# Genetic Selection of Training Sets for (Not Only) Artificial Neural Networks

Jakub Nalepa[1,2](✉), Michal Myller[1,2], Szymon Piechaczek[1,2],
Krzysztof Hrynczenko[1,2], and Michal Kawulok[1,2]

[1] Silesian University of Technology, Gliwice, Poland
{jakub.nalepa,michal.kawulok}@polsl.pl,
{michmyl139,szympie318,krzyhry690}@student.polsl.pl
[2] Future Processing, Gliwice, Poland

**Abstract.** Creating high-quality training sets is the first step in designing robust classifiers. However, it is fairly difficult in practice when the data quality is questionable (data is heterogeneous, noisy and/or massively large). In this paper, we show how to apply a genetic algorithm for evolving training sets from data corpora, and exploit it for artificial neural networks (ANNs) alongside other state-of-the-art models. ANNs have been proved very successful in tackling a wide range of pattern recognition tasks. However, they suffer from several drawbacks, with selection of appropriate network topology and training sets being one of the most challenging in practice, especially when ANNs are trained using time-consuming back-propagation. Our experimental study (coupled with statistical tests), performed for both real-life and benchmark datasets, proved the applicability of a genetic algorithm to select training data for various classifiers which then generalize well to unseen data.

**Keywords:** ANN · Genetic algorithm · Training set selection
Classification

## 1 Introduction

Artificial neural networks (ANNs) are an established and thoroughly researched classifier applied in a wide range of real-life applications, including computer-aided diagnosis [13,28], stock market index prediction [16], chemical engineering [9], weather forecasting [1], face and skin detection [2,8], and many other pattern recognition tasks [6,22]. ANNs belong to the *supervised* classification engines which require labeled training sets to be supplied to a training procedure—commonly the back-propagation routine [26]—whose aim is to build a well-generalizing (to the unseen data) model. Collecting *high-quality* training sets (i.e., balanced, representative, and containing only correctly labeled examples) is often a fairly user-dependent[1] and expensive process. Low- or questionable-quality data provided as a training set may easily hamper the learning and

[1] As an example, there may be massive discrepancies between two equally-experienced readers segmenting a medical image [21].

S. Kozielski et al. (Eds.): BDAS 2018, CCIS 928, pp. 194–206, 2018.
https://doi.org/10.1007/978-3-319-99987-6_15

deteriorate the performance of any supervised classifier, with ANNs not being an exception [18]. Therefore, selecting reduced (or *surrogate*) training data from *difficult* datasets (i.e., very large, imbalanced or weakly-labeled) is a vital research topic and should be considered while building a processing engine for a classification task being tackled [18–20].

### 1.1 Related Work

Training set selection algorithms have been successfully applied for supervised classifiers, including—among others—support vector machines (SVMs) [12,18,19] and random forests [14]. Such algorithms are commonly divided into several groups reflecting their underlying ideas. In basic *random* training set selection techniques, reduced sets are sampled from the entire training set $\boldsymbol{T}$ without building any surrogate model of the solution space $\mathcal{S}$ [3]. Although there exist algorithms for intensifying the search in specific parts of $\mathcal{S}$ (which seem to encompass important $\boldsymbol{T}$ vectors), their convergence abilities are still very limited.

In *geometry-based* approaches, the spatial characteristics of examples[2] in $\boldsymbol{T}$ are analyzed in search of *important* feature vectors, i.e., those influencing the resulting classification model (e.g., vectors which will likely be selected as *support vectors* in SVMs). *Geometry-based* training set selection techniques are built upon an assumption that the data geometry reflects the importance of specific examples in $\boldsymbol{T}$. This information can be extracted before learning a classifier, hence "useless" vectors can be removed from the refined sets. Such methods encompass clustering-based and clustering-free approaches—the former techniques exploit unsupervised algorithms to group $\boldsymbol{T}$ data in the first step. Then, clusters are further analyzed (e.g., examples positioned inside a cluster will most likely not influence the learning process, therefore can be safely removed [18]). Similarly, *neighborhood-based* techniques traverse local relations between $\boldsymbol{T}$ examples to find distinctive vectors [4]. Geometry- and neighborhood-based algorithms are dependent on the size of $\boldsymbol{T}$ (given as $t$) which is their very important downside (they may not be applicable in real-life scenarios in which the cardinality of $\boldsymbol{T}$ easily exceeds millions). This problem is often not exposed in randomized techniques which sample a subset of $\boldsymbol{T}$ of a size selected prior to optimization.

