DOCTORAL THESIS

Evolving intelligence: Overcoming challenges for Evolutionary Deep Learning

Mohammed Ghaith Altarabichi

Halmstad University | School of Information Technology
DoctoralThesis | Halmstad University Dissertations no. 109

HALMSTAD
UNIVERSITY

Evolving intelligence: Overcoming challenges for Evolutionary Deep Learning

Mohammed Ghaith Altarabichi

DOCTORAL THESIS | Halmstad University Dissertations no. 109

Evolving intelligence: Overcoming challenges for Evolutionary Deep Learning
©Mohammed Ghaith Altarabichi
Halmstad University Dissertations no. 109
ISBN 978-91-89587-32-8 (printed)
ISBN 978-91-89587-31-1 (pdf)
Publisher: Halmstad University Press, 2024 | www.hh.se/hup
Printer: Media-Tryck, Lund

Abstract

Deep Learning (DL) has achieved remarkable results in both academic and industrial fields over the last few years. However, DL models are often hard to design and require proper selection of features and tuning of hyper-parameters to achieve high performance. These selections are tedious for human experts and require substantial time and resources. A difficulty that encouraged a growing number of researchers to use Evolutionary Computation (EC) algorithms to optimize Deep Neural Networks (DNN); a research branch called Evolutionary Deep Learning (EDL).

This thesis is a two-fold exploration within the domains of EDL, and more broadly Evolutionary Machine Learning (EML). The first goal is to make EDL/EML algorithms more practical by reducing the high computational cost associated with EC methods. In particular, we have proposed methods to alleviate the computation burden using approximate models. We show that surrogate-models can speed up EC methods by three times without compromising the quality of the final solutions. Our surrogate-assisted approach allows EC methods to scale better for both, expensive learning algorithms and large datasets with over 100K instances.

Our second objective is to leverage EC methods for advancing our understanding of Deep Neural Network (DNN) design. We identify a knowledge gap in DL algorithms and introduce an EC algorithm precisely designed to optimize this uncharted aspect of DL design. Our analytical focus revolves around revealing avant-garde concepts and acquiring novel insights. In our study of randomness techniques in DNN, we offer insights into the design and training of more robust and generalizable neural networks. We also propose, in another study, a novel survival regression loss function discovered based on evolutionary search.

To dad, Mohamed Wassel Altarabichi.

Acknowledgements

My heartfelt appreciation goes to my wife, May Shayboun, whose unwavering support has been the cornerstone of my PhD journey. Without her, this path would have been unimaginable. I owe an immeasurable debt of gratitude to my parents, Mohamed Wassel Altarabichi and Nihad Tarabishi, for their constant love, encouragement, and unshakeable belief in my capabilities. Although my dad passed away mid-way through this journey, I hold faith that this outcome would have filled him with immense joy.

I am profoundly thankful to my esteemed mentors, Sławomir Nowaczyk, Sepideh Pashami, and Peyman Sheikholharam Mashhadi. Their unwavering support, guidance, and invaluable counsel have navigated me through the intricate paths of this research endeavor. In particular, I hold Sławomir in the highest regard as an exceptional supervisor whose mentorship has been invaluable, and Sepideh Pashami, whose blend of social acumen and genuine care has profoundly impacted my journey.

Equally, I express my gratitude to my peers and collaborators whose camaraderie and collaborative spirit have illuminated this academic voyage. Among them, I owe special thanks to Abdallah Alabdallah, for the endless discussions which inexplicably led us both down the path of pursuing a PhD.

The unflagging support and resources provided by the faculty and staff at Halmstad University have played an indispensable role in the successful culmination of this research. Their unwavering assistance and guidance were instrumental in every phase of this scholarly pursuit.

My heartfelt appreciation extends to each of these individuals; their support and contributions have been essential. This dissertation stands as a testament to their collective encouragement and assistance. Thank you, from the depths of my gratitude, for making this endeavor possible.

Mohammed Ghaith Altarabichi
January, 2024

List of Papers

The following papers, referred to in the text by their Roman numerals, are included in this thesis, sorted chronologically.

PAPER I: **Surrogate-Assisted Genetic Algorithm for Wrapper Feature Selection**
Mohammed Ghaith Altarabichi, Sławomir Nowaczyk, Sepideh Pashami, Peyman Sheikholharam Mashhadi (2021). IEEE congress on evolutionary computation (CEC) (pp. 776-785). **IEEE**, `https://doi.org/10.1109/CEC45853.2021.9504718`.

PAPER II: **Extracting Invariant Features for Predicting State of Health of Batteries in Hybrid Energy Buses**
Mohammed Ghaith Altarabichi, Sławomir Nowaczyk, Sepideh Pashami, Peyman Sheikholharam Mashhadi (2021). IEEE 8th international conference on data science and advanced analytics (DSAA) (pp. 1-6). **IEEE**, `https://doi.org/10.1109/DSAA53316.2021.9564184`.

PAPER III: **Fast Genetic Algorithm for feature selection — A qualitative approximation approach**
Mohammed Ghaith Altarabichi, Sławomir Nowaczyk, Sepideh Pashami, Peyman Sheikholharam Mashhadi (2023). **Expert systems with applications, 211**, `https://doi.org/10.1016/j.eswa.2022.118528`.

PAPER IV: **Rolling the Dice for Better Deep Learning Performance: A Study of Randomness Techniques in Deep Neural Networks**
Mohammed Ghaith Altarabichi, Sławomir Nowaczyk, Sepideh Pashami, Peyman Sheikholharam Mashhadi, Julia Handl. (2023) *submitted.*

PAPER V: **Improving Concordance Index in Regression-based Survival Analysis: Discovery of Loss Function for Neural Networks**

Mohammed Ghaith Altarabichi, Abdallah Abdallah, Sepideh Pashami, Mattias Ohlsson, Thorsteinn Rögnvaldsson, Sławomir Nowaczyk. (2023) *submitted.*

The following papers are not included in this thesis, but were published during the PhD.

