Advantages of Hybrid Deep Learning Frameworks in Applications with Limited Data

Valeriy Gavrishchaka, Zhenyi Yang, Rebecca Miao, and Olga Senyukova

***Abstract*—Recent advancements in deep learning (DL) frameworks based on deep neural networks (DNN) drastically improved accuracy in image recognition, natural language processing and other applications. The key advantage of DL is systematic approach for independent training of groups of DNN layers including unsupervised training of auto-encoders for hierarchical representation of raw input data (i.e., automatic feature selection and dimensionality reduction) and supervised re-training of several final layers in the transfer learning that compensate for data incompleteness. However, severe data limitations and/or absence of relevant problem for transfer learning can drastically reduce advantages of DNN-based DL. For example, pure data-driven auto-encoders dealing with high-dimensional input data require large amount of data for effective operation. However, hierarchical data representations can be also implemented without NN. Previously we have shown robustness of boosting-like algorithms for effective utilization of existing domain knowledge (e.g. analytical models) via discovery of compact ensembles of complementary low-complexity components. This approach can tolerate significant data incompleteness and boost accuracy of individual base models as was demonstrated in cardiac diagnostics applications. Here we argue that hybrid DL framework with auto-encoders replaced by components discovered by boosting followed by supervised NN could be more tolerant to data incompleteness compared to pure DNN-based DL. Illustrations based on cardio data from www.physionet.org are presented. The proposed framework could be utilized in many applications dealing with incomplete data including personalized medicine and rare or complex abnormalities.**

***Index Terms*—Auto-encoders, boosting, cardiac diagnostics, complexity measures, deep learning, ECG, ensemble learning, heart rate variability, hybrid learning, neural networks, physiological time series.**

I. Introduction

Recent advancements in deep learning (DL) frameworks based on deep neural networks (DNN) drastically improved accuracy of machine learning in image recognition, natural language processing and other applications [1]–[3]. Although universal capabilities of multi-layer NNs are well known, the key advantage of DL is systematic approach for independent training of groups of DNN layers. This includes unsupervised training of auto-encoders for hierarchical representation of raw input data (i.e., automatic feature selection and dimensionality reduction) and supervised re-training of several last layers in the transfer learning that compensate for data incompleteness in a particular application [1], [4]–[8].

However, in cases of severe data limitations and/or absence of relevant problem for the transfer learning, advantages of DNN-based DL are drastically reduced. For example, pure data-driven auto-encoders dealing with high-dimensional raw input data would require significant amount of data for effective operation even when stacked shallow auto-encoders are employed [6].

Advantages of hierarchical data and knowledge representations have been known well before recent raise of DNN popularity. This concept is ubiquitous in natural sciences where hierarchical approach is common in both fundamental theoretical frameworks and in practical simulations of complex systems. For example, success of realistic simulations of multi-scale spatiotemporal dynamics in plasma and space physics critically depends on proper formulation and coupling of physical models describing processes on micro- and macro scales, since it is infeasible to model wide range of scales from first principles because of computational limitations and lack of detailed initial/boundary conditions, e.g., [9].

In statistical and machine learning, a well-known example is family of boosting algorithms capable of discovering ensembles of complementary base models with much better out-of-sample performance compared to individual models [10]–[13]. The core reason of such robustness is utilization of low-complexity base models estimated one at a time and deterministic iterative approach where initial discovery of the best-on-average model is followed by additions of models focused on finer and more challenging data patterns that were missed by previous models [14]. Therefore, similar to DNN, boosting takes advantage of hierarchical knowledge representation and independent training of the model components. This makes boosting one of the alternatives to DNN-based DL.

Boosting is one of the most powerful machine learning approaches with proven success in many practical applications [10]–[18] and impressive track record of winning in various challenges/competitions on real-world machine learning problems such as Kaggle (www.kaggle.com) and others. Performance of boosting-based and similar ensemble learning solutions is often very comparable or just

Manuscript received August 5, 2018; revised October 16, 2018. This work was supported in part by the Grant of President of Russian Federation for young scientists No. MK-1896.2017.9 (contract No. 14.W01.17.1896-MK).

V. Gavrishchaka is with the West Virginia University, Physics Department, Morgantown, WV 26506 USA (e-mail: gavrishchaka@gmail.com).

Z. Yang and R. Miao are with the Applied Quantitative Solutions for Complex Systems (www.aqscs.com), Falls Church, VA 22041 USA (e-mail: zhenyiy@gmail.com, miaoxuliang@gmail.com).

O. Senyukova is with the Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Moscow 119991 Russian Federation (e-mail: olsen222@yandex.ru).

doi: 10.18178/ijmlc.2018.8.6.744

slightly inferior to the best DNNs, e.g., [20], [21]. Since discovery of boosting-based solution may often be operationally simpler and less greedy on data and computational resources, there are legitimate arguments in favor of choosing boosting rather than DNN in certain applications, e.g., [21]. However, many hybrid approaches try to combine the best features of boosting and DNNs rather than choosing just one of them. The most obvious combination is adopting NNs (with potentially different feature subsets) as base models in the boosting iterations [22]–[24]. Alternatively, boosting can be successfully adopted in the training of a single DNN [25]. Yet another type of combination is using boosting on DNN outputs for interpretation of the observed results and/or further performance improvement [20].

Generic boosting with stump or other low-dimensional classification/regression tree as base model [13], [26] may be operationally simpler than DNN in some cases. However, without additional constraints and guidance based on application domain knowledge, both approaches are pure data-driven and require large training data sets for effective discovery of useful and stable models. Domain-expert models/rules obtained by deeper understanding of the considered application scope could play a key role in cases with severe incompleteness of training data because of natural dimensionality reduction and usage of domain-specific constraints, e.g., [14], [17]. However, such simplified models are often biased and not capable to cover all possible regimes. On the other hand, comprehensive incorporation of this domain knowledge into standard DNN-based DL or generic boosting frameworks is problematic, except for straightforward guidance in factor selection.

Previously we have proposed application of boosting-like algorithms for effective utilization of all available domain knowledge (e.g., analytical and other parsimonious models) via discovery of compact ensembles of complementary low-complexity components (models) [14], [17]–[19]. This approach can tolerate significant data incompleteness and significantly increase accuracy of individual base models as was demonstrated in cardiac diagnostics [18] and in gait-based detection of neurological abnormalities [19].

While such boosting ensembles are compact, they could effectively utilize all complementary domain-expert knowledge not just best-on-average models [17]–[19]. Therefore, these ensembles can be considered as low-dimensional representations of the considered problem with particular objective (unlike generic unsupervised approaches) that could be further used in more flexible frameworks such as DNNs. Such combination may further increase model accuracy by uncovering more subtle patterns such as non-linear mixed terms that were not fully explored by boosting formulation limited to linear combination of complementary models.