Evolutionary algorithms (EAs), with genetic and memetic techniques (the latter being hybrids of evolutionary algorithms and local-search procedures) visibly constituting the mainstream, have recently gained interest in the context of selecting refined training sets, especially for SVMs [17,18]. In such techniques, which are often independent from $t$, a population of candidate solutions (i.e., surrogate sets) evolves in time in a biologically-inspired process. Since optimizing training sets using EAs requires learning an underlying model with the reduced set represented by each individual in the population (such approaches are called the *wrapper* techniques), they may become fairly computationally-intensive for larger sets. Nevertheless, they were shown to be outperforming other methods

[2] In this paper, we use *examples* and *vectors* of features interchangeably.

from the literature and to be able to deliver *high-quality* training sets (i.e., such datasets from which an effective classifier can be learned). Importantly, such methods are able to build surrogate models that contain vectors that would have been rejected by geometry-based approaches (e.g., positioned within groups of one-class examples), but turned to be important (and affecting the classifier performance) in the course of evolution.

Although the use of EAs for optimizing ANNs is not novel, such approaches were mainly applied for optimizing the weights, learning rules, and network topologies [7]. Evolutionary augmentation of "minimal" neural topologies was a basis of a famous NEAT algorithm (**N**euro-**E**volution of **A**ugmenting **T**opologies), which was shown to be outperforming other fixed-topology ANNs in reinforcement learning tasks. Importantly, candidate solutions get more complex in the course of optimization in NEAT, hence this technique also proved that strengthening the analogy with biological evolution is an important aspect of well-designed EAs. A very interesting research pathway involves evolving neural network ensembles [27] (importantly, such classifier committees are gaining research interest nowadays [15,23]). Such evolutionary ensembles are aimed at extracting diverse groups of ANNs which are learned separately and are able to provide an agreed classification outcome when classifying challenging datasets (which is in line with the divide-and-conquer strategy). There exist methods for evolving surrogate training sets for ANNs [5,10,24], however they have not been extensively investigated in the literature.

Reducing the size of a training set is slightly counter-intuitive and can appear simply incorrect, especially for larger ANN topologies[3] (with a huge number of neurons in hidden layers), because such networks are prone to memorizing small sets (they can be easily overfitted to $\boldsymbol{T}$). In this paper, we address this issue and carefully verify whether it is possible to train a well-generalizing ANN with reduced $\boldsymbol{T}$'s, extracted using a genetic algorithm.

### 1.2 Contribution

The contribution of this work is multifold. We build upon a genetic algorithm (GA) for evolving reduced training sets for SVMs [11], and exploit an analogous approach for extracting reduced sets for ANNs (**G**enetic **A**lgorithm for **T**raining **S**et selection, GA-TS), in which each candidate solution (an *individual*) in a population represents a surrogate set, and its fitness is quantified as the classification performance of the underlying model trained using the corresponding surrogate training set. These surrogate sets can be effectively used for learning other classifiers as well.

Although GA-TS is a wrapper technique (i.e., calculating the fitness value involves performing the full training procedure), our extensive and fairly rigorous experiments (multifold cross-validation backed up with statistical tests) showed

[3] Also in the context of recent advances in the field of deep learning, where data augmentation—being a process of extending training sets rather than reducing their size—became a critical step in designing deep network topologies [21].

that is offers decent convergence capabilities for benchmark and real-life datasets (the latter was extracted from the ECU database[4] in the context of skin detection and segmentation). In sensitivity analysis, we showed how the most important parameters of GA-TS influence the optimization process alongside the classifiers trained using such evolved surrogate training sets. The following insights can be learned from our experimental study:

- Surrogate sets evolved using GA-TS can be effectively used to improve performance of not only ANNs but other state-of-the-art classifiers as well (we investigated several classification engines known from the literature).
- ANNs of a predefined topology can be learned from small training sets, and they can generalize well to the unseen data.
- Increasing the number of generated children (beyond two) and population size in GA-TS slows down the convergence and does not offer a visible increase in the quality of extracted training sets.
- Reducing the cardinality of $\boldsymbol{T}$ helps improve all investigated classifiers, which is in line in the conclusions reported in recent works on SVMs [17].

### 1.3 Paper Structure

This paper is structured as follows. Section 2 discusses our genetic algorithm for evolving surrogate training sets for ANNs. In Sect. 3, we report and analyze the results of our experiments performed using benchmark and real-life datasets. Section 4 concludes the paper and highlights the future research pathways.

## 2 Genetic Evolution of Surrogate Training Sets

In this paper, we build upon a GA proposed in our previous work [11], and present how to exploit an analogous algorithm for evolving training sets for ANNs (GA-TS). Also, we show that $\boldsymbol{T}$'s extracted using our technique can be effectively used for learning other state-of-the-art classifiers as well.