PAPER 1: **A vision-based indoor navigation system for individuals with visual impairmen**
Ahmed, M.U., Altarabichi, M.G., Begum, S., Ginsberg, F., Glaes, R., Östgren, M., Rahman, H. and Sorensen, M., 2019. A vision-based indoor navigation system for individuals with visual impairment. International Journal of Artificial Intelligence, 17(2), pp.188-201. https://www.diva-portal.org/smash/record.jsf?dswid=503&pid=diva2%3A1365489 [1].

PAPER 2: **Reaction Time Variability Association with Safe Driving Indexes**
Altarabichi, M.G., Ahmed, M.U., Begum, S., Ciceri, M.R., Balzarotti, S., Biassoni, F., Lombardi, D. and Perego, P., 2020. In Transport Research Arena TRA2020, 27 Apr 2020, Helsinki, Finland, 2020 https://mdh.diva-portal.org/smash/get/diva2:1366267/FULLTEXT03.pdf [2].

PAPER 3: **Predicting state of health and end of life for batteries in hybrid energy buses**
Altarabichi MG, Fan Y, Pashami S, Nowaczyk S, Rögnvaldsson T. In: Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference [Internet]. Singapore: Research Publishing Services; 2020. p. 1231–1231. https://doi.org/10.3850/978-981-14-8593-0_4515-cd [3].

PAPER 4: **Stacking ensembles of heterogenous classifiers for fault detection in evolving environments**
Mohammed Ghaith Altarabichi, Peyman Sheikholharam Mashhadi, Yuantao Fan, Sepideh Pashami, Sławomir Nowaczyk, Pablo Del Moral, Mahmoud Rahat, Thorsteinn Rögnvaldsson (2020). In 30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM15 2020, Venice, Italy, 1-5 November, 2020 (pp.

1068-1068). Research Publishing Services, https://doi.org/10.3850/978-981-14-8593-0_5555-cd [4].

PAPER 5: **Evolving Activation Functions**
Mohammed Ghaith Altarabichi (2022). Swedish Artificial Intelligence Society workshop SAIS, available online https://git.ri.se/sepideh.pashami/SAIS2022/-/blob/main/SAIS_2022_paper_4481.pdf.

Contents

List of Figures

1. Introduction

1.1 Introduction

The remarkable success of Deep Learning (DL) has revolutionized several fields, including computer vision [5–7], natural language processing [8–10], healthcare [11], video games [12; 13], and more. The success of DL is often attributed to the availability of larger datasets and increasingly potent computational resources [14]. This exponential growth in both data volume and computing capabilities has empowered us to expand the scale of Deep Neural Networks (DNN). Consequently, the DL algorithms that previously had limited success e.g., Convolutional Neural Network (CNN) [15], and Long Short-Term Memory (LSTM) [16], are now thriving. These algorithms continued to advance, driven by manual design based on human expertise and intuition. Innovative ideas e.g., ReLU [17], dropout [18], residual connections [6], and attention mechanisms [7; 9] allowed training deeper, more expressive models that generalize better to unseen data.

However, as DL algorithms improved, achieving high performance with them became more complex [19–21]. This complexity arises from the need to make a multitude of design decisions, including choosing the right architecture, fine-tuning hyperparameters, and selecting features. Importantly, these selections are often task-dependent, in the sense that different learning tasks give rise to different selections of features [22], architectures [23], and hyperparameters [24]. The search spaces of possible solutions are vast and complex. Therefore, human experts may struggle to navigate this intricate landscape manually, and the search for innovative solutions can be overwhelming.

The challenge is becoming more pronounced as the task-dependent nature continues to extend to other design choices of DNN e.g., the activation functions, as it was shown in [25; 26] that specialized activation functions discovered specifically for the task consistently outperformed ReLU in several benchmarks. The same observation seems to apply to the choice of loss function [27–29] compared to the popular cross-entropy. For all their simplicity and successes in practice, one cannot argue that ReLU and cross-entropy are the ideal activation and loss function choices for every DL task.

The inherent task-dependent nature of many DL design choices opens new research possibilities for optimizing other aspects of DNN. Evolutionary Computation (EC) methods prove to be particularly well-suited to solve such optimizations in DL, and there are several compelling reasons for this choice. Firstly, EC methods exhibit effective exploration of complex search spaces by maintaining a diverse population of solutions, thereby mitigating the risk of getting trapped in local optima [30]. Moreover, EC methods do not require derivatives or rely on assumptions of convexity [31]. This quality renders them especially advantageous for optimization tasks within DL, which frequently involve non-convex objective functions [32]. Furthermore, EC methods are capable of handling multi-objective optimization problems [33]. This capability is invaluable in scenarios where there is a need to simultaneously optimize multiple conflicting objectives, such as enhancing model accuracy while reducing training time. Collectively, these factors position EC methods as a natural choice for tackling optimization problems characterized by intricate and challenging search spaces [34; 35], and there has been a growing trend among researchers who are increasingly utilizing EC algorithms to optimize DNNs. This field is called Evolutionary Deep Learning (EDL) and can be viewed as a sub-field of Evolutionary Machine Learning (EML), where the goal is to use EC to optimize a Machine Learning (ML) model $\mathcal{M}$ for a given learning task $\mathcal{T}$. The field of EDL is no longer limited to Neural Architecture Search (NAS) [23] and Hyperparameter Optimization (HPO), or doing both NAS and HPO jointly [36]. Today, the field of EDL extends to evolving new algorithms [37], weights [38; 39], activation functions [25; 26], loss functions [27–29], and learning rate policies [40].

