any similar business structures. Performance prediction here implies elucidating optimally the underlying futuristic predictions (forecast projections) on the anticipated productivity across the *ex ante* regime. Simulation studies presented here are illustrative of practical implementation of the projected objective and hence, the studies performed are presented in sections as listed below:

(i) In Section 2, the technoeconomic aspects of robustly assessing the futuristic prospects of the underlying business are outlined and the need for a reliable forecasting technique is indicated

(ii) Suggested in Section 3 is a method toward such forecasting efforts *via* an ANN; and the required architectural aspect of such an ANN is studied *vis-à-vis* the *pros* and *cons* of applying inadequate ensemble of data exercised as ANN input during the training phase. Corresponding reliability issues on the outcome in the prediction phase of the ANN operation are also analyzed

(iii) The question of statistically enhancing the (inadequate) sampled-data space using (a) the technique of cardinality enhancement in a sampled data space and (b) obtaining multiple pseudoreplicates of the cardinality improved subset *via* bootstrapping are described in Section 4; and relevant computational scheme on scarcity removal is outlined using a pseudocode

(iv) An exemplary technoeconomic framework is identified in Section 5 where in a limited extent of sampled data set is gathered pertinent to a practical context of wind-power generation complex [20]. Relevant sampled-data acquired is concerned with the information gathered from an ensemble of power generation *versus* wind-speed/direction details over a period of time-frame. This data set is designated as the *ex post* regime pertinent to a certain wind-speed and direction conditions

(v) Designing an appropriate, bootstrapped-data based test ANN (abbreviated as BSD-ANN) for forecast purposes [6-19] is described in Section 6

(vi) The training schedule and prediction performance of the BSD-ANN are addressed in Section 7. That is, the performance of the trained ANN in yielding output that conforms to *ex ante* predictions (and forecasts) on the technoeconomics of power generation *versus* wind-speed/direction parameters is evaluated in the prediction phase. Relevant simulation experiments are described and the results are presented with appropriate discussions

(vii) Lastly, in Section 8, as a closure the efforts addressed in this study are summarized.

2. TECHNOECONOMIC BUSINESS AMBIENT AND RELIABLE FORECASTING

As indicated before, forecasting refers to predicting a statement concerning an unknown, futuristic disposition of a variable or event. In technoeconomic contexts it may refer to the "best estimate" of futuristic projections of an entity with reference to its growth or decline as a function of time [2,3]. Alternatively, technoeconomic forecasting may also depict performance projections or forecast values of engineering details *versus* a new set of causative variables. The dogmatic aspect of forecasting in depicting the unknowns in future events on the basis of known history of the past learned requires, (i) a comprehensive data set concerning the past (that is, the *ex post* details); (ii) ability of the forecasting tool to "learn" the features of the *ex post* epochs and (iii) the forecast engine judiciously track these features reliably at least within a pair of upper- and lower-bounds [3].

Making of such a forecast inference engine is the motivated inquest of this study, which proposes the ANN pursuit as a compatible strategy as described below. Specifically, the ANN method described here allows inadequate sampled-data space of the past in the pursuit of futuristic forecast details; and, as mentioned before, the data suite considered is relevant to a technoeconomic base and Nyquist sampling and WKS heuristics are prescribed for scarcity removal locally in a given sampled data subset augmenting the extent of inadequate data samples therein; and, statistical resampling *via* bootstrapping is performed to obtain an ensemble of surrogates of cardinality-enhanced sample set for use to train the ANN.

3. ANN-BASED PURSUITS IN FORECASTING EFFORTS

Artificial neural networks refer to a class of models inspired by and "made in the image of biological nervous system" [6]. The architecture of an ANN in general is made of a set of interconnected computing elements called neurons (units). The network can be designed with an input layer having a prescribed number of input units and an output layer with a single output unit. (However in general terms, the output layer of an ANN can have more than one output-target value. It is not mandatory to have only one output unit and the number of outputs is decided by the problem-specific requirement). In between, a set of hidden layers can be included with a designated number of neurons. The interconnections are synaptic links that connect to the inputs, output, or hidden neurons as illustrated in Fig. 1. A linear combiner is used to sum the output of the hidden layer so as to produce a single value (z_i) corresponding to the inputs addressed. That is, z_i implicitly denotes the weighted sum of all the inputs. It represents an activation signal and passed through a (sigmoidal) activation to produce a squashed output value (O_i), which can be compared against a teacher value (T_i) to yield an error (ε_i).

In general, there are several versions of ANN (such as Time-lagged Recurrent Networks, Generalized Feed forward Networks etc.) in vogue [23-26] with different topological features and schedules of operation. There are relative merits and demerits in their applications viewed in terms of the associated architectural complexity, convergence schedule, robustness of prediction performance, real-time usage etc. However, since the present study is exclusively devoted more on the application of ANN in forecasting strategies with a limited (scarce) data set available *a priori*, only a simple version that adequately does the intended application-specific effort is considered and adopted. It corresponds to a particular type of ANN known as the multilayer perceptron (MLP) shown in Fig.1. The MLP has a simple feed-forward architecture where in the information is rendered to propagate one way from input to output. Further, between input and output layers certain intermediate (hidden) layers are included. The resulting feed-forward, multilayer topology uses the backpropagation (BP) algorithm in successive iterations of input training set in order to minimize the output error with respect to a supervisory (teacher) value [6,14-18]. Notwithstanding the use of MLP as exemplar architecture in technoeconomic forecasting (with sparsity of data) as attempted here, the conceived approach can be applied to any other compatible ANN topology without any loss of generality.

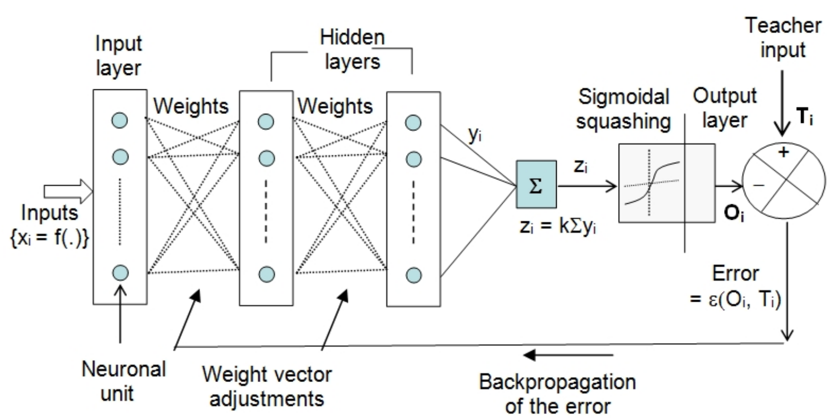


Fig. 1. The test ANN with MLP architecture

The MLP has the ability to "learn" the complex relationships between input and output patterns. The inputs to the networks correspond to certain data being fed forward; and, this feed-forward exercise enables the network to get "trained" or made to learn (or gain knowledge) on the data passed through in repeated ensembles, each having a stochastical dispersion with respect to the objective functional value. The learning process also involves a supervisory effort. That is, for a given input addressed, the squashed output obtained is compared against a teacher value (denoting the desired convergence or objective function) and the resulting error is back propagated iteratively so that, the network converges to yield almost a zero-error implying that its connecting synapses are duly weighted to hold the trained data pattern as a memory set. The training of the ANN is repeated multiple times with an ensemble of input data-sets, so that the network converges robustly. Subsequent to training, suppose a new input set (each having distinct stochastical features as mentioned earlier) is addressed. Then the network converges to the input received and concurrently the output error is seen minimized. That is, the network output designates the closeness of the new input set to the trained pattern.

The test MLP (Fig. 1) adopted in the present study has a simple architecture with only one hidden layer and both input and hidden layers consist of ten neurons each. Further, it has a single output node wherein the output is compared against a prescribed teacher value. Any resulting error is sensed and applied *via* backpropagation efforts that follow the suite of an algorithm that changes or adjusts the weights of the neural network iteratively. That is, a gradient-vector of the error surface is calculated and this vector follows the direction of steepest descent at each neural node in the backpropagation, so that the error is iteratively reduced seeking eventually the global minimum of the error trough.

In realizing ANN architectures such as MLP toward robust prediction performance, it is necessary that the test network should be trained adequately with sufficient extent of input data. In other words, for any meaningful classification of a given set of data *via* ANN method, the training of the ANN should be exercised with several iterations of input data set and each set is exhaustively defined. Then, the network will be robust in identifying and classifying a given set of new data freshly addressed on its input.

Most often the availability of ANN training data set will rather be restricted in numbers. More so, any such data set may not be of continuous to yield the full details across the sampled values of the data. Largely such data acquired in real-world context may correspond only to fragmented sampled details and as such, the data would contain only a partial information insufficient for robust training of the ANN. Essentially neural networks are data-driven, self-adaptive units, which can adjust themselves to a data without any explicit specification of the underlying model function. That is, ANN tends to represent the universe of approximation of any function with arbitrary extent of accuracy. Here, the accuracy refers to classifying a set of data of its closeness to a known class. In this sense, neural networks are estimator of an outcome based on posterior probability [14]. The question of accuracy on classification however, depends on two major considerations: (i) The complexity of the architecture and (ii) the extensiveness of the training input data set. Normally, the architecture is conceived with minimal extent of computational burden and time taken by the network toward convergence. Having designed an ANN based on reduced complexity its performance in the prediction phase is decided by the extent of input training data on the learning patterns. In general sense, it is often assumed that the network is trained with sufficiently exhaustive set of input data. But, in reality as said earlier, the available data, in a particular context could only be sparse. Further, the functional dependence of the data is largely specified only as discrete sampled entity. Therefore pertinent to the scope of present study, in order to make the test

ANN design compatible to handle limited data set and still offer acceptable performance in the prediction phase, the following considerations and queries are posed as regard to a given an ANN architecture, such as the MLP:

- How simple or complex should the ANN be in its architectural features, trained with a sparsely-indicated sampled data set and still yield robustly predictive classifications?
- How best can the ANN be trained with a sparsely-indicated sampled data set?
- How can the given sample-data population be enhanced or sparsity recovered so as to improve the confidence level of ANN prediction?
- How can a multiple set of a data be created artificially, when only an assorted and limited set of random samples are available?