Here we argue that hybrid DL framework with auto-encoders replaced by components discovered by boosting followed by supervised DNN for classification could be effective and potentially much more tolerant to data incompleteness compared to pure DNN-based DL. Illustrations based on real cardio data from www.physionet.org are presented.

II. Deep Learning Based on Neural Networks

Many properties of NNs have been discovered well before current resurgence of interest in these algorithms in the form of DL and DNNs. For example, formal mathematical results of NNs universality and their capabilities have been proven by Kolmogorov and Cybenko [27], [28]. Cybenko's theorem states that feed-forward NN with just one-hidden layer and sigmoid activation function is capable of approximating uniformly any continuous multivariate function to any desired degree of accuracy [28]. However, these results do not provide any direct recipes for finding optimal NN for any given problem and training data.

Based on Cybenko's theorem, optimal NN with good approximation should exist for any problem that meets reasonable continuity requirements. However, multi-factor nature of the majority of practical problems leads to the set of challenges that are collectively called the curse of dimensionality [29]. In the context of NN, this challenge is due to large number of weights and complex error surface with many local minima [29]. A direct global optimization of NN weights for avoiding local minima cannot solve the problem because of high-dimensionality of the problem, which is prohibitive to any stochastic or heuristic optimization algorithms including Genetic Algorithms (GA). Only after iterative back-propagation (BP) algorithm for training NNs with any number of hidden layers proposed in [30], many practical NN-based applications emerged.

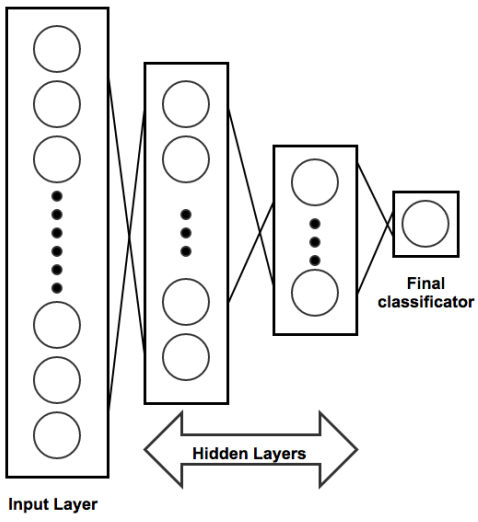


Fig. 1. Schematic diagram of standard MLP based on supervised training.

However, while BP was routinely and successfully used for NN training in many practical situations, discovery of optimal NN in each particular application was still facing many serious challenges without a single universal solution. Many problems such as vanishing or exploding gradients are limitations of BP algorithm and can be encountered in many NN architectures including well-known multi-layered perceptron (MLP) [1], [2], [29]. Some NN types may provide very powerful modeling framework but are especially hard to train in practice. For example, while recurrent NNs (RNN) could potentially find the best solutions in problems dealing with time series and general sequence forecasting, the training algorithm, back-propagation through time (BPTT), could be notoriously unstable in practice [31], [32].

Active research efforts to resolve or alleviate these limitations of NN-based frameworks and machine learning algorithms in general, resulted in development of modern DNN-based DL approaches [1], [2]. Widespread adoption of DL frameworks began after 2012 when AlexNet (convolutional DNN) significantly outperformed other machine learning approaches in ImageNet Large Scale Visual Recognition Challenge [3]. This result facilitated explosive growth of DNN-based applications in computer vision, bioinformatics, healthcare, fundamental sciences, business and other areas [1], [2], [20].

DNNs are often regarded just as multi-layered NNs which were made available for real-world applications because of possibility to train them with modern computing resources such as massively parallel GPU-based systems (www.nvidia.com). However, the main advantage of DL, capable of alleviating many existed issues, comes from the structured approach to DNN training and hierarchical representation which can be outlined as follows [1], [2], [5], [6].

DNN-based DL is not just NN with large number of hidden layers, it is important paradigm realizing importance of hierarchical representation of data with increasing degree of abstraction [1], [2], [5], [6]. This paradigm is not new. For example, in fundamental sciences, theoretical and simulation frameworks are often focused on different spatiotemporal scales and account for interaction (energy flow) across these scales, e.g., [9]. In traditional machine learning (ML), process of feature selection could often include such hierarchical representations without explicit formalization. Boosting-like ensemble learning is an example of intrinsically hierarchical algorithm. It starts from global scale classification/regression model at first iteration and focuses on more detailed modeling of sub-populations and sub-regimes in subsequent iterations [10]–[12], [14].

Although NN-based implementation of DL paradigm is not the only choice, DNN provides universal framework for modeling complex and high-dimensional data. Especially attractive feature of DNN approach is the capability of covering all stages of data-driven modeling (features selection, data transformation, and classification / regression) within a single framework, i.e., ideally, practitioner can start with raw data in the domain of interest and get ready-to-use solution [1], [2].

The key differences between standard multi-layered NN and DNN-based DL are illustrated in Fig. 1 and 2. As an example of a standard NN framework, schematic MLP diagram is shown in Fig. 1. In this case, input features/factors presented to NN in the first layer are assumed to be already selected outside NN by other means ranging from simple correlation analysis to different flavors of principal component analysis (PCA) and other statistical and machine learning tools, e.g., [26]. Once inputs are chosen, one can start supervised training of MLP using BP algorithm. In this training procedure, all weights from all layers are updated at each BP iteration or epoch [29], [30].

The obvious limitation of this standard NN framework is absence of universal approaches to feature selection and dimensionality reduction that would be a self-consistent part of the framework itself and applicable in any domain of interest. Large dimensionality of inputs directly translates to large number of weights. Since weights of all layers are updated simultaneously, already mentioned problems of large number of hard-to-avoid local minima on the multi-dimensional error surface, vanishing and/or exploding gradients and related problems are easily encountered in many practical applications.

DNN-based DL alternative to standard MLP is schematically shown in Fig. 2. The obvious difference from figure 1 is additional set of layers before the actual MLP layers for classification / regression. These additional layers effectively perform generic feature selection and dimensionality reduction via unsupervised pre-training, filtering and input transformations [1], [4]–[6]. In some cases this pre-processing may include domain-specific set of filters and transformations such as in CNN-based DL for image recognition [3], [7]. However, the most generic application-independent approach is based on auto-encoders as illustrated in Fig. 2.