The work in this thesis serves two main objectives within the fields of EML/EDL. The first goal is to use EC methods to come up with new knowledge about designing DNNs. For this purpose, we identify a specific knowledge gap within our understanding of DL algorithms and introduce an EC algorithm designed specifically to optimize this previously unexplored aspect of DL design. Our analysis of the optimization outcomes is centered on the goal of uncovering innovative designs and acquiring novel insights and perspectives. Within this objective, we carried a work to improve our understanding of randomness techniques in training DNNs and their complex interactions in (Paper IV), and to discover novel loss functions for DNN in survival analysis (Paper V). Also, we have extended this objective by using the EC algorithm to identify knowledge in the applied field of Electric Vehicles (EVs) in (Paper II) by designing an algorithm to identify invariant features that generalize to unseen settings for hybrid buses e.g., new configurations and operating conditions.

Our second objective is focused on making EML/EDL algorithms more computationally efficient by the reduction of the high computational cost associated with these methods. The goal is to reduce the EC algorithms' run-time without trading off the quality of the final solutions. We propose methods to elevate the computation cost using approximate models based on concepts derived from the research on progressive sampling and optimization with approximate fitness function in (Paper I) and (Paper III).

1.2 Challenges

EC algorithms have faced the practical challenge of long run-time when used for optimizations in ML/DL [35]. The iterative process of EC involves running a large number of fitness function evaluations – where each evaluation requires training an ML model $\mathcal{M}$ from a learning algorithm $\mathcal{A}$. The training of $\mathcal{M}$ is needed in optimizations e.g., Feature Selection (FS) or Hyper-parameter Optimizations (HPO), to score the fitness of the candidate feature/hyper-parameter solutions. The computational cost of the EC algorithm is dependent on the time complexity of the learning algorithm $\mathcal{A}$. As most of the run-time is spent on fitness evaluations [22], we may approximate the time complexity of the EC algorithm $O(EC)$ as:

$$O(EC) \approx t \cdot O(\mathcal{A}) \tag{1.1}$$

where t is the total number of fitness function evaluations, and $O(\mathcal{A})$ is the time complexity of the learning algorithm $\mathcal{A}$. The long-run time of EC can be attributed for two reasons according to 1.1. First, O($\mathcal{A}$) can lead to long run-time, in particular for large datasets. Second, the search space is often high dimensional with very well-performing regions, often requiring a large t to arrive at novel and high-quality solutions.

1.2.1 Expensive model training

The time complexity of many learning algorithms $\mathcal{A}$ is dependent on the training dataset size, and can be expressed as a function of the number of features n and the number of instances k. For example, the training complexity of Decision Trees (DT) is $O(nk^2)$ [41], kNN is $O(n^2k)$ [42], while nonlinear SVM is between $O(n^2)$ and $O(n^3)$ [43]. Consequently, the complexity order of an EML algorithm where $\mathcal{A}$ is DT is $O(nk^2t)$, and is $O(n^2kt)$ where $\mathcal{A}$ is kNN. It is obvious that EML will not scale for high dimensional datasets, the run-time in such scenarios could take weeks [44].

The problem is evident given how data-hungry most ML models are; especially DL, but also many simpler ML models, do not converge quickly, and continue to improve significantly with more labeled data. Progressive sampling (PS) offers a method to examine the connection between sample size and

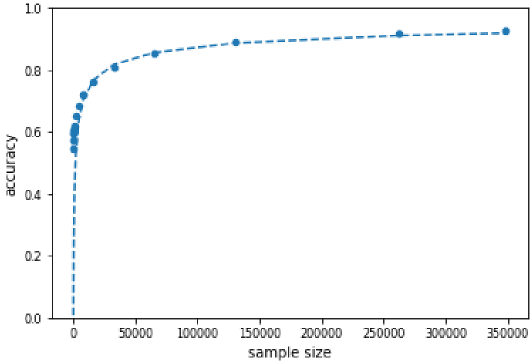


Figure 1.1: Learning Curve of the `covtype` data set using a Decision Tree as a model, and a Geometric Sampling Schedule $S_g = \{32, 64, 128, ..., 262\,144\}$. The last point to the right represents training with all the available 348 861 training set instances.

model accuracy by utilizing batches that gradually increase in size. This relationship can be visualized through a learning curve that portrays how model accuracy changes with varying sample sizes. Typically, a well-behaved learning curve tends to adhere to an inverse power law function [45]. We may observe from Fig. 1.1 showing the learning curve of the covtype data set from the UCI ML repository[2] that the convergence point is not reached, for a simple DT classifier, even when training on more than $250K$ instances.

The expensive model training challenge is even worse for EDL as the training of DL models tends to require more time than conventional ML. The full training of a ResNet50 network on the ImageNet dataset [46] using a modern TPU could take days, even weeks. It is not uncommon for EDL papers to report run times in the order of months of GPU time [23; 25]. Therefore, it is crucial for EDL algorithms to come up with cheaper alternatives than evaluating candidate solutions by training $\mathcal{M}$.

1.2.2 Large and complex search spaces

The task of using EC to identify novel high-performing solutions with DL is challenging because of two general characteristics of the search space of such problems. First, the search space is vast due to the large number of possible design choices [20]. Second, the promising regions within the search space are rare, as only very specific design choices lead to high performance [19; 21]. It is a search for a tiny-sized needle in a colossal haystack. The identification of

[2] http://archive.ics.uci.edu/ml

high-performing solutions in such spaces will likely require a large number of t, often in the order of thousands of evaluations [47].