Relevant to the question on architectural considerations, the associated complexity is decided by the number hidden layers and the number of neuronal units accommodated in each layer. There are several heuristics and empirical details that have been evolved thereof [25,26] to specify the manner in which the neurons of an ANN are structured consistent with the learning algorithm pursued to train the network. That is, the learning algorithms (rules) and the associated number of neuronal units decide the extent of interconnections. More number of hidden-units is necessary to capture higher-order irregularities in a given behavior that cannot be expressed by simple co-occurrence of the states of the input units. That is, hidden units enable the construction of an internal representation of the states. Such internal representations are local with each connection strength between the units correspond to a meaningful relation between the units established *via* specified algorithm such as Hebb rule [27]. Eventually, the trained ANN bears in its memory (as steady-state coefficients of interconnection weights) of the trained patterns. The associated stable-states assumed by N neuronal units correspond to $2^{\alpha N^3}$ with α being an asymptotic constant. The computational complexity of the ANN is decided by the severity of achieving robustly this gross extent of stable-states [6,28]. As such, structural design of an ANN constitutes a very important phase in its construction and the architecture of an ANN has significant impact on a network's information processing capabilities. Given a learning task, an ANN with only a few connections and linear nodes may not be able to perform the task of capturing the intricate details in the training ensemble; and an ANN with a large number of connections and nonlinear nodes may over fit noise in the training data and fail to have good generalization and convergence ability [29].

In general, ANN architecture design is still very much a human expert's job. It depends heavily on the expert experience and a tedious trial-and-error process; and finding the most appropriate ANN structure is a very time consuming process [30]. Since there are no fixed or designated rules in determining the ANN structure or its parameter values, a large number of ANNs may have to be constructed with different structures and parameters before determining an acceptable model. For example, the selection of the optimal number of hidden layers (and hidden nodes) has a significant impact on the performance of a neural network, though typically decided in an *ad hoc* manner.

In short, any conceivable ANN architecture (specified in terms of its number of hidden layers and the associated neuronal units) should primarily be decided by the extensiveness of the problem in hand to be solved. Such extensive attributes refer to the following: (i) The amount of data to be used in training the test ANN expressed in terms of the number of ensemble sets available, number of items in each ensemble, and variety/diversity of the ensemble. The gross feature of the data set and the underlying cardinality, in turn dictate microscopically resolvable details in the local (internal) representation of the ANN ‘black-box’. Larger the size of the gross features would require significantly an excessive number of hidden units; and as such, the associated distributed interconnections would become enormous making the learning much harder. So, given a set of data intended for data-mining *via* ANN, the prudent choice of internal complexity in an ANN is decided by facilitating just an adequate number of hidden layers supporting minimally optimal hidden units to perceive sufficient extent of associative capabilities and fault tolerance [31].

Typically, the number of neurons could be marginally higher so as to generate a compatible configuration of decision areas, complex enough for a given problem; however, if the number of neurons is excessive, the corresponding interconnections may become extensively complex as indicated above. Hence, there is a possibility of risk in achieving the balance between the converging trends of the connection weights *versus* desired computed outcomes using only the available examples. Lest, the network may generate noisy artefacts undermining the eventual output performance [32,33].

A heuristic suggestion with an empirical formalism in the literature [34] suggests that the number of hidden neurons should be between the size of the input layer and the size of the output layer (typically, about 2/3 the size of the input layer, plus the size of the output layer). Relevant choice is supposed to enable stable ANN learning. Hence, a rule-of-thumb on the number of hidden neurons (N_h) being equal to the geometric mean of the number of input and output neurons (N_i and N_o) respectively is indicated in [34]. Likewise, an approximate formula is also suggested in [34] to calculate the learning rate (η). It is given by: $\eta = 32/(N_i \times N_o)^{1/2}$.

Further, given a hidden architecture, the gradient-descent based optimizations in neural networks using backpropagation algorithm and the required learning rates etc. can also be possibility decided by invoking error-metrics other than the MSE, such as Csiszar's generalized error-metrics as proposed by one of the authors in [35].

Notwithstanding the methods based mostly on empirically-decided, rule-of-thumb suites as above in deciding the architectural complexity and evolving a relevant topology for the ANN compatible for use in a given problem, the appropriate ANN configuration can rather be more pragmatically chosen first on the basis of the following considerations mentioned earlier: (i) Extensiveness of the data being handled and the output performance sought; (ii) ability of the network to converge robustly and (iii) the convergence complying with real-time or non-real time applications [36].

Specific to the first consideration, the choice of the number of input neurons (N_i) should be consistent with the extent of the data available for application at the input layer. That is, the value of N_i should be pro-rated on one-to-one basis with the data structured for input toward training and prediction phases. Correspondingly, the number of output neuronal units (N_o) should match the number of output variables being sought in the prediction phase. Lastly, the number of hidden layers (and the neurons thereof) should be minimal to avoid computational complexity mentioned earlier and avoid undue iterations for learning convergence. Funahashi [37] among others, has shown that if a network which is able to take an arbitrary input pattern in the first layer, and provide an arbitrary desired output pattern in the last layer, all that is necessary is 3 layers (1 input layer + 1 hidden layer + 1 output layer) for minimal complexity (in Kolmogorov sense). Further, if any desired input-output function can be approximated in systems with one output neuron, such single-output systems can be easily concatenated into larger ones (with more outputs) which have essentially arbitrary approximate input-output properties.

The second and third considerations imply seeking a guarantee on network convergence (on real or non-real time basis) toward assimilating the input pattern assigned. It is entirely an application-specific issue [36]. Again, in general simpler networks yield faster convergence. In the present context, the MLP is chosen since no real-time, on-the-fly urgency exists in the technoeconomic forecasting being exercised.

Related to the above topological and convergence issues, yet there are other parameters such as learning rate, bias and nonlinear squashing being implemented in the test network. Invariably, the choice of nonlinearity is justifiable *vis-à-vis* the stochastics of the data as observed by Neelakanta et al. [38]. As such, as a limiting stochastically-justifiable function refers to the hyperbolic function that can be adopted in trial simulations. Further, relevant squashing nonlinearity can also be dynamically set as described in [39].

The learning rate (η) decides the slope of convergence [35]. If it is too small, the search for global minimum takes excessive time (in terms of iterations performed). Should it be of large value, the slope of convergence may over-step and miss the objective value set by the supervising (teacher) parameter. As such, (η) can be taken at the first instant of trial simulation as a small value (in the order of say, 10^{-3}) and, depending on the convergence (or divergence) trend observed, it can be altered.

A bias value is also introduced in ANN simulations to the summed and weighted inputs so as to reduce the closeness of the estimated error being offset far from the desired zero value. That is, relevant to the statistics of the data being processed, if the mean of the function being regressed is offset significantly, a bias value can be introduced to cancel this offset, so that the estimated error (between the regressed data and objective function) is small enough and feasibly reduced toward zero. Again, the need for the bias value is more of application- and data-specific.

In view of the above considerations, regardless of the variety in the types of ANN architecture that prevail, a feed-forward MLP network with 10 input neurons (consistent with the data in hand as described later), a single hidden layer with 10 neuronal units and a single neuron output is considered in this study. As will be seen with simulated results, it is simple lending itself of its parameters being chosen and varied transparently and accommodates the training and prediction data sets commensurately. In short, the 10 input neurons (N_i) are interconnected with 10 neurons set on a hidden layer; and the output facilitated on a single neuron for comparison against a supervisory value. The value of N_i = 10 is decided by a segment of input sample space of 10 values available in the window of the test data set being adopted (as detailed later). The number of neurons in the hidden layer is determined automatically by adapting to network complexity. That is, the design procedure complies with heuristics of providing a maximum of 10 neurons in the hidden layer facilitating minimal complexity of input-to-hidden layer connectivity. It balances the performance against the internal network complexity.

While the present study envisages a simple architecture commensurate with the data being handled, it is also designed to meet the context of a limited sample population for use as ANN inputs. Hence an appropriate scheme of ANN training methodology is formulated so as realize an optimal prediction model and yield a reliable network performance even under the conditions of source data being sparse. Relevant considerations are discussed in the following section.

4. APPLYING BOOTSTRAPPED DATA FOR ANN TRAINING

In the event of available data set being inadequate for robust ANN training, indicated in [19] is a method to build an ANN model with fairly accurate and reliable estimation performance. Hence advocated is a method of resampling based on the so-called statistical bootstrap method introduced by Efron in 1979 [40]. This method of bootstrapping became a popular statistical strategy in constructing more samples from a given sample set. When it is required to expand the realization of a statistical distribution of an inadequate variable set, creating a set of new bootstrap samples (of the variable) can provide a better understanding of the average and variability of the original, sparsely-known distribution of the statistical process. Bootstrap method in ANN modeling processes has been applied in the past to estimate the generalization error of an ANN; and such ANNs are known as bootstrapped ANN where, each bootstrap sample is used to exercise the underlying ANN effort in the training phase. The relationship between model inputs and model outputs is approximately elucidated in such ANN strategies. Bootstrapped ANN models have been used in a variety of applications, including experimental data-processing, prediction of foreign exchange rates, predicting curls in paper-making modeling of an airfoil.

The basic heuristics of bootstrap method and its use in ANNs is as follows: Assume a small sample-set $\{(x_1, y_1), (x_2, y_2),\ldots, (x_n, y_n)\}$ where x_i is the i^{th} input vector, depicting certain independent observations in a study and y_i denotes the corresponding dependent outcome variable; further, the observation pair (x_i, y_i) is independent and identically-distributed pertinent to an unknown/sparsely-known distribution, $F_{X,Y}$. A bootstrap approach can be invoked to replace this unknown/sparsely-known distribution $F_{X,Y}$ by a new, empirical distribution $F^*_{X,Y}$ with a probabilistic density on (x_i, y_i). That is, instead of generating new samples from the unknown distribution $F_{X,Y}$, a set of bootstrap samples are created by sampling with replacement of the empirical distribution, $F^*_{X,Y}$ with a probability density 1/n on each pair (x_i, y_i). Thus, bootstrapping involves "a sample substituted with a replacement". Because of the replacement process, each bootstrap sample may include multiple copies of some observations and no copies of other observations from the original set. The set of bootstrap samples can be expressed as $\{(x_1, y_1)_b, (x_2, y_2)_b,\ldots, (x_n, y_n)_b\}$, with b = 1, 2, …, B, and the total number of bootstrap samples, B considered in practice, may range usually from 50 to 200; and it could be even higher based on how many bootstrapped sets are needed (for example, to form an adequate ensemble of inputs to train an ANN). So, the value of B is need-based and is not decided by any optimal criterion. In the present study, an ensemble value of B = 50 is used purely on notional basis.

Given a data-set, an ensemble of B bootstrapped surrogate data-sets can be constructed and used to train an ANN, B times. Hence, the average performance of ANNs on their corresponding validation sets can be adopted as an estimate of the generalization error of the ANN models developed. Thus, bootstrapping of a data set and its application in ANNs are relevant to the state of sparse details available in a subset subjected to psedoreplication. The level of sparsity specifies the extent to which one can discard the associated (small) coefficients in the subset under resampling without much conceptual loss. More generally, sparsity recovery is a fundamental modeling tool that permits efficiently accurate statistical estimation and classification.

Relevant to the data set of [20] being adopted in the present study, as will be detailed in a later section, the entirety data being used toward ANN training is presumably regarded as sparse; and the stretch of (*ex post*) sparsely-sampled data space is specified as the set X: $\{x_j\}_{j = 1, 2, \ldots}$ with distribution F_X. This *ex post* data space is divided into $W_{k = 1, 2, \ldots, K}$ windows. The event spaces in each window are designated as epochs of the available samples at j = 1, 2, … with their values at the (random locations) (j) across the k^{th} window of the data space. Each window space is considered as a subset to be pseudoreplicated eventually.