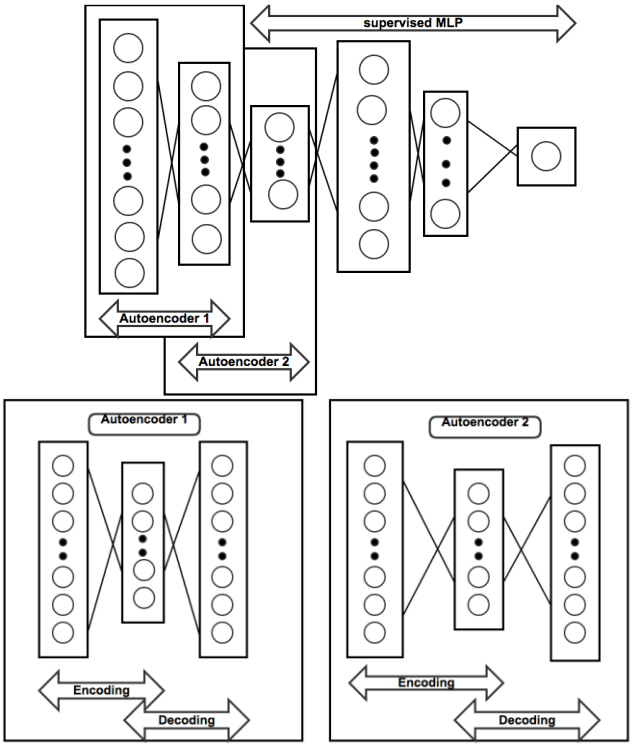


Fig. 2. Schematic diagram of DNN-based DL framework with stacked auto-encoders for unsupervised pre-training followed by standard MLP for classification or regression.

Auto-encoder in its basic form is equivalent to MLP with output layer equal to input layer [4], [5]. The training is based on standard BP used in supervised MLP training. The only difference is that input features are presented at both input and output layers during training, i.e., NN builds representation of its input in hidden layer(s) (encoding process) and then tries to recover original input from this representation (decoding process) as schematically shown in Fig. 2. Since only inputs are used in training, effectively, it is unsupervised learning. Typically, number of nodes in hidden layer(s) is significantly less then number of inputs. In this case, auto-encoder discovers compact representation of the original input information, i.e. performs generic dimensionality reduction [4]–[6]. However, if objective is to discover sparse

representations uncovering complex non-linear dependencies (patterns), then size of hidden layer is made larger than number of inputs. In the final NN, only encoding layers of auto-encoders are used as shown in Fig. 2.

Unsupervised pre-training of DNN using auto-encoders or other approaches is even more important is applications with large amount of unlabeled data but more limited availability of labeled data which is often the case. Indeed, standard supervised learning would use only labeled data, while information contained in the unlabeled data is ignored. Unsupervised pre-training is capable to discover rich set of patterns and representations from unlabeled data. After that DNN could be further fine-tuned via supervised training using available labeled data.

Thus, while NN structure in standard MLP and DL approaches may look the same, the key difference of true DL is that NN is trained layer-by-layer which leads to much more robust results and alleviates potential overfitting. First set of layers (e.g., auto-encoders) are trained in unsupervised fashion with ability to use most of the data (labeled and unlabeled). Then, MLP classifier is trained using usual supervised learning, while weights from the first set of layers are kept constant. Finally, one could choose to fine-tune all NN layers with supervised training on labeled data.

Important concept of layer-by-layer learning in DNNs goes well beyond just two major groups of layers, i.e. with unsupervised (e.g., auto-encoders) and supervised (e.g., standard MLP) learning. This allows further alleviation of often encountered problems due to data incompleteness. For example, while one can train single auto-encoder with multiple hidden layers, in practice, this approach would have serious problems if data are limited. Therefore, often used alternative is a stack of shallow auto-encoders (e.g., each with only one hidden layer) that are trained one at a time [6]. Example in Fig. 2 shows a stack with two such auto-encoders.

Another robust technique of layer-by-layer training is transfer learning with many practical applications in image recognition and other fields [7], [8], [33]. For example, millions of images in hundreds of categories are available for DNN training. However, one may have just a few hundred images in the domain of interest such as medical imaging for particular abnormality [7], [8]. In this case, NN is first pre-trained on available categories not directly related to problem of interest. Then one could keep weights constant in majority of initial layers and train just a few last layers (in MLP) on available medical images. This is transfer learning, since we transfer majority of patterns learned in the domain with large data set (i.e., abstract image descriptors) to domain with small data set. Only small fraction of final layers gets updated. Depending on the data availability for the actual problem, one may increase or decrease number of updated layers (weights). In the extreme case of very limited data set, one can even replace MLP layers with simpler model (i.e., logit regression or support vector machine).

However, severe data limitations in the context of problem dimensionality and/or absence of relevant problem for transfer learning can still drastically reduce key advantages of DNN-based DL. For example, pure data-driven auto-encoders dealing with high-dimensional input data require large amount of data for effective operation.

Existing domain-expert models/rules obtained by deeper understanding of the considered domain could play a key role in applications with severe incompleteness of training data due to natural dimensionality reduction and usage of domain-specific constraints. However, such simplified models are often biased and not capable to cover all possible regimes. On the other hand, comprehensive incorporation of this domain knowledge into standard DNN-based DL is problematic, except for straightforward guidance in factor selection. In the next section we outline a novel hybrid framework combining boosting applied to domain-expert knowledge and DNNs. This framework can potentially tolerate severe data limitations and effectively leverage advantages of existing domain-expert knowledge, boosting-based ensemble learning and DNNs.

III. Hybrid Deep Learning Frameworks Tolerant to Data Incompleteness

DNN-based DL frameworks combine ultimate flexibility for data modeling with hierarchical representations, unsupervised pre-training, transfer learning and overall layer-by-layer training which are all crucial for discovery of viable models even when data are incomplete and very complex [1], [2], [4]–[7], [33]. However, operationally, DNNs training and optimization could be very challenging in practice due to large number of hyper-parameters ranging from specific parameters of training algorithm such as learning rate to NN topology such as number of layers in each NN component (unsupervised and supervised) and number of nodes in each layer.

There are no universal recommendations for choosing optimal hyper-parameter set in each particular application. While there are rigorous mathematical results that guaranteed existence of optimal NN in each particular case [27], [28], the finding of such optimal DNN is mostly based on empirical considerations. Partial theoretical understanding of the origins of DNN-based DL success just began to emerge, e.g., [34]. Therefore, discovery of optimal DNN and its training could be very computationally intensive and unintuitive. Also, in the case of serious data incompleteness, adopting domain-expert knowledge could be critically important. However, comprehensive incorporation of domain-specific knowledge into standard DNN-based DL is problematic, except for straightforward guidance in the initial factor/feature selection.

However, alternative machine learning algorithms such as different flavors of boosting combine key advantages of DNNs such as hierarchical data representations and iterative component-wise learning with operational simplicity and ability of direct incorporation of domain-expert knowledge [10]–[14], [17], [21]. Also, performance of boosting-based models is often comparable to that of DNNs, e.g., [20], [21].