In GA-TS, a population of $N$ solutions (surrogate training sets) undergoes the evolution in search of high-quality reduced $\boldsymbol{T}$. The size of individuals $t'$ (i.e., the cardinality of reduced sets) is kept constant across the population and is set *apriori*[5] (see the flowchart in Fig. 1). Then, the parents are paired for mating and are subject to crossover (in Fig. 1, we render operations that are inherently parallelizable as stacked steps). In this step, a child inherits distinct vectors (from both classes) from both parents—if the number of vectors in an offspring is smaller than $t'$, then random examples are drawn from $\boldsymbol{T}$ to keep the size of individuals constant in the course of the optimization. An analogous approach of replacing random examples from a surrogate model with those selected from

[4] This database is available at: https://www.uow.edu.au/~phung/download.html; last access: January 4, 2018.

[5] There exist approaches for updating the size of individuals dynamically [17], however we abstract from them in this paper.

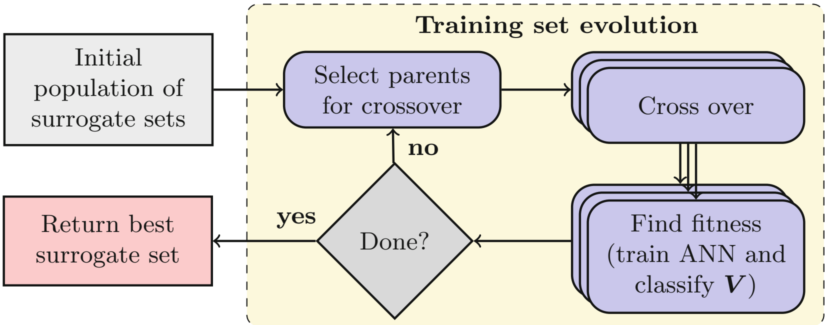
Training set evolution
Initial population of surrogate sets
Select parents for crossover
Cross over
Find fitness (train ANN and classify $\boldsymbol{V}$)
Done?
no
yes
Return best surrogate set

**Fig. 1.** Evolution of training sets using a genetic algorithm.

$\boldsymbol{T}$ is used in mutation (which is performed with the probability $\mathcal{P}_m$). The fitness of an individual is quantified as the classification performance (e.g., accuracy or area under the receiver operating characteristic curve) of a classifier (we exploit the accuracy of a trained ANN in this work) learned using the corresponding surrogate set over the validation set $\boldsymbol{V}$.

The GA is stopped if a termination condition is met—it may happen if a surrogate set of a desired quality has been obtained, the maximum number of generations have been already processed, or the maximum allocated execution time elapsed. Finally, the best individual from the last generation is returned and considered the final surrogate training set (in fact, we return a model trained using this surrogate set which is ready to use for new data). For more details on a GA to evolve surrogate training sets, we refer to [11].

## 3 Experimental Validation

GA-TS was implemented in `C++`. We analyzed several classifiers (implemented in `sklearn` in `Python`), including: artificial neural networks (ANNs), logistic regression (LR), AdaBoost (AB), Gaussian Naive Bayes (GNB), quadratic discriminant analysis (QDA), $k$-nearest neighbors ($k$-NN), decision trees (DTs) and random forests (RFs), and SVMs. The crossover and mutation probabilities in GA-TS were tuned experimentally to $\mathcal{P}_c = 0.7$ and $\mathcal{P}_m = 0.1$, respectively, and remained constant across the entire experimental study. Similarly, the learning rate of our ANN was kept constant and equal to 0.6.

### 3.1 Datasets

We focused on one UCI benchmark (*Wisconsin breast cancer*[6], 699 examples containing 10 attributes, not balanced, with 458 benign and 241 malignant examples), and one real-life set from the field of skin segmentation (*Skin*, $10^5$ examples of RGB values, balanced). The latter set contains examples of skin and

[6] This dataset is available at: https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original); last access: January 4, 2018.

non-skin pixels sampled from the ECU images (see example images rendered in Fig. 2—very high variability of inner-class vectors representing human skin can be noticed here). Both datasets were divided into a training set $\boldsymbol{T}$ from which surrogate sets are evolved, validation set $\boldsymbol{V}$ on which the fitness of individuals is calculated (during the evolution), and test set $\Psi$ used to quantify the generalization capabilities of a learned model (this set was *not* used during the evolution and contains unseen data). For *Wisconsin*, we followed the 5-fold cross-validation scenario, whereas *Skin* was divided randomly into $|\boldsymbol{T}| = 8 \cdot 10^4$ training, $|\boldsymbol{V}| = 10^4$ validation, and $|\Psi| = 10^4$ test examples.