There are two efforts involved in scarcity removal under consideration: (a) Improving the cardinality of the sampled data subset representing a window; and (b) pseudoreplicating the subset B times (as required) *via* bootstrapping.

Considering the first effort, suppose 10 samples in each window (subset space) are needed to represent adequate details for use as 10 inputs in the test ANN; but, there are lesser number of samples prevail (say, 7 samples) in that window. Hence, Nyquist sampling and WKS heuristics are invoked to obtain 3 extra samples (to make up 10). In summary, while attempting to improve the cardinality of a subset (denoting a window), additional samples as required are implanted in the subset in the vicinity of existing samples. Such proximal samples are chosen on the basis of a certain criterion. The criterion adopted here is based on Nyquist sampling theorem [43] consistent with Whittaker [44]–Shannon [45] heuristics as mentioned earlier and explained later.

Next effort is concerned with constructing pseudoreplicates of the subset, each with an enhanced cardinality of 10. Such surrogates are obtained via bootstrap procedure. The entity B mentioned above denotes the total count on the number of such bootstrapped subsets (of size 10 each) simulated for each window and adopted as an ensemble of B times training entities for the test ANN.

In general, resampling can be done by two methods namely: (i) Using subsets of available data (jackknifing); or (ii) drawing randomly with replacement from a set of data points (bootstrapping). That is, both bootstrapping and jackknifing denote sampling methods with replacement *versus* leaving out one observation at a time. In other words, in bootstrapping the original data set is sampled randomly but with replacement to produce "psedoreplicate" data set (also known as surrogate population of phantom or copies). Each pseudoreplicate consists of elements as the original data set, but may not include all the original elements; some elements may appear more than once, others not at all. This replacement procedure in general, can be repeated thousands of time leading to the desired value B; and each iteration will produce a new pseudoreplicate from which the sample statistics can be deduced at least to an approximate extent.

In contrast, jackknifing produces a limited number of pseudoreplicate data sets, each of which contains all but one of the original data elements. Given a data set with M elements, M pseudoreplicate data sets are generated, each lacking a different data element. Since jackknifing requires far fewer iterations, it is considered as an approximate of bootstrapping.
Typically, the bootstrap method gives different results when repeated on the same data, whereas the jackknife gives exactly the same result each time. Because of this, the jackknife technique is popular when the estimates need to be 'verified' several times. Whether to use the bootstrap or the jackknife may depend more on operational aspects than on statistical concerns involved. The jackknife, originally used for bias reduction, is more of a specialized method and only estimates the variance of the point estimator. This can be enough for basic statistical inference (for example, hypothesis testing, confidence intervals etc.). The

bootstrap, on the other hand, first estimates the whole distribution (of the point estimator) and then computes the variance from that. Thus, the bootstrap is mainly recommended for distribution-specific estimations [41,42]. The present study is concerned with getting distinct surrogate data sets for iterated use in ANN training; and each surrogate data represents a new set with the scarcity of available data removed. As such, bootstrap method is adopted here to realize a batch of B data set usable as an ensemble for ANN training. That is, to resample (with replacement) from the scare sample data at hand and create more number of phantom samples (or bootstrap samples) for ensemble testing of the ANN.

Further, the procedure of resampling can be done in line with 'wild bootstrapping method' due to Wu [42] where new surrogates are picked to deliberately assume heteroskedasticity (meaning variability from others). Here, the sub-populations of new samples introduced in the subset under bootstrapping (*in lieu* of actual replacements) can be rendered to have different variability. That is, the replacements are 'tweaked' to show some 'variability' on *ad hoc* basis and thereby expected to statistically augment the statistical features of the ensemble set of surrogates adopted for supervised ANN learning.

Since bootstrapping involves "a sample substituted with a replacement", natural queries of interest are as follows: To what extent can the bootstrapped samples be adopted (without any loss of generality) in a supervised process such as in an ANN *vis-à-vis* performance issues? (ii) Could the repetition of same observations-prototypes affect ANN's learning ability leading to over-/under-training and hence, influence (adversely) the memorization of the pattern involved? These queries can be overviewed in terms of the heuristic considerations presented below.

For successful and robust application of bootstrapped ANN (BSD-ANN) conceived here to support the under-sampled data environment, the following considerations can be identified:

- Cardinality improvement in a given window (subset) having inadequate sample counts: This is done by adopting a compatible sampling rate such as, Nyquist-Shannon criterion on sampling. The Nyquist-Shannon sampling theorem specifies an upper-bound on the sampling interval (ΔT) of a discretized signal (sampled data) in time-domain such that the sample contains all the available frequency information from the signal. The entity ΔT is known as Nyquist interval. In short, the Nyquist-Shannon criterion stipulates that for a lossless capture of information, the sampling should be at least twice faster than the largest frequency content of the signal. (Nyquist-Shannon theorem is generally known as Nyquist–Shannon–Kotelnikov, Whittaker–Shannon–Kotelnikov, Whittaker–Nyquist–Kotelnikov–Shannon, or simply, WKS cardinal theorem of interpolation theory) [43-45][46-49]. In summary, given a sample set satisfying Nyquist interval ΔT, it can be stated that in the vicinity of this sampling interval, the data points in the probability density function (PDF) are minimally correlated
- Type of sampling: In constructing a cardinality-enhanced subset as above both uniform (equally-spaced) or non-uniform (randomly spaced) sample space [46] [47] can be validly considered
- Information recovery and its consequences in a test ANN: Given a single subset of (cardinality-improved) sampled data subset (evolved by satisfying the Nyquist-Shannon sampling theorem), the problem is to derive an ensemble of B such sampled-data sets (from the given single subset) as required for training the test ANN. Relevant construction of B surrogates in forming the required ensemble

should conform to asymptotically linear multiples of sample-sets being necessary and sufficient to train the test ANN adequately enough *vis-à-vis* the information recovery being robust in the prediction phase. As such, when surrogates are created to form an adequate number of subsets constituting the ensemble required, the constructed pseudoreplicates would conserve the minimal correlation at each bootstrap value so as to preserve the underlying information in the data structure of the original sampled data-set adopted for bootstrapping.

The methods of (a) improving the cardinality of samples in a given window (subset space) and (b) statistically enhancing the number of subset space by creating B pseudoreplicates *via* bootstrapping are described below in terms of relevant computational algorithms outlined *via* pseudocode.

Pseudocode on scarcity removal procedure on a sparse sampled date space

// **Methodology:**

(a) Enhancing the cardinality of a subset with limited and inadequate samples therein
(b) Resampling (using statistical bootstrapping technique) to replicate the number of such subsets so as to realize an ensemble subsets of desired size.

// **Procedure (a):** Cardinality enhancement in a subset: Nyquist sampling and WKS heuristics

Initialize
Input

→ Sparsely-sampled data space of the set X: $\{x_j\}_{j = 1, 2, \ldots}$ with distribution F_X

← Identify the k^{th} window of the data space wherein the bootstrapping is to be done

← List epochs of the samples at j = 1, 2, … and note their values at each epochs at locations (j) across the k^{th} window of the data space

%% Illustrative example: Fig. 2

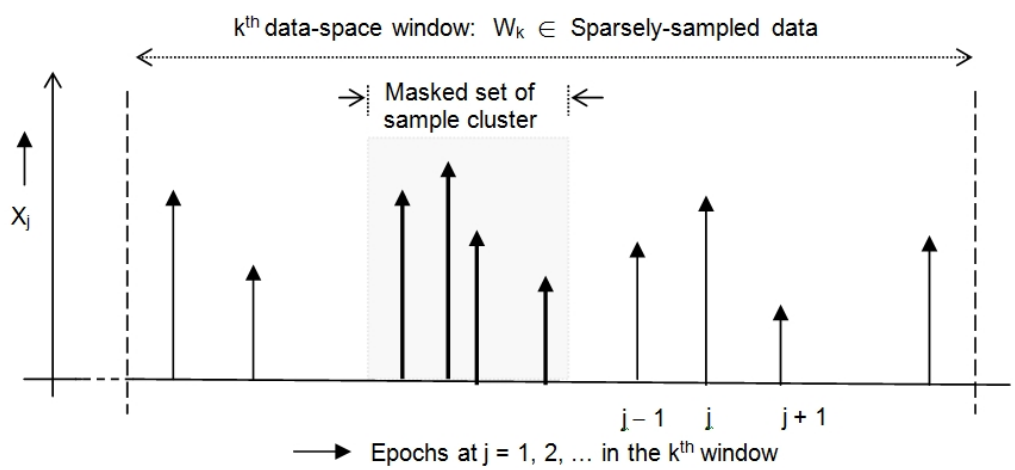


Fig. 2 Illustration of clustered and non-clustered parts of the epochs denoting sparsely sampled data set in the k^{th} window

Perform

→ Step I: Segregation of clustered sample

← Filter out/mask the clustered sample subspace as illustrated in Fig.2
← Masked subspaces with clustered (closely-packed) will be excluded for cardinality enhancement

Identify

→ Candidate samples/epochs in the unmasked, non-clustered sample space: Step I

← The identified samples, $\{X_\ell\}$ are subjected to cardinality enhancement as follows:

Perform

→ Sample-size (cardinality) enhancement in a given window/subset: A method of enhancing the number of samples/epochs artificially at the candidate samples/epochs identified in the unmasked, non-clustered sample space of Step I: $\{X_\ell\}$

← These samples in the non-clustered section can be subjected to cardinality enhancement as follows:
← Track the slope of a candidate sample/epoch X_ℓ at the location ℓ across its neighbors $X_{\ell-1}$ and $X_{\ell+1}$ at locations (ℓ – 1) and (ℓ + 1) respectively.