Adaptive boosting [10]–[12], [14], [26] combines many desirable features and is very distinct from the majority of ensemble learning algorithms, such as bagging and other "random sample" techniques, which can reduce only the variance part of the model error. Boosting can reduce both bias and variance [10]–[12], [26]. Boosting-based models demonstrate very good out-of-sample accuracy and stability

even in cases with limited training data due to intrinsic property of margin maximization during training.

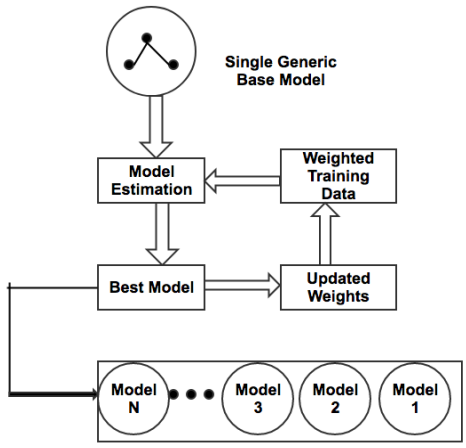


Fig. 3. Schematic diagram of generic boosting algorithm with decision stump as base model.

A typical boosting algorithm such as AdaBoost [10,26] for the two-class classification problem (+1 or -1) consists of the following steps:
following steps:

$$w_n^1 = 1/N\,, \tag{1}$$

$$\varepsilon_t = \sum_{n=1}^{N}\left(w_n^t I(-y_n h_t(x_n))\right), \tag{2}$$

$$\gamma_t = \sum_{n=1}^{N}\left(w_n^t y_n h_t(x_n)\right), \tag{3}$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1+\gamma_t}{1-\gamma_t}\right) - \frac{1}{2}\ln\left(\frac{1+\rho}{1-\rho}\right), \tag{4}$$

$$w_n^{t+1} = w_n^t \exp\left(-\alpha_t y_n H_t(x_n)\right)/Z_t\,, \tag{5}$$

$$H(x) = \sum_{t=1}^{T}\alpha_t h_t(x) / \sum_{t=1}^{T}\alpha_t\,. \tag{6}$$

Here N is the number of training data points, x_n is a model input value of the n-th data point and y_n is class label, T is the number of iterations, $I(z) = 0$ ($z<0$), $I(z)=1$ ($z>0$), w_n^t is the weight of the n-th data point at t-th iteration, Z_t is normalization constant, $h_t(x)$ is the best model at t-th iteration, ρ is a regularization constant, and $H(x)$ is the final combined model (meta-model).

Boosting starts with equal and normalized weights for all training data (step 1). Base classifiers $h_t(x)$ are trained using weighted error function ε_t (step 2). The best $h_t(x)$ is chosen at the current iteration. The data weights for the next iteration are computed in steps (3)–(5). At each iteration, data points misclassified by the current best model (i.e., $y_n h_t(x_n) < 0$) are penalized by the weight increase for the next iteration. AdaBoost constructs progressively more difficult learning problems that are focused on hard-to-classify patterns defined by the weighted error function (step 2). The final meta-model (6) classifies the unknown sample as class +1 when $H(x) > 0$ and as -1 otherwise.

From the above description, it is clear that typical boosting algorithm is based on utilization of low-complexity base models estimated one at a time and deterministic iterative approach where initial discovery of the best-on-average model is followed by additions of models focused on more challenging data patterns/regimes that were poorly modeled in previous iterations [10], [14], [26]. Therefore, similar to DNN, boosting takes advantage of hierarchical knowledge representation and independent training of the model components.

In pure data-driven approaches a typical choice of the base model is decision stump (i.e., one-level decision tree) as shown in Fig. 3 where boosting procedure is schematically represented. In this case just one generic and application-independent base model is used. The final model is multi-level tree constructed over many boosting iterations. However, the out-of-sample performance of such large tree discovered by boosting is much better than that of the same tree obtained by simultaneous global optimization of the parameters of multi-level tree [26].

Generic boosting and its various extensions such as XGBoost [13] often demonstrate superiority over other algorithms in many applications and competitions. Its performance also often approaches that of DNNs. However, since discovery of boosting-based solution may often be operationally simpler, there are legitimate arguments in favor of choosing boosting rather than DNN in certain applications [21]. However, many hybrid approaches try to combine the best features of boosting and DNNs rather than choosing just one approach and discarding the other.

The most obvious combination is adopting DNNs (with potentially different feature subsets) as base models in the boosting iterations [22]–[24]. Alternatively, boosting can be successfully adopted in the training of a single DNN [25]. Yet another type of combination is using boosting on DNN outputs for interpretation of the observed results and/or further performance improvement [20].

Generic DNNs and boosting algorithms as well as most of their combinations are flexible but often pure data-driven approaches which require significant amount of training data for discovery of accurate and stable models. Domain-expert models and other existing knowledge obtained by deeper understanding of the considered domain could play a key role in applications with severe incompleteness of training data due to natural dimensionality reduction and usage of domain-specific constraints. However, such simplified models are often biased and not capable to cover all possible regimes. On the other hand, comprehensive incorporation of this domain knowledge into standard DNN-based DL is problematic, except for straightforward guidance in factor selection.

Similarly, boosting algorithms in their original form, such as shown in Fig. 2, are also not suitable for generic incorporation of variety of domain-expert knowledge such as analytical models, rules and constraints. However, boosting can be applied to the pool of the well-understood and low-complexity domain-expert models to produce an interpretable ensemble of complementary base models with significantly higher accuracy and stability as suggested in [14], [17]–[19], [35], [36]. Schematic of such algorithm is shown in Fig. 4.

Unlike generic boosting algorithms (such as in Fig. 3), the pool of base models could include any number of parameterized domain-expert and/or other low-complexity models (see Fig. 4) [14], [17]. At each boosting iteration, all models from this pool are optimized one at a time according to the weighted error function (2) and the best model is added to the ensemble. Such procedure can test and utilize complementary value of any number of available domain-expert models without overfitting. Also, proper parameterization could allow discovery of many complementary models even from a single domain-expert model. Unlike boosting with generic and simple tree-based model, domain-expert base models could already capture significant number of regimes and impose important application-specific constraints. This facilitates discovery of compact model ensembles which combine high accuracy with interpretability since well-understood base models are used [14], [17].