Large space of possibilities

The search spaces of possible architectures, features, or hyperparameters are often complex and immense in size. The size of the search space of e.g., the HPO problem grows exponentially with more hyperparameters to tune [48]. Such extensive search space presents a significant challenge in the quest for finding the best-performing setting, as navigating through it efficiently requires a large number of evaluations t, even for sophisticated optimization techniques e.g., Bayesian methods [49; 50]. In our work of optimizing randomness in (Paper V) to optimize all 22 randomness techniques at once requires 10^{20} evaluations, each taking 10 minutes to train, for a total time on the order of 100 000 multiples of the age of the universe. Another example from our work in (Paper V), where we used a tree-based representation similar to [25; 51; 52] to represent survival loss functions, is that the number of possible loss functions is on the order of 10^{21}.

The Edge of Chaos

The multitude of design decisions is not the only hurdle to achieving higher performance with DL. A number of papers have shown that only very specific choices of hyperparameters lead to good performance with deep networks [19; 21], an observation known in the literature as “Edge of Chaos”. This difficulty meant that practitioners must identify these specific choices of weight initialization schemes [53], adding noise to optimization [54], architecture selections e.g., batch normalization [55], skip/residual connections [6], to achieve high performance with deep models.

We have observed this characteristic concerning the scarcity of good configuration from the space of possible solutions in our work in (Paper IV). As we showed, in practice, it is impossible to identify well-performing regions of the search space at random [24] even with thousands of evaluations. This failure of random search is a clear indicator that high-performing regions are rare within the vast space of possible configurations.

1.3 Research questions

The work in this thesis aims to answer two research questions. The first question aims to exploit the opportunity of using EC to optimize aspects of DL design with the goal of generating new generalizable insights and knowledge.

In the second, we are trying to address the run-time challenge of using EC for ML/DL:

- **RQ1** Can we come up with new knowledge about designing DNN using EC?

  This research question is meant to exploit the opportunity of using EC to uncover new knowledge about DL. Automatic search methods have demonstrated a lot of successes in enriching our knowledge of DL by discovering new loss functions e.g., Baikal Loss [27], activation functions e.g., Swish [52], layers e.g., novel skip connections in LSTM [14], and architectures [56].

  Our approach to answer this question is based on designing EC algorithms that optimize previously unexplored DL design choices, to address gaps in our understanding of DL algorithms, and to come up with novel designs/configurations.

- **RQ2** How do we reduce the run-time of EML/EDL using approximate computationally efficient models (surrogates) without compromising the quality of solutions?

  The long run-time of using EC to optimize ML/DL is widely considered a major challenge for the applicability of these methods. The problem is recognized in feature selection, where existing EC methods are described as "unfit to solve big data tasks" [22], with the majority of research limited to small datasets e.g., less than 1 000 instances. The same challenge is also recognized for HPO [57], NAS [23], and practically, for any optimization using EC with DL [58].

  The practical solutions to the run-time problem often compromise solution quality for shorter run-time e.g., filter and hybrid FS. Our objective is to answer this question by guiding the optimization through surrogate models that significantly reduce the total run-time while matching – or even exceeding – the performance realized with a classical EC setting.

1.4 Contributions

Contributions in the fields of EML/EDL could be broadly realized in one of two forms. The first is **new knowledge about DL/ML**. A novel scientific knowledge in the field of DL could be discovered by exploiting the ability of evolution to uncover unfamiliar, yet useful behavior [59]. An example of a creative discovery of EC that was not directly intuitive can be given from our work in (Paper V). We have examined how to improve the performance of

DNN by injecting different kinds of randomness into the training process. For this purpose, we carried out an HPO search to optimize the levels of different randomness techniques e.g., dropout, or gradient noise. An interesting setting was found during the search in MNIST datasets with a weight initialization variance set to zero by our PSO optimizer. The settings achieved 99.66% test accuracy, reducing the test error by 41.38% in comparison to baseline default settings. This constant initialization is surprising as it, at first glance, seems to prevent different neurons from learning different features. However, the weight symmetry [60] in this setting is broken with activation noise [61], allowing training while using a constant zero initialization of the weights. It must be noted that activation noise was not originally designed with the purpose of overcoming poor/constant initialization. The injection of noise into the output of the activation function was originally proposed to solve an entirely different problem, the saturating behavior of some activation functions like Sigmoid and Tanh[62]. In our work, the EC method discovered an alternative benefit.

Below we provide the complete list of contributions that fall under discovering new knowledge in DL achieved in this thesis and the appended papers:

- We show that different optimizers, such as Adam and Gradient Descent with Momentum, work best with distinct patterns of noise (Paper IV).

- Our empirical results suggest that data augmentation and weight initialization are the top contributors to performance improvement in CNNs (Paper IV).

- Our results showed superior performance of DropConnect compared to the much more popular variant Dropout with fully connected layers (Paper IV).

- We propose Gradient Dropout (GD), a method of randomly masking the gradient of some parameters during backpropagation to improve the performance of DNNs (Paper IV).

- We propose Loss Noise (LN), a method of injecting noise to loss function calculations to improve the performance of DNNs (Paper IV).

- We propose MSCE_{Sp}, a novel survival regression loss function that performs significantly better than the alternative Mean Squared Censored Error (MSCE) (Paper V).

- We demonstrate the importance of the non-zero gradient for the censored cases part of our proposed loss function MSCE_{Sp} (Paper V).