Go to

→ Step II: Cardinality enhancement procedure *via* bilateral slope tracking

← Here new samples/epochs are constructed *via* Nyquist sampling/WKS interpolation as follows: (See Fig. 3)

← With reference to a candidate sample/epoch X_ℓ at the location ℓ and its neighbors X_ℓ and X_ℓ placed at (ℓ – 1) and (ℓ + 1) respectively, two fresh epochs are constructed bilaterally as illustrated at ℓ_{B1} and ℓ_{B2}
← The values, (X_j at j = ℓ_{B1}) and (X_j at j = ℓ_{B2}) are decided by the slopes across the candidate sample and its nearest neighbors
← The locations of these samples/epochs namely, at j = ℓ_{B1} and j = ℓ_{B2} are set by the sample-intervals ΔW_{B1} and ΔW_{B2} respectively juxtapositioned with reference to the candidate sample/epoch at j = ℓ
← The sample-intervals ΔW_{B1} and ΔW_{B2} are constrained by the norms of the sampling theorem and Nyquist sampling rate
← (X_j at j = ℓ_{B1}) and (X_j at j = ℓ_{B2}) are bootstrapped samples obtained with reference to the candidate sample/epoch at j = ℓ having bilateral offsets ΔW_{B1} and ΔW_{B2} (that are less than or equal to Nyquist sampling interval)

%% Illustrative example: Fig. 3

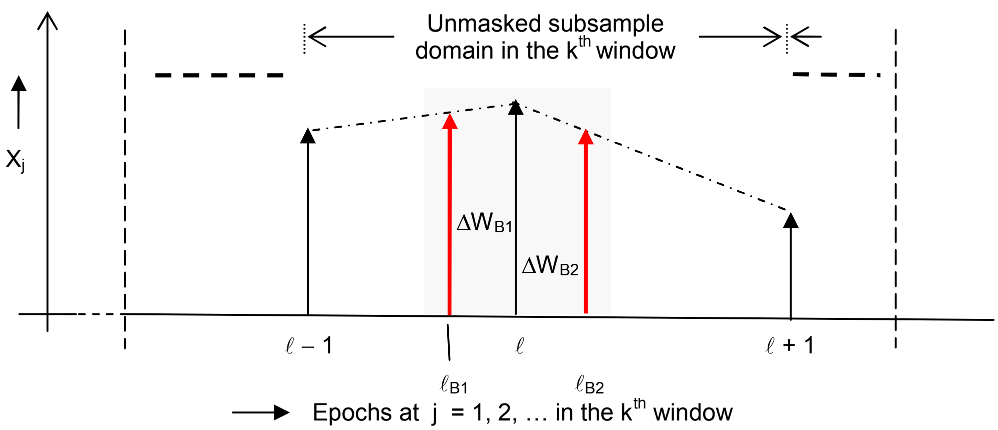


Fig. 3 Illustration of constructing new samples in the unmasked subsample domain in the kth window

Go to **Step III**

// **Procedure (b)**: Bootstrapping the cardinality-enhanced subset to obtain B surrogate subsets for use an ensemble of B sets in training the test ANN

Define

→ B: Number of bootstrapped data subsets (surrogates) to be created for each window (and used as ensemble in training the test ANN)

Perform **Do Loop**

→ **For** β = 1 to β = B (= 50 in the present study) for each window to generate B data set ensemble

Set → β = 1

← Generation of the first surrogate set for the kth window of the data space where in the bootstrapping is to be done

→ Constructing a pseudoreplicates of cardinality-enhanced subset (of kth window, originally with sparse samples) *via* resampling procedure of bootstrapping

Next

Continue Do Loop

→ Resampling is repeated to a desired extent of B times in realizing B new surrogates

→ Steps II -III can be repeated (to a desired extent of B times meaning creating a set of bootstrap samples for use as an ensemble of B runs pertinent to each window). This bootstrapping is exercised for each window

End Do Loop

← Thus, all the windows of interest can be subjected to population enhancement (each with B surrogate samples) as required
← Constructed B subsets are used as the input ensemble to train the ANN B times

End Procedures (a) and (b)

---

5. FORECASTING WITH SPARSELY-SAMPLED TECHNOECONOMIC DATA SPACE

To illustrate the efficacy and practical applicability of the proposed strategy, the technoeconomic data considered here for forecast applications refers to the performance details of a power-grid pertinent to electric energy accessed from wind-propelled electric generators [20]. Relevant details on the exemplar technoeconomic system chosen are as follows:

Abundant wind resources on the earth lend themselves as viable candidates of extensive energy source conceived *via* electric generators using wind-turbines as prime-movers. Hence, landscapes of wind-turbine farms are emerging profusely to cope with global demand for clean energy; and, the footprints of such farms are mushrooming and scattered all around the world at different geographical locales under diverse climatic conditions. As such, it can be expected that depending on location and season, the wind turbines may face significant variations in wind-speed and direction over different times of the year. Correspondingly, it can be concurrently expected that the electric power generated would fluctuate extensively with appreciable magnitude. Further, such fluctuations could largely be continuous, partly intermittent and mostly stochastical. Further, the electric current generated is proportional to the generator shaft-torque. Typically, the wind-propelled shaft is connected through a set of gear-train to the hub of the turbine and the output power is limited by controlling the torque produced by turbine blades. (Ailerons at the tip of the blades are used thereof to reduce the blade lift and this is necessary during high winds in order to protect the equipment facing uncertain fluctuations in wind characteristics).

In any case, it is obvious that the electric-power tapped from a wind-farm should be judiciously imparted to the power-grid despite of the associated stochastics so that the load-sharing is accomplished smartly across different energy sources supported on the grid *in situ* and evenly distributed. This will assure a productive technoeconomic ambient on the scenario in question.

In order to facilitate productive technoeconomics to achieve logical and intelligent load-sharing, it is necessary to know *a priori* the extent of power output available from a given wind-power resource; and, prediction of power generated from a set of electric wind-turbines can help proactive decisions on load-sharing on the grid. In other words, forecasting electric power output from a wind-turbine source can be a viable technoeconomic support in making of a successful smart-grid.

The question of reliable forecasting in such contexts of wind-power generator systems will depend on the following: Mostly, the data acquired in the past (*ex post*) on the performance of a wind-turbine system is pertinent to seasonal fluctuations in wind-speed and wind-direction *vis-à-vis* the corresponding electric power output delivered. Normally, relevant sampled-data are measured and stored (as regard the prevailing conditions of wind-speed and directions in a specified period); and, the stored data is subsequently adopted for predictions on the power delivery in future (*ex ante*). However, in order to make reliable predictions/forecast, first a compatible algorithm should be developed relating the power generated *versus* wind-speed and wind-direction details. That is, the ex post data corresponds to the result of a compatible mathematical algorithm which correlates the generated electric power with wind speed.

Hence, in the context of wind-turbine based electric-power generation outlined above, the method proposed here toward *ex ante* forecasting on the underlying technoeconomics is to adopt an ANN compatible to accommodate bootstrapped sample data at its input. The efficacy of the strategy proposed is determined by considering limited data set availed from [20] and expanded *via* bootstrapping. Hence forecast results are obtained and compared against those of [20] in the *ex ante* states of a single variable depicting the state of wind-speed/direction.

The data adopted in forecasting of [20] conform to a large, continuous, thorough and abundantly exhaustive set on the technoeconomics of wind-power generation. Relevant forecasting provides load-sharing details of the grid complex. On the other hand, the forecast evaluated in the present method uses only a limited and sparsely sampled data set pertinent to the same database. Hence, the merit and efficacy of the proposed strategy (that uses only a limited *ex post* sampled data) *versus* the forecast results based on exhaustive *ex post* data of [20] are elucidated. The following section provides the details on experimental simulations performed on a test ANN and the results obtained thereof are presented.

6. EXPERIMENTAL SIMULATIONS: DETAILS

The pursuit based on using ANNs in forecast efforts described in [20] relies on the ANN being trained with an exhaustive set of data available. Relevant data refers to details on generated power fluctuations *vis-à-vis* wind-speed and wind direction gathered at 5 second and 10 minute averages. Correspondingly, the simulations in [20] show a robust performance of the test ANN adopted. Thus, in general, the feasibility of using ANN is now a proven technique for the application in question when an abundantly exhaustive input data is available to train the ANN as in [20].

However, suppose the data-set in question is sparse; that is, when it represents only a limited sample space, not so exhaustive. The query as posed earlier is that, to what extent such a limited sampled data-set can train the ANN robustly and yet yield reliable output predictions. The above query is significantly important for, in many circumstances data acquisitions may not be continuously feasible nor totally exhaustive. Mostly, data is collected as samples either periodically and/or on aperiodic basis; and as such, (i) no continuous information may be available; and, (ii), the extent of ensemble data (in terms of samples available) is limited in number. Therefore, in such limited (sampled) data space, the information addressed to train the ANN may rather be inadequate. As a result, the training of ANN will not be robust enough to yield reasonably accurate predictions.

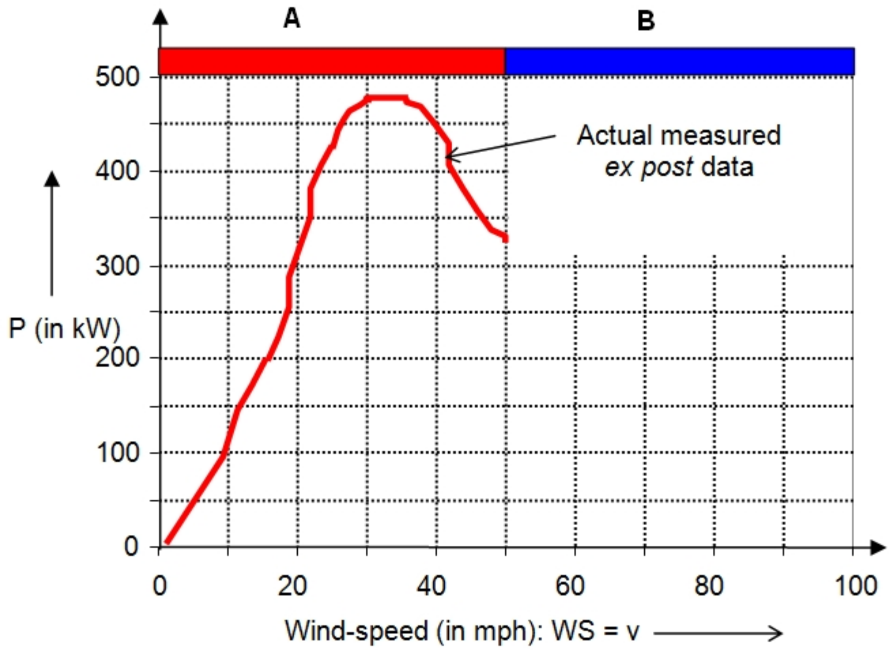


Fig. 4. Wind-power generation *versus* wind-speed averaged across an ensemble of wind directions. (A and B denote the assumed *ex post* and *ex ante* regimes indicated for forecasting purposes in the present study). The ex post subset corresponds to the available data space of wind-speed (up to 50 mph) as in [20]

Hence, as mentioned earlier, attempted in this study is to invoke the concept of statistical bootstrapping and apply it to the limited data space of the *ex post* regime. Then the resulting, enhanced extent of data availed from bootstrapping is used for ANN training so that subsequent output prediction becomes more robust and reliable in practical contexts. That is, without increasing the architectural complexity, realizing a robust ANN-based prediction is studied by constructing adequate number of bootstrapped sampled data-sets. They are then addressed as the input sets during the training phase of to test ANN. In short, it is surmised that the augmented data space (derived *via* bootstrapping the available sparse data-set) when applied to a test ANN of simple architecture would enable robust prediction performance on the output results sought.