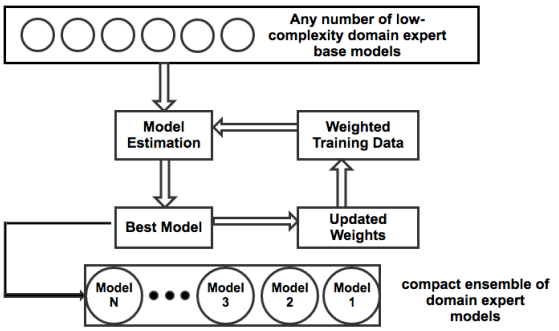


Fig. 4. Schematic diagram of boosting-like algorithm with base models inspired by existing domain-expert knowledge.

The described advantages of boosting-based frameworks for incorporation of domain-expert knowledge into the final model are not directly applicable to standard DNN-based architectures. For example, while boosting can work with unlimited number of potential domain-expert models, since all of them are estimated and added to ensemble one a time, using all of them as inputs to DNN even with pre-training layers could easily become impractical for limited training data sets (curse of dimensionality). In this case, one could be forced to consider only limited number of best models and underutilize complementary value of models, which offer expertise in certain regimes.

Alternatively, when number of domain-expert models is limited, DNN does not offer any direct means of constructing additional (complementary) models with different set of parameters from a given model. DNN can only use original set of domain-expert models as inputs, which could eliminate potentially important factors. Usage of raw data inputs in addition or instead of domain-expert models could easily encounter problem of limited training data.

However, even though boosting seems to be more natural for incorporation and enhancement of the domain-expert knowledge, its flexibility is still inferior to DNN-based DL. After all, boosting finds weighted linear combination of models. While such combination is capable to represent very complicated (non-linear) decision boundaries in classification problems, it may still miss important mixed terms that could be easily captured by DNN representation. Therefore, combination of the two approaches in capturing the most of domain-expert knowledge and lowering requirements for training data seems natural.

Here we propose one of such combination where auto-encoder (pre-training) layers of DNN (see Fig. 2) are replaced by compact ensemble of domain-expert models discovered by boosting (see Fig. 4). This allows lowering dimensionality of the problem without requirement of large training data as in case of auto-encoders usage. On the other hand, supervised training of subsequent DNNs layer may further increase boosting model accuracy by incorporating mixed components to the original linear combination of models in the boosting ensemble. If number of important models in the ensemble is too large, auto-encoders could be also applied to this ensemble if enough training data is available. In any case, requirement on training data would be significantly less compared to direct usage of raw data in pure DNN-based DL.

It should be noted that proposed usage of components from boosting ensemble is very different from stacking-like combination of models via NN or simpler algorithm. Indeed, stacking combines several complete models with comparable performance to get additional and often small gain in performance. In our case, the complete model is boosting ensemble. As in any ensemble model, only final aggregated output is used for final prediction. However, previously, we have demonstrated utility of direct usage of information encoded in the boosting components (base models) which, by construction, are experts in particular regimes or sub-populations [18], [35]. This approach, called ensemble decomposition learning (EDL), has been shown to be effective in rare states/events description and forecasting [18], [35]. Here we propose using this implicitly encoded representation of sub-regimes and sub-populations based on significantly enhanced domain knowledge as input to DNN for further training towards objective of interest.

IV. Application Examples

The proposed framework for leveraging domain-expert knowledge, boosting and DNNs is generic and could be especially attractive in applications with limited data. In this section, we provide illustrative examples supporting possibility of synergetic combination of boosting-based discovery of compact ensembles from models inspired by domain knowledge and DNNs. More detailed examples in wider scope and comparison with other modeling frameworks will be discussed elsewhere.

Combination of physics-based and general analytical models with machine learning frameworks is known to be effective in variability analysis of physiological time series [17]–[19], [35], [36]. One of the well-known applications of this methodology is heart rate variability (HRV) analysis approved as one of the modalities for cardiac diagnostics [37]. Compared to traditional ECG analysis of waveforms, HRV metrics computed from time series of beat-to-beat (R-R) intervals are much more tolerant to noise and capable of detecting cardiac and non-cardiac (e.g., psychological) abnormalities lacking well-defined ECG waveform patterns [17], [18], [38]. HRV analysis is often based on complexity

measures inspired by theoretical results in nonlinear dynamics (NLD) and by spectral metrics heavily used in science and engineering for time series analysis [17], [37], [39]–[43].

However, the accuracy and stability of such variability measures tend to decrease significantly when applied to shorter data segments [17]. This limitation diminishes the predictive capability of these measures for early detection of both short-lived precursors of emerging physiological regimes and abnormalities with transient patterns. Previously, we have demonstrated that performance of HRV indicators dealing with short time series could be significantly improved through optimal combination of complementary complexity measures using boosting [17]–[18], [35], [36].

The well-known NLD-inspired HRV metrics are based on detrended fluctuation analysis (DFA) [39], [40], multi-scale entropy (MSE) [41], and multi-fractal analysis (MFA) including MFA extension of DFA [42]. DFA was proven to be useful in revealing the extent of long-range correlations in time series including HRV applications [39], [40]. First, the investigated time series of length N is integrated. Next, the integrated time series is divided into n boxes. All boxes have the same length. In each box, a least-square line is fit to the data with y coordinate denoted by $y_n(k)$ (representing the trend in that box). Finally, the integrated time series, $y(k)$, is detrended as follows:

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[y(k) - y_n(k)\right]^2}\,. \tag{7}$$

A linear relationship on the plot of log $F(n)$ vs. log n indicates power law (fractal) scaling characterized by a scaling exponent β (slope of the fitted straight line) which is used as physiological state indicator.

Multi-scale entropy (MSE) method [41] has been introduced to resolve limitations of traditional single-scale entropy measures. First, a coarse-graining process is applied to the original time series, x_i. Multiple coarse-grained time series are constructed by averaging the data points within non-overlapping windows of increasing duration, τ:

$$y_j^{(\tau)} = \frac{1}{\tau}\sum_{i=(j-1)\tau+1}^{j\tau} x_i\,. \tag{8}$$

Here, τ represents the scale factor and j=1…N/τ. The duration of the coarse-grained time series is N/τ. Next, entropy is calculated for each time series and plotted as a function of the scale factor. Different signatures of this curve, including originally suggested entropy difference between two scales [41], can serve as HRV and other physiological indicators.

Spectral HRV indicators based on frequency-domain analysis are often superior in accuracy and stability to the time-domain linear indicators. One of the widely accepted indicators of this type is a power spectrum ratio of the low-frequency band (0.04-0.15 Hz) to the high-frequency band (0.15-0.4 Hz) [37], which we will refer to as LFHF indicator. In certain regimes, the accuracy of such power spectrum indicators could be comparable to the best NLD approaches.