**Fig. 2.** Example ECU images used to elaborate the *Skin* dataset.

### 3.2 Experiment 1: *Skin* Dataset

In this experiment, we verify the impact of the GA parameters, i.e., the population size ($N$), number of children generated for each pair of parental solutions during crossover ($N_{\mathrm{ch}}$), and size of surrogate sets being evolved ($t'$), on its search capabilities and quality of final solutions (ANNs learned using evolved $\boldsymbol{T}$'s)—each variant of GA-TS was executed 10×. We focused on *Skin*, and our ANN was fairly small (built with two hidden layers containing 6 and 3 neurons, respectively). The evolution of training sets was terminated if there was no improvement in the best fitness in five consecutive generations.

In Table 1, we gather the results (alongside the convergence time of our genetic technique) obtained using ANNs learned using surrogate sets extracted with GA-TS run in all investigated configurations. For very small sets (containing .1% of samples from the entire training set $\boldsymbol{T}$), GA-TS prematurely converges to the final solutions, and these surrogate training sets are low-quality (they are much worse compared to other sets, and the differences are statistically relevant

**Table 1.** Classification performance of ANNs trained using *Skin* surrogate sets extracted using GA-TS for all investigated configurations: Acc—accuracy obtained for the test set $\Psi$, $\Delta\eta$—gain in the average fitness (note that the gain can be negative because we do not exploit any elitism), $\tau$—GA convergence time; for all measures we report the minimum, average, and maximum values (the best values are boldfaced).

| | $\mathrm{Acc}_{\mathrm{min}}$ | $\mathrm{Acc}_{\mathrm{avg}}$ | $\mathrm{Acc}_{\mathrm{max}}$ | $\Delta\eta_{\mathrm{min}}$ | $\Delta\eta_{\mathrm{avg}}$ | $\Delta\eta_{\mathrm{max}}$ | $\tau_{\mathrm{min}}$ | $\tau_{\mathrm{avg}}$ | $\tau_{\mathrm{max}}$ |
|---|---|---|---|---|---|---|---|---|---|
| $t'$ | **Population size:** $N = 10$**, number of children:** $N_{\mathrm{ch}} = 2$ | | | | | | | | |
| $.001 \cdot t$ | .500 | .758 | .871 | −.093 | .003 | .075 | **23** | **173.5** | **671** |
| $.01 \cdot t$ | .819 | .851 | .869 | −.058 | .001 | .047 | 56 | 432.3 | 902 |
| $.03 \cdot t$ | .819 | .849 | .870 | −.307 | .006 | **.330** | 70 | 528.1 | 1722 |
| $.05 \cdot t$ | **.838** | **.870** | **.883** | **−.021** | **<.001** | .028 | 93 | 768.9 | 1697 |
| $.1 \cdot t$ | .815 | .867 | .881 | **−.021** | **<.001** | .026 | 140 | 1271.1 | 2900 |
| $N$ | **Cardinality of sets:** $t' = .01 \cdot t$**, number of children:** $N_{\mathrm{ch}} = 2$ | | | | | | | | |
| 10 | .799 | .842 | .870 | −.097 | <.001 | **.092** | 79 | **396.4** | 1168 |
| 20 | **.819** | **.851** | .869 | −.058 | **.001** | .047 | **56** | 432.3 | **902** |
| 40 | .500 | .816 | **.883** | **−.031** | −.001 | .035 | 115 | 1017.2 | 3185 |
| $N_{\mathrm{ch}}$ | **Cardinality of sets:** $t' = .01 \cdot t$**, population size:** $N = 10$ | | | | | | | | |
| 1 | .795 | .839 | **.882** | −.500 | **.001** | .048 | 53 | **324.9** | **561** |
| 2 | **.819** | **.851** | .869 | **−.058** | **.001** | .047 | 56 | 432.3 | 902 |
| 10 | .793 | .832 | .877 | −.080 | <.001 | **.069** | **51** | 919.4 | 3450 |

at $p < .01$ in Wilcoxon test—the null hypothesis saying that these GA-TS variants give the same-quality surrogate sets can be safely rejected). On the other hand, increasing the cardinality of evolved training sets does not necessarily improve their quality (the differences are not statistically important; $p$-value equals .52 for the Wilcoxon test), but notably slows down the convergence (see sets with 5% of $\boldsymbol{T}$ examples compared with those containing 10% of $\boldsymbol{T}$ vectors). Similarly, the population size should be kept as small as possible to speed up exploration of the solution space (candidate solutions with relatively small fitness values need more time to be improved in larger populations—see the minimum $\Psi$ accuracy for $N = 40$). Finally, generating even two children (instead of one) for each parental pair can help boost the exploitation and create well-fitted offspring solutions. However, it causes increasing the execution time of GA-TS, because all generated children must be assessed during the evolution (hence, it requires training a classifier using the surrogate set represented by a new individual). Therefore, selecting training sets can be seen as a multi-objective optimization problem—on one hand, performance of the final classifier is to be maximized, on the other hand—a selection algorithm should work as fast as possible (also, training time of a classifier should be minimized).