The data adopted in the present study is same as that in [20]. Shown in Fig. 4 is the measured wind-turbine power generated (P in kW) *versus* the average wind-speed (in mph) gathered from the measured cluster of data with varying wind directions. The combined influence of wind-speed (v = WS) and wind-direction (θ) on the general trend in power generation (P) is approximately specified by a functional relation proportional to: $f(v) \times g(\theta)$ where the functions f(.) and g(.) are explicitly depicted in Fig. 5 in normalized scale. Specific to Fig. 5, the details obtained as field data, the functions f(.) and g(.) are empirically modeled in [20] as compressing functions created *via* error-and-trial method using an exhaustive field data set. That is, on the basis of extensive data collected in the wind-farm consistent with the power generation profile of Fig.4 f(.) and g(.) are decided to fit closely the empirical suites. These functions are now considered as predictive algorithms in the present study concerning forecast predictions on wind-power generated *versus* (v and θ). That is, the functions f(.) and g(.) are adopted in the test ANN simulation experiments. (The functional value of [f(v) × g(θ)] is taken in normalized form and normalization is done with respect to the maximum value).

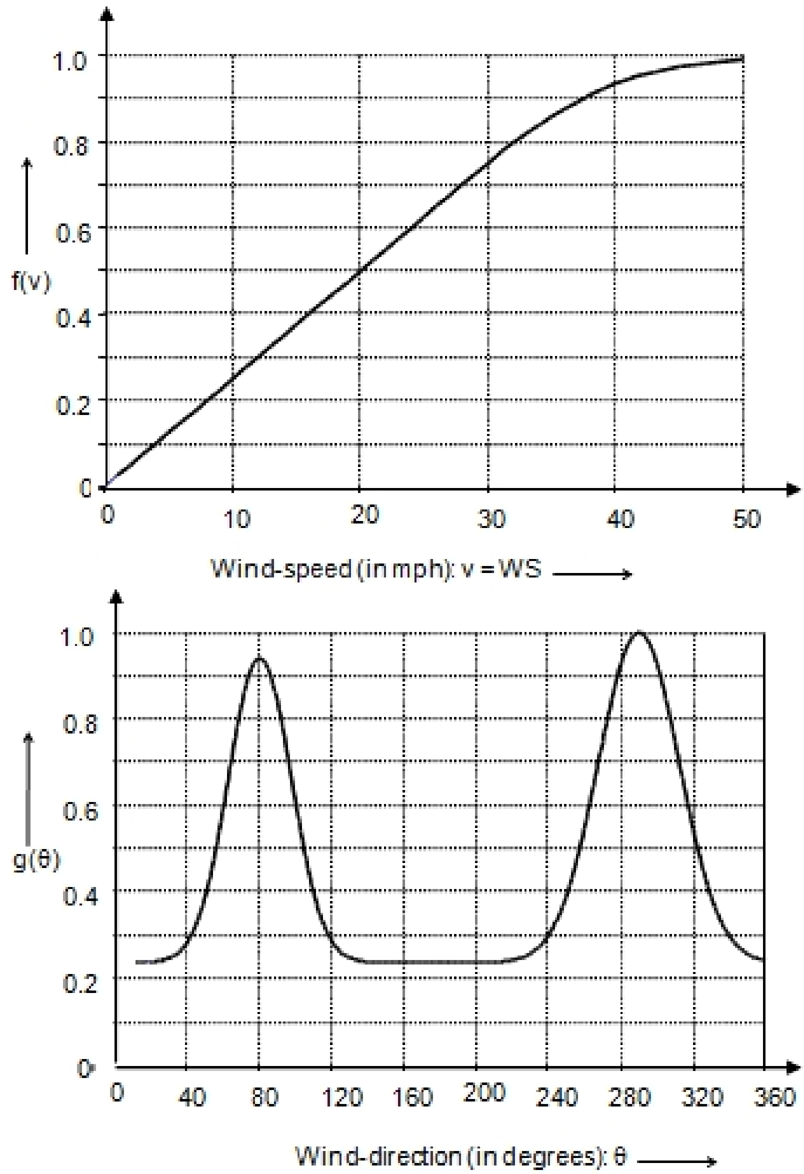


Fig. 5. Functional relations that decide the wind-power generation (P) *versus* wind-speed (v) and wind direction (θ) and P is proportional to: f(v) × g(θ) in normalized scale

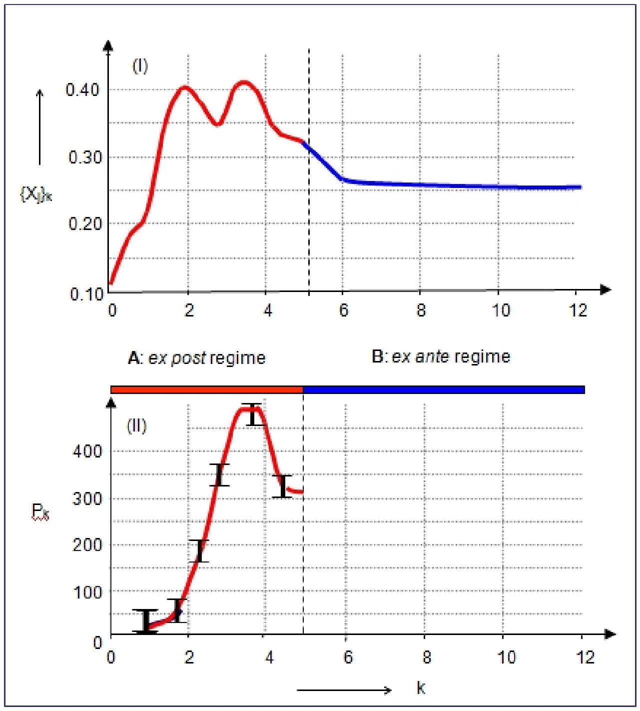


Fig. 6 (I). ***Ex post*** **$(X_j)_k = \{f(.) \times g(.)\}_k$ (normalized) values correspond to the data on f(v) and g(θ) of Fig. 5** ***versus*** **window # k of the wind-speed up to k = 5 (or v ≈ 50 mph). The** ***ex ante*** **(blue line) values of $(X_j)_k$ refer the extrapolated details on the** ***ex post*** **data regressed in the vicinity of (*ex post*)-to-(*ex ante*) transition adopted towards forecasting [3]. (II) Power delivery P_k in kW** ***versus*** **window # k. (As mentioned in the text, the window number k is an index representing the k^{th} sub-segment window (denoted as W_k) of the wind-speed (v = WS in mph). Explicitly, W_k corresponds to the sub-segment of wind-speed sequenced in** ***ex post*** **regime as follows: (k = 1: WS = 0-8; k = 2: WS = 8-16; k = 3: WS = 16-24; k = 4: WS = 24-32; k = 5: WS = 32-40). The window designations for the** ***ex ante*** **regime are: (k = 6: WS = 40-48; k = 7: WS = 48-54; k = 8: WS = 54-60; ….; k = 13: WS = 92-100)**

For forecasting purposes, the stretch of data in Fig. 4 is bifurcated into two assumed regimes, *ex post* (A) and *ex ante* (B) as illustrated. In each regime, sub-segments of wind-speed are designated as W_k. They correspond to wind-speeds sequenced in *ex post* regime as follows: (k = 1: WS = 0-8; k = 2: WS = 8-16; k = 3: WS = 16-24; k = 4: WS = 24-32; k = 5: WS = 32-40); further, the window designations for the *ex ante* regime are: (k = 6: WS = 40-48; k = 7: WS = 48-54; k = 8: WS = 54-60; ….; k = 13: WS = 92-100).

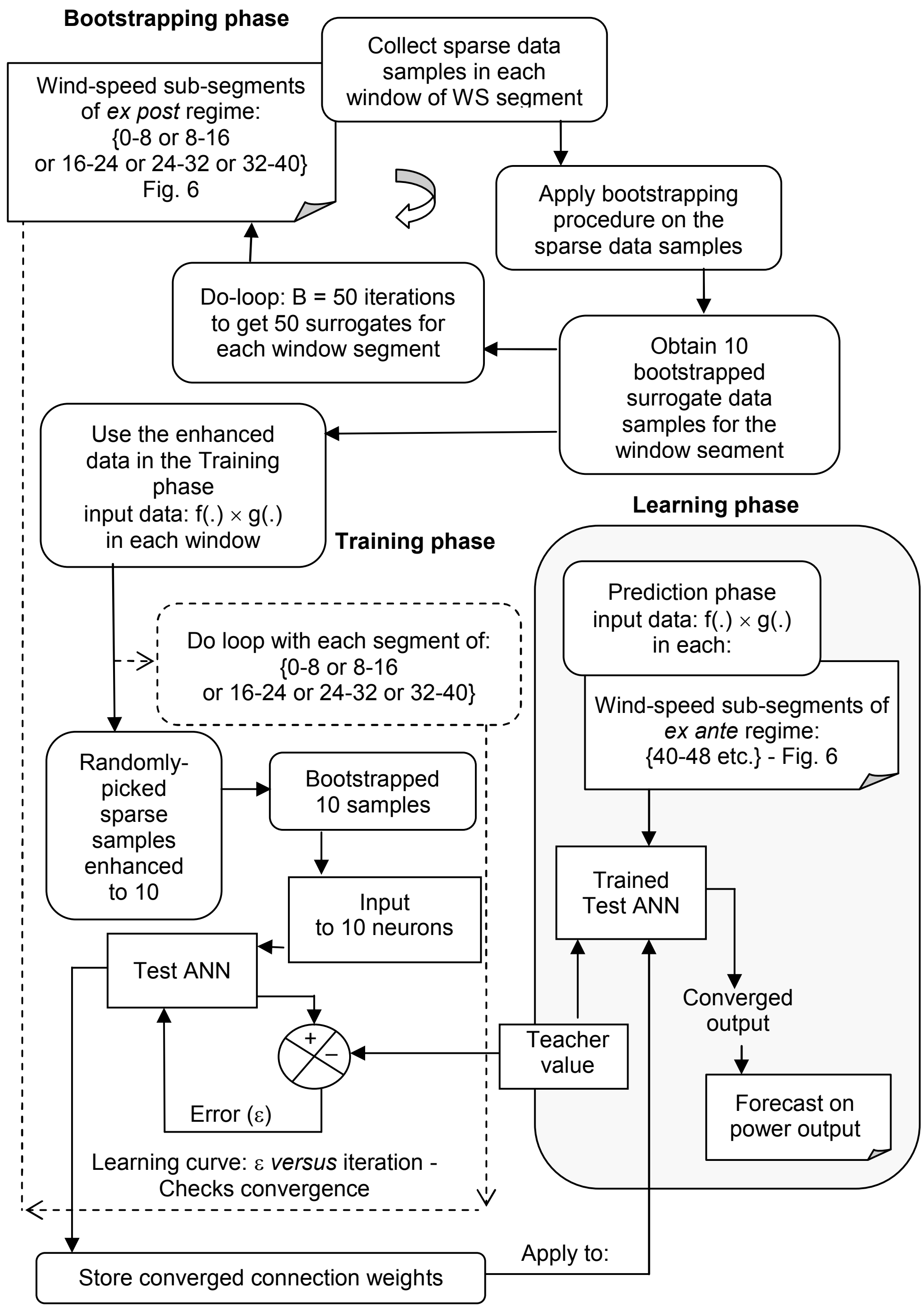


Fig. 7. Flow-chart on the procedure for constructing the bootstrapped sample-space and applying it on the test ANN in the training phase and subsequent prediction phase exercised for forecasting in the *ex ante* regime

In the assumed *ex post* regime (A), $0 \leq (WS = v) \leq 40\text{-}50$, a set of sparsely sampled-data are noted across each of the wind-speed segments namely, {k = 1, k = 2, …, k = 5} $\Rightarrow$ {v = 0-8, 8-16, 16-24, 24-32 and 32-40}. Then, in each window (segment), the prevailing sparse samples (less than 10 values) are enhanced to an extent of 10 *via* bootstrapping exercised on 1 or 2 or on more sparse-samples as necessary (Table 1). Next, these 10 bootstrapped samples in each segment are adopted as inputs to train the test ANN. Iteratively, B = 50 surrogate data sets are generated to represent as the ensemble of training sets for each window segment presumed.