A natural choice of base models within boosting framework are low-complexity base classifiers, where each of the classifiers uses just one complexity measure, β_i, out of several available choices [17]:

$$y = h(\beta_i[p_i], \gamma) \tag{9}$$

Here γ is a threshold level (decision boundary) and pi is a vector of adjustable parameters of the chosen measure. In our case, β_i may correspond, for example, to either DFA scaling exponent, slope of MSE curve, or power spectrum ratio. Applying boosting steps to a set of such base classifiers (9) with different measures β_i and optimizing over (p_i, γ) on each boosting iteration, we can obtain a compact ensemble of measures with significantly better accuracy and stability.

Previously, we have demonstrated that boosting-based combination of DFA, MSE, and LFHF indicators parametrized according to (9) and optimized one at a time at each boosting iteration can significantly increase accuracy of cardiac abnormality detection even when short R-R segments of just several minutes are used [17]–[18], [35], [36]. Here we reproduce the main features of these results on similar (but expanded) data set and illustrate that addition of DNN to the boosting-based framework could further improve accuracy of the generic normal-abnormal classification/ranking for multiple cardiac abnormalities.

Analysis presented in this section is based on real-patient ECG data from http://www.physionet.org. We used long R-R records (up to 24 hours each) from 52 subjects with normal sinus rhythm, 27 subjects with congestive heart failure (CHF), 84 subjects with long-term atrial fibrillation (LTAF), and 12 subjects from Sudden Cardiac Death (SCD) database who had sustained ventricular tachyarrhythmia and most had an actual cardiac arrest. Additionally, we used more than 100 30-minute records from 48 subjects with paroxysmal atrial fibrillation (PAF) and up to 30-minute records for each of 47 subjects with different types of arrhythmia. We have also added 78 intervals (each of 30 min) from patients with supraventricular arrhythmias to expand the arrhythmia data set. It should be noted that, while various cardiac abnormalities can be accompanied by arrhythmia, a separate arrhythmia sample, considered here, represents arrhythmia-only condition.

Here we use collection of 256-beat R-R segments obtained by 128-beat shifts (i.e., half overlapped) from R-R time series described above. The total number of R-R segments used for calculation of DFA, LFHF, and MSE indicators is more than 1.35×105, among which about 3.6×104 are data from normal subjects. For training we used balanced set with 3.6×104 of 256-beat segments (i.e., equal number of segments from normal subjects and patients with different cardiac abnormalities), which is just slightly above 25% of all data. 1.8x104 segments are from normal subjects (i.e. less than half of data in this category) and 6×103 are from each of the following categories: CHF (i.e. less than 30%), LTAF (i.e., less than 9%) and SCD (~ 90%). All reported AUC metrics are computed on the full set of 1.35×105 samples (which are mostly out-of-sample). It should be noted that training set size could be reduced even further, since we did not observe any signs of overfitting due to usage of parsimonious analytical indicators as base models.

As shown previously, for all specific abnormality types and

generic normal-abnormal classification, DFA and LFHF indicators always demonstrated significantly better performance compared to MSE [18]. Application of boosting to parameterized DFA and LFHF base indicators did not show any noticeable gain compared to single DFA or LFHF model, which indicated insufficient variability of these models even after parameterization. However, when MSE-based indicators were included as base models in addition to DFA and LFHF measures, boosting was capable of discovering ensembles with classification/ranking accuracy more than 10% higher than that of single DFA or LFHF model e.g., [18], [36]. Here ranking or differentiation capability is measured by the full or partial area-under-curve (AUC) metric applied to Receiver Operating Characteristic (ROC) curve. Thus, in this setting, the main boosting-induced gain is due to complementary of original measures rather than additional optimization of the parameterized measures.

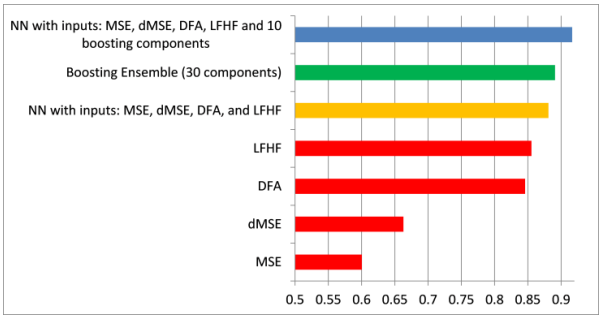


Fig. 5. AUC of single measures (red), NN with standard measures as inputs (orange), boosting ensemble with parameterized single measures as base models (green), and NN with standard measures and boosting components as inputs (blue).

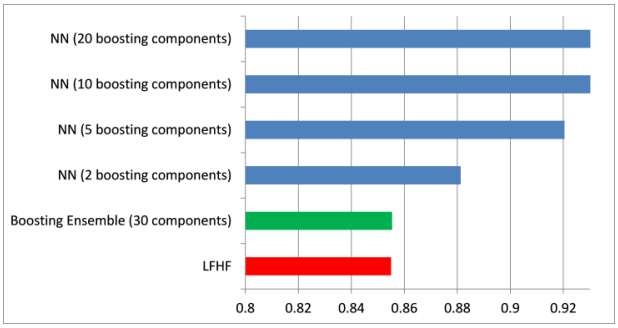


Fig. 6. AUC of a single LFHF measure with standard parameters (red), boosting ensemble with LFHF as only base model (green), and NN with LFHF boosting components as inputs (blue).

Illustration in Fig. 5 shows that combination of already existing boosting-based results and DNN could further improve accuracy of the normal-abnormal classification and produce results better than those obtained from boosting or DNN alone. Here we compare AUC metrics of the individual measures (DFA, LFHF, dMSE and MSE) calculated on standard set of parameters [37], [39]–[43], [17] with NN and boosting models based on these indicators as well as combined boosting and NN model.

Results in Fig. 5 confirm intuitively expected accuracy ranking of different models. First, NN with 4-indicator input is capable to increase accuracy of the best single indicator by mixing complementary benefits of each such indicator. Boosting not only uses these four indicators with standard parameters, but also finds optimal parameters for these indicators at each boosting iteration, which enhances capabilities for finding proper complementary models and increase ensemble accuracy. This could explain better performance of boosting compared to NN. However, difference in performance is small since additional variability of indicators due to parameterization is limited. One could expect more pronounced performance enhancement in cases with higher variability of the parameterized base models. Although in this case we show boosting ensemble with 30 components, the main boosting effect is already achieved after 10-15 iterations, i.e., with much more compact ensemble.

Finally, as shown in Fig. 5, the highest AUC value is achieved by combination of boosting and NN. In this case, besides four indicators with standard parameters we add first 10 indicators from boosting ensemble to NN inputs. These additional indicators have parameters optimized for detection of particular subset of patterns. Therefore, they add complementary information in addition to indicators with standard parameter set. This explains better performance compared to original NN with four inputs. However, accuracy is also higher compared to boosting ensemble. This demonstrates that NN optimally adds mixed terms into the final model which improves performance of the original boosting ensemble based on linear combination of base models.