Example ECU images segmented using selected classifiers trained with the entire training set, and reduced training sets (evolved using GA-TS and sampled randomly from $\boldsymbol{T}$) are rendered in Fig. 3 (also, we present a manual ground-truth

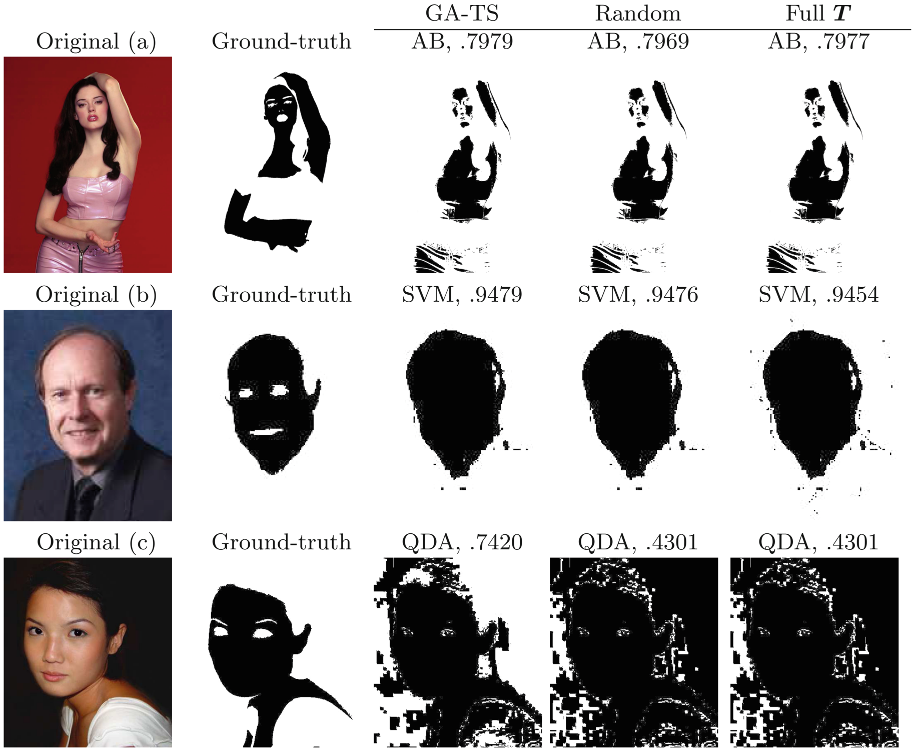

**Fig. 3.** Example ECU images segmented using selected classifiers learned with: a set of $0.25 \cdot |\boldsymbol{T}|$ vectors evolved using GA, randomly sampled from $\boldsymbol{T}$, and the entire $\boldsymbol{T}$.

segmentation of those images). It can be seen that sets evolved using a genetic technique allowed for learning well-performing models for very heterogeneous data (see inner-class variability in Fig. 2). It is especially visible for QDA, where the classification accuracy was improved by a margin greater than 30% (lots of false positives were filtered out). Hence, GA-TS can be effectively applied for other learners (even if the quality of surrogate sets during the evolution was quantified using a different classifier) as well to help boost their classification performance over difficult training sets. Also, it is possible to easily change the underlying classification engine that is used to elaborate the fitness value.

In Table 2, we report the classification accuracy alongside processing time (training and classification) over the selected ECU images for all investigated classifiers (learned using the entire set $\boldsymbol{T}$, and surrogate training sets extracting with our GA-TS and sampled randomly from $\boldsymbol{T}$—both sets contained $0.25 \cdot |\boldsymbol{T}|$ vectors). The results show that evolved sets not only do allow for obtaining highest-quality classification scores, but are also helpful in decreasing the execution time of the classification engine (especially when compared with learners trained using the entire training set). Interestingly, removing samples from $\boldsymbol{T}$

**Table 2.** Accuracy for selected ECU images (see a–c in Fig. 3) elaborated using all investigated classifiers trained with surrogate sets obtained using GA-TS and random sampling, and with the entire training set $\boldsymbol{T}$, alongside their training and classification (of all pixels in the corresponding image) times ($\tau^{\mathrm{T+C}}$, in seconds). The best results for each image are boldfaced.