The second type of contributions comes in the form of **better EML/EDL algorithms**. We primarily qualify better algorithms as ones that lead to higher quality final solutions with a cheaper run-time. In our papers, we have presented the following novel algorithms:

- The feature selection algorithm SAGA from (Paper I), is shown to identify a significantly higher accuracy feature subset than ones found with a wrapper GA, while being three times faster.
- The feature selection algorithm GADIF from (Paper II), is shown to identify invariant features that lead to a better generalization of the machine learning models to an unseen domain.
- The approach proposed in (Paper IV), allows EC algorithms to scale better for large datasets with over 100K instances, independently from the used EC as shown with results of CHC and PSO.
- The algorithm SAGA$_{loss}$ proposed in (Paper V) can identify specialized differentiable loss functions for survival analysis regression that maximizes the C-Index performance.
- The PSO optimizer of randomness we proposed in (Paper V) handled 22 randomness interventions; no previous work from the literature focused on discussing the compatibility and effectiveness of random techniques in DL as comprehensively.

1.5 Summary of the papers

- **Paper I: Surrogate-assisted genetic algorithm for wrapper feature selection**

  This paper [63] addresses the challenge of the long run-time of feature selection. We answer **RQ2** by proposing a novel multi-stage feature selection framework that leverages multiple levels of approximations or surrogates to improve the efficiency and quality of feature selection solutions, especially on large datasets.

  We propose the algorithm Surrogate-Assisted Genetic Algorithm (SAGA) designed based on our framework to guide the evolutionary search in an efficient manner. During the early exploration phase, SAGA utilizes surrogates to make informed decisions, switching to the evaluation of the original function only in the final exploitation phase. We demonstrate that the upper-bound run-time of SAGA's surrogate-assisted stage

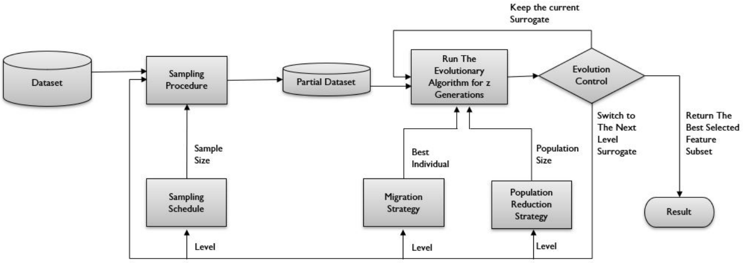

Keep the current Surrogate
Dataset
Sampling Procedure
Partial Dataset
Run The Evolutionary Algorithm for z Generations
Evolution Control
Sample Size
Best Individual
Population Size
Switch to The Next Level Surrogate
Return The Best Selected Feature Subset
Sampling Schedule
Migration Strategy
Population Reduction Strategy
Result
Level
Level
Level

Figure 1.2: A flowchart of the framework of surrogate-assisted evolutionary search for neural network optimization.

is at worst equal to that of a wrapper Genetic Algorithm (GA) while scaling better for complex learning algorithms with a high number of instances. Our empirical results show that SAGA significantly reduces computation time compared to a baseline wrapper GA while achieving solutions of significantly higher accuracy. On average, SAGA arrives at near-optimal solutions approximately three times faster than a wrapper GA. Importantly, the paper emphasizes the design of an evolution control approach to prevent surrogates from leading the evolutionary search toward false optima.

The flowchart in Figure 1.2 outlines the framework along with the key components: evolutionary algorithm, sampling procedure, evolution control, migration strategy, and population reduction.

- **Paper II: Extracting invariant features for predicting state of health of batteries in hybrid energy buses**

This study [64] focuses on the importance of monitoring the health of batteries in electric vehicles (EVs), which are a critical and costly component of these vehicles. To ensure the reliability of EVs and optimize their sustainability, it is crucial to track battery deterioration and efficiently utilize remaining battery capacity, especially in the context of electric buses with varying configurations and operating conditions. The challenge lies in developing degradation models for each unique combination of settings, considering factors like limited failure data for new settings, data heterogeneity, scarcity of data for less common configurations, and a lack of comprehensive engineering knowledge. Thus, the study aims to automate the transfer of machine learning models to new settings by identifying features that remain consistent across different

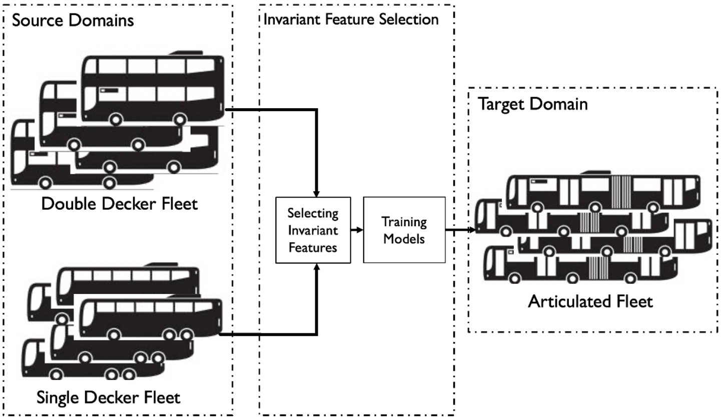


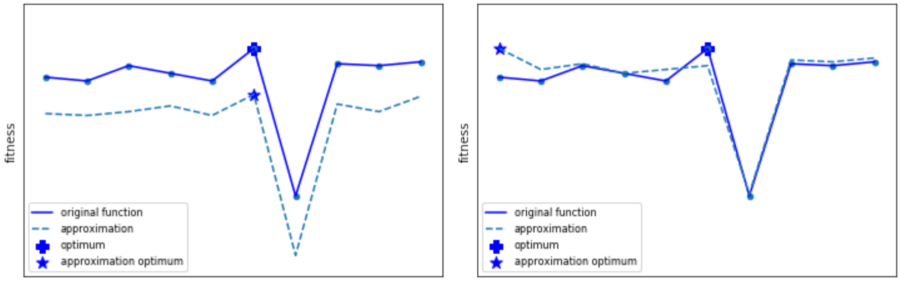
Figure 1.3: An illustration of the GADIF algorithm

scenarios.