Even more encouraging and less expected result of the synergetic combination of boosting and NN is presented in Fig. 6. Here we used boosting with just one base model – parameterized LFHF indicator. Therefore, at each boosting iteration only LFHF indicator with different set of parameters was added to the ensemble. As already mentioned, while LFHF and DFA measures are often the best single indicators, their variability remain limited after parameterization. Therefore, when set of base models is restricted to LFHF (or both DFA and LFHF), effect of boosting remain very limited as shown in Fig. 6, where 30-component boosting ensemble shows almost the same performance as single LFHF indicator.

However, boosting is known for its ability for continuous margin increase even when formal training error stops decreasing. This helps boosting out-of-sample performance. In our case, boosting also tries to find complementary set of LFHF indicators with different parameter sets, even though in-sample and out-of-sample performance of boosting ensemble do not show noticeable increase. This could be due to intrinsic boosting limitation of using linear combination of base models. However, when just first 5-10 components from boosting ensemble are used as inputs to NN, AUC can be increased by up to 10% compared to single LFHF indicator or boosting ensemble (see Fig. 6). Once again, NN was able to significantly enhance performance through flexible non-linear mixing of components discovered by boosting.

Optimal hyper-parameters used in NN-based solutions presented in this section (Fig. 5 and 6) were found by simple search on rather coarse-grain parameter grid, which suggests that further fine-tuning of hyper parameters and performance improvements are possible. Here we used H2O.ai (www.h2o.ai) implementation of feed-forward NN with rectifier activation function in hidden layers. Our final choice was NN with two hidden layers and decreasing number of nodes in proportion of 3:1. 50% dropout rate was chosen. It

should be noted that we did not observe any change in model performance ranking presented in Fig. 5 and 6, when NN architecture and learning parameters were varied around this quasi-optimal set of hyper-parameters.

Further research on the proposed combination of boosting and DNNs is warranted and results for a wider scope of applications will be presented elsewhere. We will also adopt multi-objective optimization framework for better search of NNs hyper-parameters which already shows encouraging preliminary results.

V. Conclusion

We have proposed combination of boosting-like algorithms and DNNs in applications with limited training data and existing domain-expert knowledge in the form of analytical and other parsimonious models or indicators. In particular, we have argued that hybrid DL framework with auto-encoders replaced by components discovered by boosting followed by supervised NN could be more tolerant to data incompleteness compared to pure DNN-based DL. We have illustrated that, in application dealing with detection of multiple cardiac abnormalities from short time series of beat-to-beat (R-R) intervals, boosting, DNNs and existing complexity measures can be synergistically combined to achieve significantly better performance compared to each individual technique. Further research and wider scope applications of the proposed framework are warranted.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, 436-444, May 2015.

[2] L. Deng, and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, nos. 3-4, 197-387, June 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *the Neural Information Processing System*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. pp. 1097-1105, vol. 2, 2012.

[4] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, 504-507, July 2006.

[5] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, 625-660, Feb. 2010.

[6] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proc. International Conference on Acoustics, Speech, and Signal Processing*, 2013, pp. 3377-3381.

[7] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Trans. Med. Image*, vol. 35, no. 5, 1285-1298, May 2016.

[8] S. Christodoulidis, M. Anthimopoulos, L. Ebner, A. Christe, and S. Mougiakakou, "Multisource transfer learning with convolutional neural networks for lung pattern analysis," *IEEE J. Biomed. Health Inform*, vol. 21, no. 1, 76-84, Jan. 2017.

[9] S. Banerjee and V. V. Gavrishchaka, "Multimoment convecting flux tube model of the polar wind system with return current and microprocesses," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 69, no. 16, 2071-2080, Nov. 2007.

[10] R. E. Schapire, "The design and analysis of efficient learning algorithms," Ph.D. dissertation, Massachusetts Institute of Technology Cambridge, MA, 1992.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, 337-407, Apr. 2000.

[12] J. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, 1189-1232, Oct. 2001.

[13] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785-794.

[14] V. V. Gavrishchaka, "Boosting-based frameworks in financial modeling: Application to symbolic volatility forecasting," in *Econometric Analysis of Financial and Economic Time Series Advances in Econometrics*, T. B. Fomby, D. Terrell, Eds. Bingley, UK: Emerald Group Publishing Limited, 2006, vol. 20, part 2, pp. 123-151.

[15] K. Wohlrabe and T. Buchen, "Assessing the macroeconomic forecasting performance of boosting: Evidence for the United States, the Euro area and Germany," *Journal of Forecasting*, vol. 33, no. 4, 231-242, Mar. 2016.

[16] A. Agapitos, A. Brabazon, and M. O'Neill, "Regularised gradient boosting for financial time-series modelling," *Computational Management Science*, vol. 14, no. 3, 367-391, July 2017.

[17] V. V. Gavrishchaka and O. V. Senyukova, "Robust algorithmic detection of cardiac pathologies from short periods of RR data," in *Knowledge-Based Systems in Biomedicine and Computational Life Science, Studies in Computational Intelligence*, vol. 450, T. D. Pham, L.C. Jaim, Eds. Heidelberg, Germany: Springer, 2013, pp. 137-153.

[18] O. Senyukova, V. Gavrishchaka, M. Sasonko, Y. Gurfinkel, S. Gorokhova, and N. Antsygin, "Generic ensemble-based representation of global cardiovascular dynamics for personalized treatment discovery and optimization," in *Computational Collective Intelligence: 8th International Conference, Proceedings, Part I*, N. T. Nguen, L. Iliadis, Y. Manolopoulos, B. Trawinski, Eds. pp. 197-207, vol. 9875, 2016.

[19] V. Gavrishchaka, O. Senyukova, and K. Davis, "Multi-complexity ensemble measures for gait time series analysis: Application to diagnostics, monitoring and biometrics," in *Advances in Experimental Medicine and Biology*, C. Sun, T. Bednardz, T. D. Pham, P. Vallotton, and D. Wand, Eds. Cham, Switzerland: Springer International Publishing, 2015, vol. 823, pp. 107-126.

[20] Z. Che, S. Purushotham, R. Khemain, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv*:1512.03542 [stat.ML], 11 Dec 2015.

[21] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *Proc. the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 3553-3559.

[22] M. Moghimi, M. Saberian, J. Yang, L.-J. Li, N. Vasconcelos, and S. Belongie, "Boosted convolutional neural networks," in *Proc. the British Machine Vision Conference*, 2016, vol. 24, pp. 1-13.

[23] H. Schwenk and Y. Bengio, "Boosting neural networks," *Neural Computation*, vol. 12, no. 8, 1869-1887, Aug. 2000.