| Image→ | (a) | | (b) | | (c) | |
|---|---|---|---|---|---|---|
| Classifier↓ | Acc | $\tau^{\mathrm{T+C}}$ | Acc | $\tau^{\mathrm{T+C}}$ | Acc | $\tau^{\mathrm{T+C}}$ |
| **Classifiers trained using surrogate sets selected using GA-TS** | | | | | | |
| ANN | .740 | 67.30 | .759 | 4.32 | .607 | 18.09 |
| LR | .808 | **27.05** | .861 | 1.05 | .600 | 7.68 |
| AB | .798 | 515.12 | .858 | 16.71 | .595 | 113.04 |
| GNB | .293 | 179.01 | .553 | 5.38 | **.817** | 46.11 |
| QDA | .278 | 184.13 | .775 | 9.39 | .742 | 42.01 |
| SVM | .263 | 489.12 | **.948** | 23.56 | .407 | 112.72 |
| $k$-NN | .804 | 211.57 | .757 | 6.89 | .682 | 55.55 |
| DT | .798 | 27.74 | .858 | **.95** | .595 | 7.89 |
| RF | **.818** | 454.78 | .771 | 15.13 | .681 | 116.63 |
| **Classifiers trained using surrogate sets selected randomly** | | | | | | |
| ANN | .733 | 76.71 | .745 | 3.52 | .682 | 19.66 |
| LR | .808 | 30.05 | .861 | .98 | .600 | 6.92 |
| AB | .797 | 518.19 | .857 | 16.47 | .594 | **2.28** |
| GNB | .289 | 182.09 | .557 | 6.02 | .815 | 42.45 |
| QDA | .274 | 199.36 | .743 | 6.41 | .430 | 42.62 |
| SVM | .263 | 456.79 | **.948** | 24.35 | .336 | 115.02 |
| $k$-NN | .802 | 211.89 | .756 | 7.12 | .682 | 56.23 |
| DT | .797 | 28.03 | .857 | 1.01 | .594 | 7.96 |
| RF | .795 | 451.84 | .807 | 16.07 | .594 | 113.98 |
| **Classifiers trained using the entire $\boldsymbol{T}$** | | | | | | |
| ANN | .728 | 71.83 | .324 | 12.79 | .684 | 30.54 |
| LR | .814 | 29.62 | .862 | 1.89 | .605 | 7.16 |
| AB | .798 | 549.14 | .858 | 23.18 | .595 | 143.30 |
| GNB | .289 | 179.26 | .557 | 5.14 | .815 | 44.62 |
| QDA | .277 | 204.52 | .743 | 5.35 | .430 | 42.42 |
| SVM | .258 | 348.32 | .945 | 300.03 | .408 | 721.22 |
| $k$-NN | .805 | 215.02 | .756 | 7.42 | .683 | 58.15 |
| DT | .798 | 27.25 | .858 | 1.01 | .595 | 8.11 |
| RF | .796 | 444.65 | .858 | 15.70 | .595 | 122.63 |

helped improve the accuracy—it may exhibit the presence of "noisy" vectors in $\boldsymbol{T}$ which should be rejected from the training set before the training process to maximize the generalization capabilities of a model.

### 3.3 Experiment 2: *Wisconsin Breast Cancer* Dataset

In this experiment, we compared pretty small ANNs (two hidden layers with three neurons each) trained using surrogate models (of various sizes) with state-of-the-art classifiers learned with the entire $\boldsymbol{T}$ for the *Wisconsin* benchmark dataset (as already mentioned, we performed 5-fold cross-validation and present the results averaged across the folds). Similarly to the previous experiment, each variant of GA-TS was executed 10× for each fold.

The classification accuracies reported in Table 3 confirm that ANNs outperform other approaches and deliver very fast operation and training (see $\tau^{\mathrm{T+C}}$). Also, the results are fairly consistent across the folds (see the ANN standard deviation $\sigma$). The increase in classification abilities of ANNs show that GAs can be successfully used for evolving $\boldsymbol{T}$'s not only for SVMs [17]. It is interesting to note that training of ANNs did not significantly vary across different surrogate set cardinalities (the boost in classification accuracy was not notable either). This is an indicator showing that this dataset is not very challenging, and even for an entire training set the learning process is affordable.