We propose a feature selection method, called Genetic Algorithm for Domain Invariant Features (GADIF), employing a GA to select a set of invariant features for training machine learning models. These selected features maximize invariance across diverse settings as depicted in Fig. 1.3. The proposed GA incorporates a fitness function that considers both task performance and domain shift. We compare the performance of GADIF to classical feature selection methods without any transfer learning mechanisms, specifically in the context of adapting to unseen domains. The experimental results demonstrate that using invariant features results in better generalization of machine learning models to previously unencountered domains. In essence, GADIF helps ensure the robustness and adaptability of machine learning models when dealing with varying EV settings, ultimately improving our understanding of electric vehicle technology, in line with **RQ1**.

- **Paper III: Fast Genetic Algorithm for feature selection — A qualitative approximation approach**
  This paper [65; 66] addresses the challenge of applying EC in real-world scenarios where evaluating the fitness function can be computationally expensive. EC often requires numerous fitness function evaluations, which can be impractical for tasks like machine learning model training. To mitigate this issue, we propose a two-stage surrogate-assisted evolutionary approach for feature selection, particularly on large datasets, to answer **RQ2**.

  We introduce the concept of "Approximation Usefulness" to ensure the



(a) Useful approximation with large error. **(b)** Poor approximation with small error.

Figure 1.4: A qualitatively useful approximation for combinatorial optimization is shown in (a). The approximation correctly identifies the maximum of the original function, even though the approximation error is large. On the other hand, the approximation in (b) offers better quantitative approximation (values closer to the original fitness), but it leads to a false optimum.

correctness of EA computations when using approximations, such as meta-models or surrogates, for the fitness function. We develop a method to construct a lightweight qualitative meta-model through active data instance selection. This meta-model is then utilized for feature selection within the GA-based CHC (Cross-generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation) algorithm, resulting in a variant called CHC$_{QX}$.

The experimental results demonstrate that CHC$_{QX}$ converges faster to feature subset solutions with significantly higher accuracy compared to the original CHC, especially for large datasets containing over 100,000 instances. The approach's effectiveness is also extended to Swarm Intelligence (SI), a branch of Evolutionary Computation (EC), where we introduce a qualitative approximation adaptation of Particle Swarm Optimization (PSO).

Fig. 1.4a shows an example of a valuable qualitative approximation, according to Equation3.3.

- **Paper IV: Rolling the Dice for Better Deep Learning Performance: A Study of Randomness Techniques in Deep Neural Networks**

This paper conducts a comprehensive investigation into the effects of various randomness techniques in DNNs on network performance. Injecting randomness during DNN training is known to reduce overfitting and enhance generalization, but how different randomness methods interact and contribute to performance remains unclear. To address this, we categorize randomness techniques into three types: data, network, and optimization, and propose two new techniques: adding noise to the loss function and random masking of gradient updates.

Using a Particle Swarm Optimizer (PSO), we explore high performing configurations for injecting randomness to maximize DNN performance in computer vision tasks. We evaluate over 30,000 configurations, analyzing the individual and combined effects of randomness techniques to answer **RQ1**. The results highlight that randomness in data augmentation and weight initialization significantly improves performance, as seen in Fig. 1.5. Additionally, different optimizers prefer specific types of noise patterns.

- **Paper V: Improving Concordance Index in Regression-based Survival Analysis: Discovery of Loss Function for Neural Networks**

In this paper, we explore the use of evolutionary algorithms to optimize a survival loss function for neural networks with the goal of improving

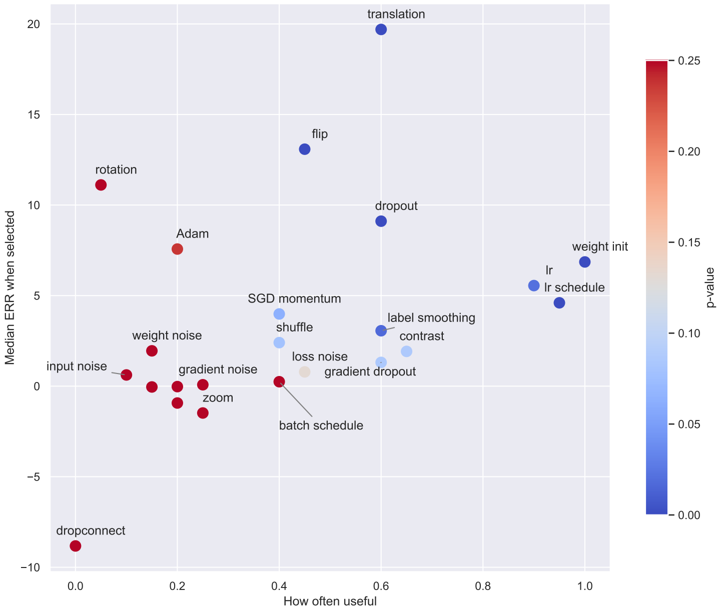


Figure 1.5: Visualization of randomness techniques, by comparing the number of times each one achieved a positive ERR to its median ERR, i.e., how often versus how much each technique contributed (across 20 ablation runs, for the CNN network, all four datasets). Colors indicate the p-value of Student's t-test for the null hypothesis of "technique has median ERR ≤ 0."

the C-Index performance. The paper contributes SAGA$_{loss}$ an algorithm designed to optimize a specialized neural network's differentiable loss function specifically for maximizing the C-Index. Based on our observations from the evolutionary search, we propose a new survival regression loss function called MSCE$_{Sp}$ as observed in Fig. 1.6. MSCE$_{Sp}$ outperforms the commonly used Mean Squared Censored Error (MSCE) loss function in the context of survival analysis. We highlight the significance of the non-zero gradient for the censored cases component of the loss function to answer **RQ1**.