[24] A. J. C Sharkey, "Boosting using neural networks," in *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*, A. J. C. Sharkey, Ed. London, UK: Springer London, 1999, pp. 51-78.

[25] S. Shalev-Shwartz, "SelfieBoost: A boosting algorithm for deep learning," *arXiv*:1411.3436v2 [stat.ML], Apr. 8, 2017.

[26] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, New York, NY: Springer New York Inc, 2001.

[27] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Dokl. Akad. Nauk. SSSR*, vol. 114, pp. 953-956, 1957.

[28] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function," *Math. Control Signals Systems*, vol. 2, no. 3, 303-314, Dec. 1989.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, NY: Springer-Verlag, 2006.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, Oct. 1986.

[31] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, 1550-1560, Oct. 1990.

[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, 1735-1780, Nov. 1997.

[33] Z. Huang, Z. Pan, and B. Lei, "Transfer Learning with deep convolutional neural network for SAR target classification with limited labeled data," *Remote Sens*., vol. 9, no. 9, 907, Aug. 2017.

[34] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv*:1312.6120v3 [cs.NE], 19 Feb. 2014.

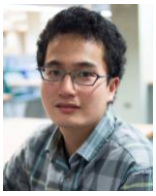
[35] O. V. Senyukova and V. V. Gavrishchaka, "Ensemble Decomposition Learning for Optimal Utilization of Implicitly Encoded Knowledge in Biomedical Applications," *Proceedings of Computational Intelligence and Bioinformatics*, 2011, pp. 69-73.

[36] O. Senyukova, V. Gavrishchaka, and M. Koepke, "Universal multi-complexity measures for physiological state quantification in intelligent diagnostics and monitoring systems," in *Proc. Biomedical Informatics and Technology*, 2014, 76-90.
[37] Task Force of the European Society of Cardiology the North American Society of Pacing Electrophysiology, "Heart rate variability: Standards of measurement, physiological interpretation, and clinical use," *Circulation*, vol. 93, no. 5, 1043-1065, Mar. 1996.
[38] O. Senyukova, V. Gavrishchaka, and K. Tulnova, 2017, "Multi-expert evolving system for objective psychophysiological monitoring and fast discovery of effective personalized therapies," in *Proc. IEEE Conference on Evolving Adaptive Intelligence Systems*, 2017, pp. 1-8.
[39] J. Belair, L. Glass, U. A. D. Haien, and J. Milton, *Dynamical Disease: Mathematical Analysis of Human Illness*. New York: AIP Press, 1995.
[40] C.-K. Peng, S. Havlin, E. H. Stanley, and A. L. Goldberger, "Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series," *Chaos*, vol. 5, 82-87, Sep. 1995.
[41] M. Costa, A. L. Goldberger, C.-K. Peng, "Multiscale entropy analysis of physiologic time series," *Physical Review Letters E*, vol. 71, 021906, Feb. 2005.
[42] D. Makowiec, A. Dudkowska, M. Zwierz, and A. Rynkiewicz, "Scale invariant properties in heart rate signals," *Acta Physica Polonica B*, vol. 37, no. 5, 1627-1639, May 2006.
[43] A. Voss, S. Schulz, R. Schroederet, M. Baumert, M., and P. Caminal, "Methods derived from nonlinear dynamics for analysing heart rate variability," *Phylosophical Transactions of the Royal Society A*, vol. 367, 277-296, Jan. 2009.

Valeriy V. Gavrishchaka received his MS and PhD degrees in computational and theoretical physics from Moscow Institute of Physics and Technology (Moscow Region, Russian Federation) and from West Virginia University (Morgantown, West Virginia, USA) in 1989 and 1996, respectively.

From 1997 to 2002 he worked as a multi-disciplinary research scientist and consultant at Science Applications International Corporation (McLean, Virginia) on a wide range of problems in plasma / space physics and space weather forecasting using physics-based models / simulations and wide range of machine learning approaches. From 2002 to 2010 he worked for several multi-billion New York based hedge funds as head of quantitative research and quantitative strategist for multi-frequency algorithmic trading. He continues to provide services in quantitative finance and AI-intensive business applications.

Dr. Gavrishchaka is also an adjunct professor of physics at Physics Department of West Virginia University and leader of multiple projects in research group for applied quantitative solutions in complex systems (www.aqscs.com). His main research interests include development and applications of novel multi-disciplinary approaches and integrated frameworks leveraging existing domain knowledge and advanced machine learning techniques. He is an author of more than 70 publications in mainstream scientific journals and referred conference proceedings that are frequently cited as summarized in his Google Scholar and Research Gate profiles.



Zhenyi Yang received his BS and MS degrees in financial mathematics from University of Michigan Ann Arbor and The Johns Hopkins University in 2011 and 2013.

Currently, he works as a financial engineer in a major financial company (Washington, DC). He is an all-round data scientist and computer science expert. He is also actively involved in several inter-disciplinary projects of the research group for applied quantitative solutions in complex systems (www.aqscs.com).

Mr. Yang's current research interest includes development and real-life applications of ensemble-based models, deep learning neural networks and other machine learning algorithms. He successfully applies novel machine learning frameworks to challenging problems in biomedicine and quantitative finance.

Xuliang (Rebecca) Miao received his MS degree of financial mathematics from Johns Hopkins University in 2015; BS degree of applied mathematics and computer science from University of California, San Diego in 2014.

Currently she is working in a major financial firm (Washington, DC) and focuses on the risk management, quantitative analysis and machine learning. She is also actively involved in several inter-disciplinary projects of the research group for applied quantitative solutions in complex systems (www.aqscs.com).

For the last five years, Ms. Miao has applied her programing and analytical skills in several fields, which include biostatistics, high-performance computing and credit risk. Her current research is focused on development and applications of novel machine learning algorithms (including various ensemble-based and deep learning frameworks) to challenging problems in biomedicine and quantitative finance.

Olga Senyukova received the M.S. degree in applied mathematics and informatics in 2008 and Ph.D. degree in 2012 from Lomonosov Moscow State University (MSU), Faculty of Computational Mathematics and Cybernetics (CMC).

She is an assistant professor in CMC MSU, head of medical image analysis group in Graphics and Media Lab. Her research interests include automatic diagnostics of diseases, segmentation of anatomical structures in medical images, personalized medicine and treatment planning.

Dr. Senyukova manages the project devoted to cardiac data analysis supported by the Grant of President of Russian Federation for young scientists. She has journal and conference proceedings papers with medical doctors from Children's Clinical and Research Institute Emergency Surgery and Traumatology directed by Leonid Roshal, Research Clinical Center of JSC Russian Railways and others, several chapters in edited books by Springer and one patent. She is a member of programming committees of several IEEE and other international conferences.