**Table 3.** Accuracy for *Wisconsin breast cancer* using the investigated classifiers, alongside their training and classification (of the test set) times ($\tau^{\mathrm{T+C}}$, in seconds). The best results in each column are boldfaced.

| Classifier | $\mathrm{Acc}_{\mathrm{avg}}$ | $\sigma(\mathrm{Acc}_{\mathrm{avg}})$ | $\tau^{\mathrm{T+C}}_{\mathrm{avg}}$ | $\sigma(\tau^{\mathrm{T+C}}_{\mathrm{avg}})$ |
|---|---|---|---|---|
| **Classifiers trained using the entire $\boldsymbol{T}$** | | | | |
| LR | .962 | .032 | .003 | .001 |
| AB | .948 | .035 | .165 | .014 |
| GNB | .965 | .014 | **.002** | **.001** |
| QDA | .958 | .018 | **.002** | .002 |
| SVM | .956 | .027 | .010 | .001 |
| $k$-NN | .949 | .042 | .008 | .007 |
| DT | .936 | .028 | .091 | .199 |
| RF | .962 | .023 | 1.418 | .019 |
| ANN | .964 | .019 | .190 | .037 |
| **ANNs trained using surrogate sets of various cardinalities $t'$** | | | | |
| ANN, $t' = 0.2 \cdot t$ | **.979** | **.005** | .188 | .031 |
| ANN, $t' = 0.1 \cdot t$ | .976 | .006 | .184 | .049 |
| ANN, $t' = 0.05 \cdot t$ | .977 | **.005** | .183 | .040 |

## 4 Conclusions and Future Work

In this paper, we exploited a genetic algorithm for evolving surrogate training sets (of significantly lower cardinalities than the original set) for ANNs. Also, we showed that such evolved sets can be effectively used for other state-of-the-art classifiers without introducing any changes in the core algorithm. Experimental validation (coupled with statistical tests) performed on real-life and benchmark datasets confirmed the applicability of this evolutionary training set selection scheme in practice. Classifiers learned with surrogate sets were confronted with those trained using the entire set and random surrogate sets, and they were shown to be outperforming the latter techniques. It is an indicator that training set selection should be used in practice for large, heterogeneous and potentially noisy datasets in order to select only those examples which are distinctive.

It would be interesting to see if a surrogate training set evolved for a given classifier can be directly used to improve a different learner—it could be potentially useful to decrease the fitness evaluation time in the course of optimization, and exploit the fastest classifiers to quantify the quality of surrogate sets during the evolution. Although the results reported in this paper are a very good indicator that it is possible, this issue requires further investigation. We currently work on a fully hands-free algorithm for evolving SVMs, in which not only surrogate models are optimized, but also features and kernel hyperparameters evolve. This algorithm will be applied to segment brain lesions from FLAIR MRI sequences and other medical images [25]. Finally, we aim to tackle the problem of selecting training sets for deep neural networks by means of evolutionary computation.

**Acknowledgments.** JN and MK were supported by the National Science Centre, Poland, under Research Grant No. DEC-2017/25/B/ST6/00474, and JN was supported by the Silesian University of Technology under the Grant BKM-509/RAu2/2017. This work was partially supported by the Polish National Centre for Research and Development (Innomed grant, POIR.01.02.00-00-0030/15).

## References

1. Abhishek, K., Singh, M., Ghosh, S., Anand, A.: Weather forecasting model using artificial neural network. Proced. Technol. **4**, 311–318 (2012)
2. Aibinu, A., Shafie, A., Salami, M.: Performance analysis of ANN based YCbCr skin detection algorithm. Proced. Eng. **41**, 1183–1189 (2012)
3. Balcázar, J., Dai, Y., Watanabe, O.: A random sampling technique for training support vector machines. In: Abe, N., Khardon, R., Zeugmann, T. (eds.) ALT 2001. LNCS, vol. 2225, pp. 119–134. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45583-3_11
4. Cervantes, J., Lamont, F.G., López-Chau, A., Mazahua, L.R., Ruíz, J.S.: Data selection based on decision tree for SVM classification on large data sets. Appl. Soft Comput. **37**, 787–798 (2015)
5. Cho, S., Cha, K.: Evolution of neural network training set through addition of virtual samples. In: Proceedings of IEEE CEC, pp. 685–688 (1996)