The *ex post* values of $(X_j)_k = \{f(.) \times g(.)\}_k$ are first gathered from the data on f(v) and g(θ) of Fig. 5 *versus* window # k of the wind-speed up to k = 5 (or v $\approx$ 50 mph). Relevant plot of $(X_j)_k = \{f(.) \times g(.)\}_k$ is presented in Fig. 6(I).Correspondingly, for each subset indicated above, the mean value of power delivery P_k expressed *via* the (normalized) functional relation $f(v) \times g(\theta)$ is determined and used as the corresponding teacher value in ANN simulations (Table 1).

The *ex ante* (blue line) values of $(X_j)_k$ refer the extrapolated details on the *ex post* data regressed in the vicinity of (*ex post*)-to-(*ex ante*) transition adopted towards forecasting as described in [3]. (II) Power delivery P_k in kW *versus* window # k. The window number k is an index representing the k^{th} sub-segment window (denoted as W_k) of the wind-speed (v = WS in mph).

The complete procedure adopted in constructing the bootstrapped sample-space and applying it on the test ANN during training phase and enabling forecasting in the prediction phase of the *ex ante* regime is illustrated in Fig. 7.

7. RESULTS AND DISCUSSIONS

To perform the training schedule on the test ANN, an ensemble of B = 50 subsets (for each window are generated) with random sparse-samples followed by bootstrapping as indicated above. Table 1 lists the entire ensemble of simulated (bootstrapped) sample-space adopted in test ANN training and the corresponding details on the convergence are also presented in Table 1. Examples of relevant learning curves (deduced with *ex post* data) confirming the convergence of ANN training are presented in Figs. 8(a) - 8(e).

Indicated in Table 2 are details on the prediction phase of *ex ante* regime spanning the wind-speed sub-segments (40-48), (48-56) …, (96-100). In each sub-segment, the converged test ANN output expressed in {f(.) × g(.)] is indicated and the corresponding actual and predicted values of the generated wind-power in kW are also presented. Fig. 9 illustrates the summarized results with the corresponding percentage error observed in the predicted data.

Table 1. Example details on the training phase with *ex post* (A) data

Test ANN: Description

Multi-layer perceptron (MLP) depicting a feed-forward, multilayer architecture, which uses backpropagation algorithm in successive iterations to minimize the output error sensed with respect to a supervisory (teacher) value (Fig. 1).

Number of input neurons:	10
Number of hidden layers:	1
Number of output unit:	1
Nonlinear squashing function:	Hyperbolic tangent
Error function:	Mean-squared error (MSE)
Bias value:	0

Test ANN: Training phase with *ex post* data

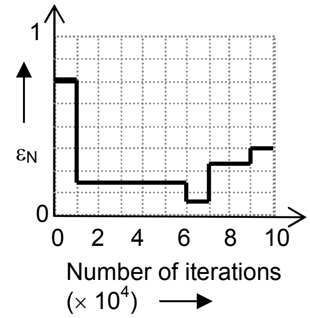
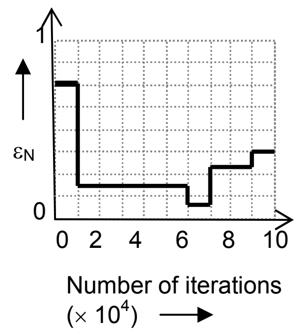
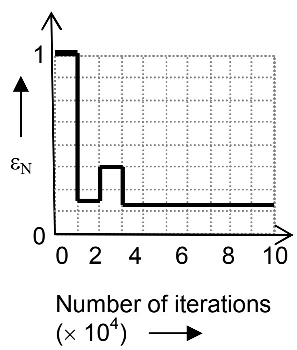
Input values $(X_j)_k$, j = 1, 2, …, 10 correspond to: [f(.) × g(.)] values known at the i^{th} instants of sample occurrence within the k^{th} sub-segment window of wind-speed (WS in mph) denoted by $(W)_k$ across the *ex post* regime divided into k-windows. Explicitly, $(W)_k$ corresponds to the sub-segments of wind-speed: k = 1: WS = 0-8; k = 2: WS = 8-16; k = 3: WS = 16-24; k = 4: WS = 24-32; k = 5: WS = 32-40; and, each window has 10 epochs or samples that may include bootstrapped values. Corresponding to each window segment, a teacher value T_k is prescribed and it is equal to: Ensemble average of $\{X_j\}_k \Rightarrow \{[f(.) \times g(.)]_j\}_k$ training data adopted in the k^{th} sub-segment of interest across the *ex post* regime

Examples of teaching schedule illustrating the convergence of the network

(The input values shown **bold** correspond to entities obtained to enhance the sparse sample-set to 10 in number *via* bootstrapping)

Note: The input values shown depict one subset of B different possible subsets constituting the training ensemble.

Window $(W)_k$ In the *ex post* regime: ANN training phase	**Input values $\{X\}_j\|_k$**	**Converged error (ε) (×10^{-4}) at test ANN output**	**$(T)_k$**	**Learning curves: Normalized error (ε_N) *versus* Number of iterations. (Normalization done with respect to maximum error seen at the start of the iterations)**

k = 1 WS: 0-8 mph	0.000 **0.075** 0.235 0.075 **0.075** 0.100 0.100 **0.250** **0.250** 0.255	0. 50	0.155	 Fig. 8(a). Learning curve for $W_{k=1}$
k = 2 WS: 8-16 mph	**0.025** **0.025** 0.100 **0.235** **0.235** **0.235** **0.235** 0.235 **0.250** **0.250**	6.80	0.198	 Fig. 8(b). Learning curve for $W_{k=2}$
k = 3 WS: 16-24 mph	**0.025** **0.025** 0.100 **0.235** **0.235** **0.235** **0.235** 0.235 **0.250** **0.250**	2.40	0.250	 Fig. 8(c). Learning curve for $W_{k=3}$
	0.250 0.235 **0.275** **0.275** **0.300** **0.300**	1.40	0.271	

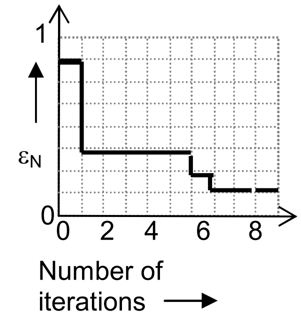
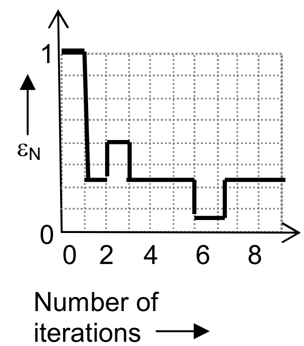
k = 4 WS: 24-32 mph	**0.235** 0.236 0.375 0.236			 Fig. 8(d). Learning curve for $W_{k=4}$
k = 5 WS: 32-40 mph	0.250 **0.235** 0.275 **0.236** 0.300 **0.236** **0.235** 0.236 0.375 0.236	2.40	0.261	 Fig. 8(e) Learning curve for $W_{k=5}$

Table 2. Test ANN: Prediction phase (B: *ex ante*) results and forecast data on power generation

Window $(W)_k$ In ex ante regime: **ANN prediction phase**	***Ex ante* regime: Input values $\{X\}_j$**	**Converged test ANN output value of:** [f(.) × g(.)]	**Power generated in kW**	
			Actual Value [20]	**Predicted *via* test ANN**
k = 6 WS: 40-48 mph	**0.250** **0.250** 0.275 **0.236** 0.300 **0.236** 0.235 0.236 0.375 0.237 0.250 0.235	0.272	479	470

	0.275			
k = 7	**0.236**			
WS: 48-56 mph	0.300			
	0.236	0.263	325	316
	0.235			
	0.235			
	0.375			
	0.375			
	0.250			
	0.250			
	0.275			
	0.236			
k = 8	0.300	0.250	300	285
WS: 56-64 mph	**0.236**			
	0.235			
	0.236			
	0.236			
	0.236			
	0.250			
	0.235			
	0.235			
	0.236			
k = 9	**0.300**	0.273	275	300
WS: 64-72 mph	**0.300**			
	0.325			
	0.236			
	0.375			
	0.236			
	0.250			
	0.250			
	0.275			
	0.236			
k = 10	0.300	0.271	270	290
WS: 72-80 mph	**0.326**			
	0.325			
	0.235			
	0.375			
	0.236			
	0.250			
	0.250			
	0.235			
	0.275			
k = 11	**0.236**	0.249	270	280
WS: 80-88 mph	**0.236**			
	0.235			
	0.236			
	0.236			
	0.236			
	0.250			
	0.250			
	0.100			

k = 12 WS: 88-96 mph	**0.235** **0.235** **0.235** **0.235** 0.235 0.375 0.236	0.243	270	280
k = 13 WS: 96-100 mph	**0.250** **0.250** 0.275 **0.236** **0.236** 0.275 0.235 0.236 **0.375** **0.375**	0.233	270	282

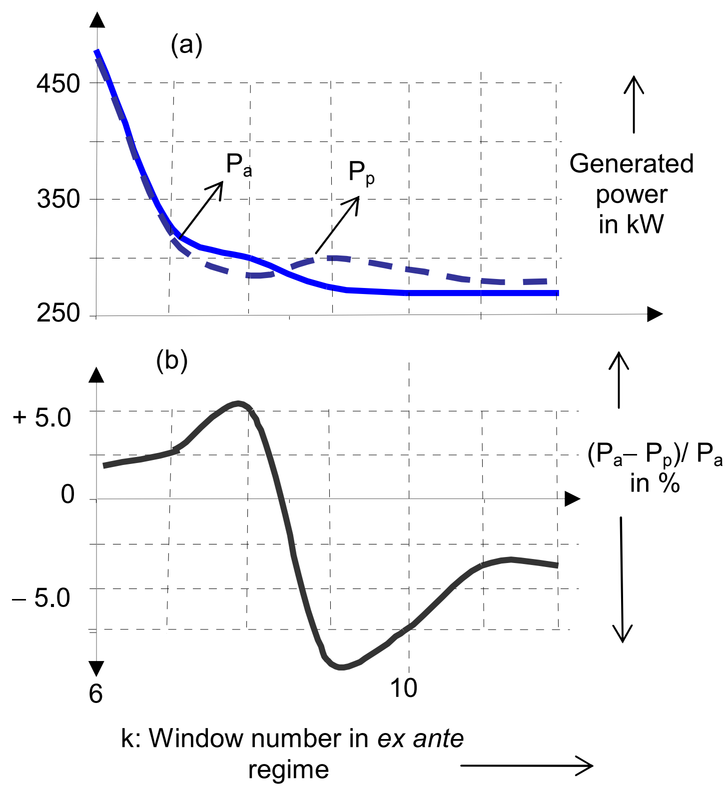


Fig. 9. Predicted results with the test ANN. (a) The actual power generated (P_a in kW) as in [20] and the predicted power (P_p in kW) versus window sub-segments k = (6, 7,…,13) as defined earlier; and (b): Percentage error of predicted result with respect to actual data of [20].