The experimental results presented in the paper demonstrate the effectiveness of the evolutionary-discovered loss functions and the proposed MSCE$_{Sp}$ function. These functions perform generally better than the traditional MSCE loss function on 19 benchmark datasets, emphasizing the potential of evolutionary optimization for improving survival analysis in neural networks.

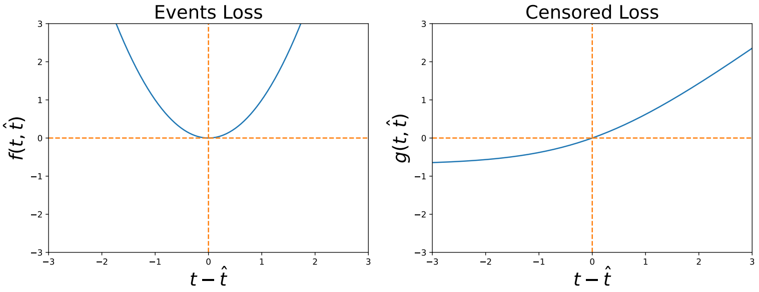


Figure 1.6: Our proposed survival loss function MSCE$_{SP}$.

2. Background

2.1 EC Model Search in Machine Learning

The common goal of a supervised ML task $\mathcal{T}$ given a labeled dataset $X_i = \{(\vec{x}_i, y_i), ..., (\vec{x}_n, y_n)\}$ of n samples and k features is to construct a model $\mathcal{M}$ from a learning algorithm $\mathcal{A}$ that can generalize to unseen examples. The partial set X^{tr} is used to train $\mathcal{M}_\theta$ via optimizing its parameters θ to minimize a predefined loss $\mathcal{L}(X^{tr}; \mathcal{M}_\theta)$, while setting aside the set X^{te} as unseen examples. A validation dataset $X^{va} \subset X^{tr}$ is often kept to estimate $\mathcal{L}(X^{te}; \mathcal{M}_\theta)$ by monitoring $\mathcal{L}(X^{va}; \mathcal{M}_\theta)$ during training.

This process of optimizing the model parameters θ is known as the training optimization loop, and is only one part of the ML pipeline that typically involves: data pre-processing, model search, model training, and model evaluation [58]. In this work, the focus is on EC model search methods, where the general goal is to find better models by selection of features, tuning hyper-parameters, and architecture choices. EC Model search optimizations e.g., Feature Selection (FS) and Hyper-parameter Optimization (HPO), can be viewed as bilevel optimizations [67] of two levels, where an EC method is used in the outer level to optimize features/hyper-parameters and the inner (training) level optimizes the model parameters θ. In the next subsections, we discuss the literature on using EC methods for FS and HPO.

2.1.1 Feature Selection

The task of feature selection involves selecting a new feature subspace $\mathbb{R}^l$ from the original feature space $\mathbb{R}^k$ (where $l \leq k$). The objective is to train a machine learning model $\mathcal{M}$ with feature subset μ while maintaining or improving performance (e.g., accuracy, $\mathcal{U}$) compared to the full feature space $\mathbb{R}^k$. The task of finding the optimal feature subset μ^* can be expressed as the maximization of the fitness function $\mathcal{F}$:

$$\mu^* = \arg\max_{\mu} \mathcal{F}(\mu; \mathcal{A}, X^{tr}, X^{va}, \mathcal{U}). \tag{2.1}$$

The FS methods using EC are usually grouped into three main types: filter, hybrid, and wrapper [22]. Filter methods, in contrast to wrapper algorithms, offer computational efficiency by relying on metrics like correlation [68], mutual information [69; 70], ReliefF [71], Fisher score [72], inconsistency rate [73; 74], and even ensembles of such metrics [75] to gauge the suitability of feature subsets. However, filter methods have a fundamental drawback as they are agnostic to the underlying ML model $\mathcal{M}$ [76]. In contrast, wrapper methods assess feature subsets by considering their impact on the performance of the machine learning algorithm, often yielding superior results [77; 78]. An extensive review of 22 diverse filter methods revealed that no single filter method consistently outperforms all others [79], a result that suggests that choosing the right filter approach for the learning task is challenging.

Hybrid methods adopt a two-stage strategy by combining both filter and wrapper approaches. In this hybrid paradigm, the initial step involves applying a filter to the features, with the primary aim of narrowing down the search space. Only the top-ranked features are subsequently utilized by the EC method in the second phase. While this technique has been employed in various studies [80–84], it exhibits two significant limitations. Firstly, the reduction in the search space primarily pertains to the features, making it less beneficial for datasets with a large number of instances. Secondly, low-ranked features might possess significance when combined with other features, thereby leading to the oversight of potential feature interactions.

The final approach to feature selection is the wrapper that relies on the ML model $\mathcal{M}$ to explicitly evaluate the fitness of feature subsets. A number of ideas were proposed to use a small portion of the training set to train $\mathcal{M}$ for evaluations to reduce the computational cost [42; 85–87]. However, the trade-off between computational efficiency and accuracy for these approaches is unclear, since all experiments reported in these papers lack direct comparison against a classical wrapper that uses all available data.

In summary, while the wrapper approach is widely regarded as the gold standard for achieving optimal solution quality in feature selection, its practical utility is constrained by the substantial computational expenses associated with its implementation. The real-world feasibility of the wrapper approach is hindered by the considerable computational resources and time required for its execution.