6. Chojaczyk, A., Teixeira, A., Neves, L., Cardoso, J., Soares, C.G.: Review and application of artificial neural networks models in reliability analysis of steel structures. Struct. Saf. **52**, 78–89 (2015). http://www.sciencedirect.com/science/article/pii/S016747301400085X
7. Ding, S., Li, H., Su, C., Yu, J., Jin, F.: Evolutionary artificial neural networks: a review. Artif. Intell. Rev. **39**(3), 251–260 (2013)
8. Hilado, S.D.F., Dadios, E.P., Gustilo, R.C.: Face detection using neural networks with skin segmentation. In: Proceedings of IEEE CIS, pp. 261–265 (2011)
9. Himmelblau, D.M.: Applications of artificial neural networks in chemical engineering. Korean J. Chem. Eng. **17**(4), 373–392 (2000)
10. Kamp, R.G., Savenije, H.H.G.: Optimising training data for ANNs with genetic algorithms. Hydrol. Earth Syst. Sci. **10**, 603–608 (2006)
11. Kawulok, M., Nalepa, J.: Support vector machines training data selection using a genetic algorithm, SSPR/SPR. In: Gimel'farb, G. (ed.) Structural, Syntactic, and Statistical Pattern Recognition. LNCS, vol. 7626, pp. 557–565. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34166-3_61
12. Li, Y.: Selecting training points for one-class support vector machines. Pattern Recogn. Lett. **32**(11), 1517–1522 (2011)
13. Liu, B.: Application of artificial neural networks in computer-aided diagnosis. In: Cartwright, H. (ed.) Artificial Neural Networks. MIMB, vol. 1260, pp. 195–204. Springer, New York (2015). https://doi.org/10.1007/978-1-4939-2239-0_12
14. Millard, K., Richardson, M.: On the importance of training data sample selection in random forest image classification: a case study in peatland ecosystem mapping. Remote Sens. **7**(7), 8489–8515 (2015)
15. Mirończuk, M., Protasiewicz, J.: A diversified classification committee for recognition of innovative internet domains. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (eds.) BDAS 2015-2016. CCIS, vol. 613, pp. 368–383. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34099-9_29
16. Moghaddam, A.H., Moghaddam, M.H., Esfandyari, M.: Stock market index prediction using artificial neural network. J. Econ. Finan. Adm. Sci. **21**(41), 89–93 (2016)
17. Nalepa, J., Kawulok, M.: Adaptive genetic algorithm to select training data for support vector machines. In: Esparcia-Alcázar, A.I., Mora, A.M. (eds.) EvoApplications 2014. LNCS, vol. 8602, pp. 514–525. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45523-4_42
18. Nalepa, J., Kawulok, M.: Adaptive memetic algorithm enhanced with data geometry analysis to select training data for SVMs. Neurocomputing **185**, 113–132 (2016). http://www.sciencedirect.com/science/article/pii/S0925231215019839
19. Nalepa, J., Kawulok, M.: Selecting training sets for support vector machines: a review. Artifi. Intell. Rev. pp. 1–44 (2018). https://doi.org/10.1007/s10462-017-9611-1
20. Nguyen, H.B., Xue, B., Andreae, P.: Surrogate-model based particle swarm optimisation with local search for feature selection in classification. In: Squillero, G., Sim, K. (eds.) EvoApplications 2017. LNCS, vol. 10199, pp. 487–505. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-55849-3_32
21. Pawełczyk, K., et al.: Towards detecting high-uptake lesions from lung ct scans using deep learning. In: Battiato, S., Gallo, G., Schettini, R., Stanco, F. (eds.) ICIAP 2017. LNCS, vol. 10485, pp. 310–320. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68548-9_29

22. Plechawska-Wojcik, M., Wolszczak, P.: Appling of neural networks to classification of brain-computer interface data. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (eds.) BDAS 2015-2016. CCIS, vol. 613, pp. 485–496. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34099-9_37
23. Przybyła-Kasperek, M.: Two methods of combining classifiers, which are based on decision templates and theory of evidence, in a dispersed decision-making system. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (eds.) BDAS 2015-2016. CCIS, vol. 613, pp. 109–119. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-34099-9_7
24. Reeves, C.R., Taylor, S.J.: Selection of training data for neural networks by a genetic algorithm. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 633–642. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056905
25. Starosolski, R.: Lossless Compression of medical and natural high bit depth sparse histogram images. In: Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., Kostrzewa, D. (eds.) BDAS 2015. CCIS, vol. 521, pp. 363–376. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18422-7_32
26. Wesolowski, M., Suchacz, B.: Artificial neural networks: theoretical background and pharmaceutical applications: a review. J. AOAC **95**(3), 652–668 (2012)
27. Yao, X., Islam, M.M.: Evolving artificial neural network ensembles. IEEE Comput. Intell. Mag. **3**(1), 31–42 (2008)
28. Zhang, G., Yan, P., Zhao, H., Zhang, X.: A computer aided diagnosis system in mammography using artificial neural networks. In: Proceedings ICBEI, vol. 2, pp. 823–826 (2008)