From the results presented above the following can be inferred:

- The proposed ANN strategy (of MLP architecture) is able to yield forecast projections when it is trained with an ensemble of *ex post* data
- In this context, the ANN can be exclusively designed for *ex ante* forecasting in even when the an ensemble set of sampled-data availed *ex post* is sparse and insufficient
- That is, even if the available *ex post* data samples are inadequate in number, relevant cardinality of the set can be enhanced *via* cardinality enhancement using Nyquist sampling and WKS considerations. The associated information recovery criterion is that the new samples generated should be asymptotically linear multiples and meet the Nyquist-Shannon condition [44-49].
- The cardinality improved subset of each window can be subjected to resampling to obtain a desired extent of B surrogates forming the ensemble set for ANN training
- Once adequate pseudoreplicates are constructed each with sufficient cardinality, the sparsity recovered information is addressed as input data at the test ANN; and, the associated pattern gets mapped on to the synoptic weights. When any new pattern is addressed later at the input, the test ANN classifies this new input detail *vis-à-vis* the mapped pattern
- With a real world technoeconomic data on wind-power generation complex, the efficacy of the proposed approach is illustrated. For the purpose of forecasting, the wind-speed domain is bifurcated into two regimes namely, A (*ex post*) and B (*ex ante*) as illustrated in Figs. 4 and 6. The test ANN is trained with *ex post* (Region A) details and the predicted results in *ex ante* (Region B) stretch of wind-speed are within about $\pm$ 10% of the actual value of the wind-power generated
- The associated convergence and forecast predictions are evident from the results indicated in Tables 1 and 2
- Without any loss of generality, the present method of forecast predictions can be exploited in similar technoeconomic contexts where insufficiency of data samples is observed; and, the performance of such structures can be elucidated optimally in terms of the underlying futuristic predictions (forecast projections) and the anticipated productivity across the *ex ante* regime.
- The accuracy of forecasting however relies on the extent to which the test ANN is trained with the ensemble of data (bootstrapped or otherwise) adopted during training. Specifically, the forecast trend would follow the extensiveness and stochastics of the *ex post* details that prevail just prior to and in the vicinity of (*ex post*)-to-(*ex ante*) transition as observed by Neelakanta and Yassin in [3]. Further, as indicated in [3], any features of forecast details (with constancy or jagged variations) as seen ahead in the *ex ante* frame could be true only to an extent within a cone-of-forecast. The method of constructing such progressive error-bar on the forecast values is detailed in [3].
- It will be of interest to determine the prognostic aspects of generated power when the wind speed is in the vicinity of 60 mph or higher. As per the details of Fig. 2 in [20], the generated power remains steady at wind speeds greater than 60 mph; and it appears that most commercial wind turbines currently do not operate for wind speed values higher than 50 mph. As such, specific considerations in elucidating the wind-turbine performance at 50 mph or more *via* forecasting and interpreting the results could be an open-question for future research if relevant field details are available. The present study, however is confined to the forecast results as presented in Fig 9 using the available details of Figs. 2, 7 and 8 in [20] viewed in terms of relevant functional relation [f(.) $\times$ g(.)].

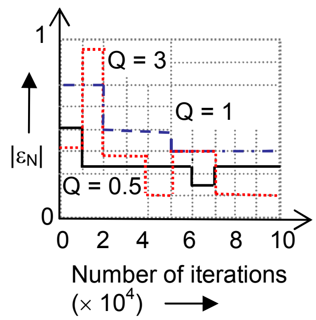
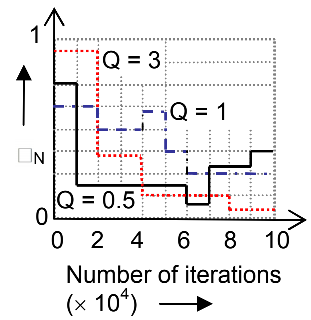
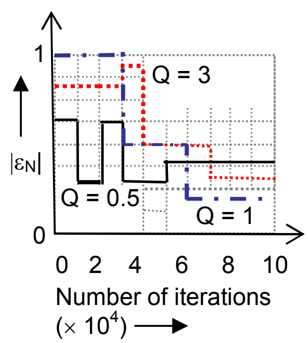
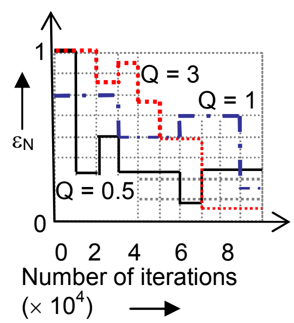
- The present study is confined to generating bootstrapped samples for use in an ANN context pertinent to one-dimension data. However how new interpolated samples can be generated relevant to a multidimensional scheme is an open-question for future research.

In summary, the present study proposes an overall strategy of using the ANN-based approach to realize robust forecasting with sparse data; and the methodology is illustrated with typical database information using a simple MLP architecture and its associated parameters as indicated earlier. However, it is desirable to make a comparative assessment on the performance efficacy of the approach using the test ANN as a function of its parameters. For example, suppose another type of nonlinearity is adopted (*in lieu* of the hyperbolic tangent function) in minimizing the cost-function *via* the regularization suite of backpropagation-specified iterations. Methods thereof (such as using splines that approximate the associated nonlinearity piece-wise are well known [25]). In general, relevant choice of nonlinear approximation of the underlying neurocomputing system warrants the associated basis function to be continuous, globally and locally; further, it is elementary and multiple times differentiable (required in gradient-seeking efforts of BP).

While the hyperbolic tangent function (adopted in this study) as well as other functions (like splines), satisfy the requirements as basis functions, yet another class of stochastically-justifiable sigmoids can be conceived toward emulating a generalized nonlinear function. It is known as the Langevin-Bernoulli function $L_Q(x)$ given by: $L_Q(x) = (1 + 1/Q) \times \coth[(1 + 1/Q)x] - (1/Q) \times \coth[(1/Q)x]$. As indicated by one of the authors in [50], $L_Q(x)$ is a viable, differentiable and a comprehensive logistic sigmoid in emulating the input (x) *versus* (nonlinear) output relation compatible for ANN applications; and $[(1/2) \leq Q < \infty]$ denotes the upper- and lower-bound parameters depicting the extent of disorder in the system. Further, it is shown in [50] that when $Q \rightarrow ½$, the $L_Q(x) \rightarrow \tanh(x)$. Hence, the choice of Q = 0.5 leads to the traditional hyperbolic squashing; and Q > 0.5, leads to more logistic activation feasibility and universal approximation of the nonlinearity involved. Typically, for $½ \leq Q \leq 3$, simple MLP architectures (even with a single hidden layer) are capable of approximating any continuous function on a compact set. Presently, $L_Q(.)$ function is used in the test ANN and illustrated in Table 3 are results on learning curves obtained with a couple of exemplary data sets (of windows: k = 1 and k = 5) adopted in original simulations (and presented earlier in Tables 1 and 2).

Another consideration of importance in specifying the goodness-fit of the neurocomputing system performance is to exemplify the results with at least a pair of error-metrics (for example, MSE and entropy measure such as Kullback-Leibler divergence [50]). Hence, cross-validations can be accomplished as indicated in Table 3 as regard to relative learning capabilities. The Kullback-Leibler error (KLE) can be specified as follows: Referring to Fig. 1, considering the entities (O_i and T_i) in normalized forms, the KLE denotes the mutual/cross entropy of divergence equal to: (KLE1 + KLE2)/2 where KLE1 = $[O_i \times \log_2(O_i/T_i)]$ bits; and KLE 2 = $[T_i \times \log_2(T_i/O_i)]$ bits. (The MSE denotes: $\frac{1}{I}\sum_{i=1}^{I}|O_i - T_i|^2$). The results on learning capabilities of the test ANN adopted show that regardless of the type of nonlinearity chosen and the error-metric used the test ANN shows robust convergence in learning schedules even when the input data conforms to a bootstrapped surrogate of a sparse data-space.

Table 3. A comparative assessment of test ANN performance in terms of learning curve convergence characteristics *vis-à-vis* distinct nonlinear modeling and using two different error-measures (MSE and KLE): Exemplary training phase details and results

Exemplar windows and input values: $\{X\}_j\|_k$	**Normalized learning curves with KLE metric**	**Normalized learning curves with MSE metric**
k = 1 WS: 0-8 mph **$\{X\}_j\|_{k=1}$** 0.000 **0.075** 0.235 0.075 **0.075** 0.100 0.100 **0.250** **0.250** 0.255	 Fig. 10(a). Learning curve for $W_{k=1}$	 Fig. 10(b). Learning curve for $W_{k=1}$
k = 5 WS: 32-40 mph **$\{X\}_j\|_{k=5}$** 0.250 **0.235** 0.275 **0.236** 0.300 **0.236** **0.235** 0.236 0.375 0.236	 Fig. 10(c). Learning curve for $W_{k=5}$	 Fig. 10(d). Learning curve for $W_{k=5}$

Lastly, the simulation results on ANN prediction phase presented in Table 2 are pertinent to a single subset of bootstrapped sets {B = 50} constituting the test ensemble. The compiled data on the average and standard deviation on the converged error-values are listed in Table 4. Hence the confidence interval of the predictions estimated provides the prediction efficacy as shown in Table 4.

Table 4. Statistics on the prediction phase (*ex ante*) results compiled with bootstrapped subsets of {B} = 50

Window $(W)_k$ In ex ante regime: ANN prediction phase	Predicted values of power generated in kW: Average value (μ) and standard deviation (σ)	Power generated in kW: Results as per [20]	95 % Confidence interval of predicted values	
			Lower end-point	Upper end-point
k = 6 WS: 40-48 mph	468 ± 2.5	479	467.3	468.7
k = 7 WS: 48-56 mph	312 ± 3.0	325	311.2	312.8
k = 8 WS: 56-64 mph	288 ± 1.9	300	287.5	288.5
k = 9 WS: 64-72 mph	305 ± 2.5	275	304.3	305.7
k = 10 WS: 72-80 mph	298 ± 3.0	270	297.2	298.3
k = 11 WS: 80-88 mph	275 ± 3.0	270	274.2	275.8
k = 12 WS: 88-96 mph	290 ± 2.8	270	289.2	290.8
k = 13 WS:96-100 mph	281 ± 1.5	270	280.6	281.4

8. CONCLUSION

Pursuant to a summary of details on the study performed as outlined above, relevant to the major queries posed earlier, the conclusive remarks and response can be listed as follows:

- Regarding ANN architectural complexity *versus* training the ANN with a sparsely available sampled data set and ascertaining robust predictive classifications: As elaborately presented in Section 3, the test ANN should be first designed with architectural simplicity/complexity (in Kolmogorov sense) with minimal hidden layer and neuron units); and based on observed performance, necessary enhancements can be incorporated to achieve desired input-output results.
- Given a sparsely-specified sampled data set, obtaining enhanced or sparsity recovered details (so as to improve the confidence level of ANN prediction) dictates relevant network convergence issues (namely, real or non-real-time considerations). It is however, entirely an application-specific requirement as per [36].
- Creating multiple set of a data artificially, when only an assorted and limited set of random samples are available could be accomplished by applying statistical either by bootstrapping or *via* jackknifing (with relative merits and demerits) as outlined earlier.

In all, the proposed BSD-ANN suggests the feasibility of assessing forecast projections effectively even with only a limited (sparse) sampled-data set available in practical technoeconomic contexts. Bootstrapping-based enhancement of sampled-data can be up to 30-40 % of the available data. The efficacy of the proposed method in the contexts of technoeconomics is illustrated with a practical real-world system wherein, sparsity recovery is done and relevant bootstrapped data sample-set is effectively applied to a test ANN for robust forecast purposes.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Neelakanta PS, Baeza D. Next-generation telecommunication and Internet engineering. New York: Linus Publications; 2009.
2. Tourinho RC, Neelakanta PS. Evolution of forecasting of business-centric technoeconomics: a time-series pursuit via digital technology, i- Business. 2010;2:181-200.
3. Neelakanta PS, Yassin RR. Information theoretics-based technoeconomic forecasting: application to telecommunication service industry. Netnomics. 2012;13:45-78.
4. Yusof FM, Aziz RA. Forecasting management and strategic adaption: case evidence from Malaysia. Proceedings IPEDR. 2011:4:428-32.
5. Henrion M. The value of knowing how little you know: The advantages of a probabilistic treatment of uncertainty in policy analysis. PhD Dissertation. Carnegie Mellon University, Pittsburg, USA; 1982.
6. Neelakanta PS, De Groff D. Neural network modeling: Statistical mechanics and cybernetic perspective. Boca Raton: CRS Press; 1994.
7. Jia Y, Culver T. Bootstrapped artificial neural networks for synthetic flow generation with a small data sample. Journal of Hydrology; 2006;331(3-4):580-90.
8. Allende H, Nanculef R. Robust bootstrapping neural networks. In Menroy R et al. (Eds), Lecture Notes in Artificial Intelligence, (LNAI 2972): Mexican International Conference in Artificial Intelligence MICAI-2004 (April 26-30, 2004 Mexico City, Mexico) Berlin: Springer Verlag. 2004;813-22.
9. Mohammed O, Park D, Merchant R, Dinh T, Tong C, Azeem A, Farah J, Drake C. Practical experience with adaptive neural network short-term load forecasting system. IEEE Transactions on Power Systems.1995;10(1):254-65.
10. Mohatram M, Kumar S. Application of artificial neural network in economic generation scheduling of thermal power plants. Accessed 20 February 2013. Available:http://mohtaram.pbworks.com/f/ApplicationofArtificialNeuralNetworkinEconomicGeneationSchedulingofThermalPowerPlants.pdf.
11. Abananthen KS, Sainarayanan G, Chekima A, Teo J. Artificial neural network approach in data mining. Malaysian Journal of Computer Science. 2007;20(1):51-62.
12. Singh Y, Chauhan AK. Neural networks and data mining. Journal of Theoretical and Applied Information Technology. 2009;5(1):37-42.
13. Online Chapter 6: Neural methods for data mining. Business intelligence: Management approach. Accessed 20 January 2013. Available: www70.homepage.villanova.edu/matthew.../turban_online_ch06.pdf
14. Kamruzzaman SM, Sarkar AMJ. A new scheme using artificial neural networks. Sensors. 2011;11:4622-47.
15. Popescu MC, Balas VE, Popescu LP, Mastorakis N. Multilayer and perceptron networks. WSEAS Trans. Circuit and System. 2009;8(7):579-88.

16. Adepoju GA, Ogunjuyigbe SO, Alawode KO, Tech B. Application of neural network to load forecasting in Nigerian power system. The Pacific Journal of Science and Technology. 2007;8(1):68-72.
17. Kuncoro AH, Zuhal, Dalimi R. Long-term load forecasting the Java-Madura-Bali electricity system using artificial neural network method. International Conference on Advances in Nuclear Science and Engineering in Conjunction with LKSTN. 2007:177-81. (November.13, 2007 University of Batan-Indonesia).
18. Mohatram M, Tewari P, Latanath N. Economic load flow using Lagrange neural network. Accessed 26 February 2013.
Available: http://ipac.kacst.edu.sa/eDoc/2011/193208_1.pdf.
19. Twomey JM, Smith, AE. Committee networks by resampling. Intelligent engineering systems through artificial networks. New York, NY: SME Press. 1995:153-58.
20. Li S, Wunsch DC, O'Hair EA, Geisselmann MG. Using neural networks to estimate wind turbine power generation, IEEE Transactions on Energy Conversion. 2001;16(3):276-82.
21. Vigneswaran T, Dhivya S. Analyzing the probabilistic distribution of the predicted wing speed. International Journal of Computer and Information Technology. 2012;01(2):88-93.
22. Kang MS, Chen CS, Ke YL, Lin CH, Huang CW. Load profile synthesis and wind power generation predictions for an isolated power system, IEEE Transactions on Industry Applications. 2007;43(6):1459-64.
23. Deshmukh RP, Ghatol AA. Short term flood forecasting using recurrent neural networks – a comparative study. IACSIT International Journal of Engineering and Technology. 2010;2:430-34.
24. Arulampalam G, Bouzerdoum A. A generalized feedforward neural network classifier. Proceedings of the International Joint Conference on Neural Networks. 2003;2:1429-34.
25. Haykin S. Neural networks. Upper Saddle River, NJ: Prentice Hall; 1999.
26. Fausett L. Fundamentals of neural networks. Englewood Cliffs, NJ: Prentice Hall; 1994.
27. Aarts E, Korst J. Simulated annealing and Boltzmann machine. New York, NY: John-Wiley & Sons; 1989.
28. Abu-Mostafa YS, St. Jacques S. Information capacity of the Hopfield model. IEEE Transactions on Information Theory. 1985;31(4):461-64.
29. Charnail E, McDermott D. Introduction to artificial intelligence. Reading, MA: Addison-Wesley; 1985.
30. Sinha M, Kumar k, Kalra PK. Some new neural network architectures with improved learning schemes. Journal of Soft computing. 2000;4(4):214-23.
31. Rumelhart DE, McClelland JL. Parallel distributed processing (Volume 1). Cambridge, MA: The MIT Press; 1987.
32. Hornik K, Stinchcombe M, White H. Multilayer Feedforward networks are universal approximators. Neural Networks.1989;2:359-66.
33. Hornik, K. Some new results on neural network approximation, Neural Networks. 1983;6:1069-72.
34. Shibata K, Ikeda Y. Effect of number of hidden neurons on learning in large-scale layered neural networks. Proceedings ICROS-SICE International Joint Conference (Fukuoka, Japan). 2009;5008-13.
35. Neelakanta PS, Abusalah S, De Groff D, Sudhakar R, Park JC. Csiszar's generalized error-metrics for gradient descent based optimizations in neural networks using backpropagation algorithm, Connection Science. 1996;8(1):79 -114.

36. Neelakanta PS, Preechayasomboon A. Development of a neuroinference engine for ADSL modem applications in telecommunications using an ANN with fast computational ability, Neurocomputing. 2002;48;423-41.
37. Funashi K. On the approximate realization of continuous mapping by neural networks. 1989;2:183-92.
38. Neelakanta PS, Sudhakar R, De Groff D. Langevin machine: a neural network based on stochastically justifiable sigmoidal function. Biological Cybernetics. 1991;65:331-38.
39. Neelakanta PS, Dabbas M, De Groff D. Constructive ANN with dynamically set sigmoid: A simulation tool for technoeconomic forecasting. International Journal of Latest Trends in Computing. 2012;3(2):30-7.
40. Efron B. Bootstrap methods: Another look at the jackknife. Annals of Statistics. 1979;7(1):1-26.
41. Shao J, Tu, D. The jackknife and bootstrap. New York, NY: Springer Verlag; 1995.
42. Wu CFJ. Jackknife, bootstrap and other resampling methods in regression analysis (with discussions). Annals of Statistics. 1986;14(4);1261-95.
43. Marks II RJ (Ed). Advanced topics in Shannon sampling and interpolation theory. New York, NY: Springer Verlag; 1993.
44. Whittaker E. On the functions which are represented by the expansion of the interpolation theory. Proceedings of Royal Society (Edinburg) Section A; 1915;35:181-94.
45. Shannon CE. Communication in the presence of noise. Proceeding of Institution of Radio engineers. 1949;37:10-21.
46. Candes E, Walkin MB. An introduction to compressive sampling. IEEE Signal Processing Magazine. 2008;25(2):21-30.
47. Krishanan MP, Rao KR. Compressive sampling techniques for integrating video acquisition and coding. Accessed 15 January 2013. Available:http://wwwee.uta.edu/Dip/Courses/EE5359/Compressive%20Sampling%20Techniques%20for%20Integrating%20Video%20Acquisition%20and%20Coding.pdf
48. Ackakaya M, Tarokh V. Shannon- theoretic on noisy compressive sampling. IEEE Transactions on Information Theory. 2010;56(1):492-504.
49. Battle DJ, Harrison RP. Maximum entropy image reconstruction from sparsely sampled coherent field data. IEEE Transactions on Image Processing.1997;6(8):1139-47.
50. Neelakanta PS (Ed). Information –theoretic aspects of neural networks. Boca Raton, FL: CRC Press; 1999.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history.php?iid=304&id=20&aid=2468