2.1.2 Hyper-parameters Optimization

The learning algorithm $\mathcal{A}$ often has a number of settings that must be chosen prior to training known as hyperparameters (λ). The objective of a hyper-parameter search is to identify the set λ^* that can lead to an optimal model $\mathcal{M}_\theta = \mathcal{M}_\theta{}^*$ performance represented by minimizing $\mathcal{L}(X^{va};\mathcal{M}_\theta)$. This can be formalized as:

$$\lambda^* = \arg\min_{\lambda} \mathcal{L}(X^{va}; \mathcal{A}(X^{tr}; \lambda)) = \arg\min_{\lambda} \mathcal{F}(\lambda; \mathcal{A}, X^{tr}, X^{va}, \mathcal{L}). \tag{2.2}$$

The objective function $\mathcal{F}$ measures the the corresponding $\mathcal{L}(X^{va}; \mathcal{M}_\theta)$ of using a candidate hyperparameters setting λ to train $\mathcal{M}$. Equation 2.2 can be extended to maximization of a performance measure (metric) such as accuracy $\mathcal{U}$:

$$\lambda^* = \arg\max_{\lambda} \mathcal{F}(\lambda; \mathcal{A}, X^{tr}, X^{va}, \mathcal{U}). \tag{2.3}$$

The problem of identifying the optimal hyperparameter λ^* is challenging, as it is widely acknowledged that different datasets, tasks, and families of learning algorithms often require specific configurations [24]. Among the various HPO strategies, Grid Search (GS) has been a popular choice among practitioners, particularly when dealing with HPO problems that involve a relatively small number of hyperparameters, typically fewer than five [88]. However, GS is plagued by the curse of dimensionality, as it becomes quickly impractical when dealing with a large number of hyperparameters, such as those associated with DNNs [20].

An alternative to GS is Random Search (RS), a foundational approach for HPO problems related to DNNs [24]. RS is often more efficient than GS when dealing with extensive search spaces, delivering faster results [24; 89]. However, RS suffers from a fundamental drawback: it samples hyperparameters independently of past evaluations, which means it fails to fully exploit promising regions within the search space [90; 91].

To tackle complex search spaces in HPO problems, various metaheuristics have been utilized. These include Particle Swarm Optimization (PSO) [92], Genetic Algorithms (GA) [93], Bayesian methods [49; 50], and the Iterated Racing Procedure [94]. Among these, population-based methods like GAs often outperform GS and RS, especially when handling extensive search spaces [89]. Similarly, PSO has proven effective in exploring the solution space of DNN hyperparameters, enabling competitive performance even with minimal network architectures [95].

3. Methods

In this section, we introduce the overall methodology used throughout all papers. We explain the major processes while highlighting how we approached them in every single paper. The flowchart in Fig. 3.1, introduces the processes of our meta-method. We start by designing a representation of candidate solutions and defining the search space. The next step is choosing an appropriate EC algorithm, and designing a fitness function to guide the search. As surrogate models are used to guide the search in most of our studies, we explain the designed evolution controls. The final step after concluding the iterative search procedure of EC is the evaluation of the results, with two distinctive goals, either evaluation of the performance of new algorithms, or revealing insights into the problem domain.

3.1 Designing search space and solution representation

An important aspect of using EC to optimize ML models is the proper design of a balanced search space. Whether the goal is to select architectures, fine-tune hyperparameters, or discover loss functions, a trade-off always exists between the size of the search space and the quality of solutions within it [52]. A complex search space designed by including a large number of hyperparameters [20] is more likely to contain high-performing configurations. On the other hand, the size of the space grows exponentially with more hyperparameters to tune [48]. The same trade-off can be observed for other optimizations as with loss function search. An overly constrained search space will not contain novel loss functions, whereas a search space that is too large will be difficult to effectively search. The choice of how to represent the individual solutions plays a major role in determining the size of the search space.

- **Paper I**: Feature subsets are represented as binary strings. The size of the search space varied for different datasets between 10^2 - 10^{80} feature subsets.

- **Paper II**: Feature subsets are represented as binary strings. The size of the search space of the EV dataset is 10^3 feature subsets.

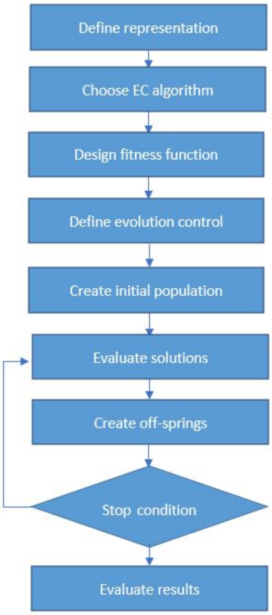


Figure 3.1: A flowchart of the overall method used to carry model search.

- **Paper III**: Feature and instance subsets are represented as binary strings. The size of the search space varied between 10^2 - 10^{80} feature subsets for different datasets.

- **Paper IV**: Hyper-parameter settings are represented with a mix of continuous and discrete values. The size of the search space is on the order of 10^{20} possible settings.

- **Paper V**: Loss functions are represented as trees of unary and binary operators, similar to the representation used by [25; 51; 52]. The number of possible loss functions is on the order of 10^{21}.

3.2 Choosing EC algorithm

The choice of an appropriate EC method is not a straightforward decision given that the "No free lunch theorem" [96] is applicable to EC. It is widely accepted that no EC methods are universally superior across a wide range of optimization problems [97; 98]. Also, EC algorithms require setting some parameters of their own to optimize their performance. We select algorithms with different exploration/exploitation characteristics based on the objective of the research.

As the identification of new knowledge in the domain of DL is a major focus, we choose algorithms/settings with strong exploration performance.

We motivate the choice of a particular EC by first excluding the possibility of an exhaustive search e.g., GS, due to the size of the search space. Second, we compare it to a baseline search procedure e.g., RS, to establish the difficulty of the problem. Afterward, we carry out experiments to identify good selections of primary settings e.g., population size, or total number of generations.

- **Paper I**: CHC [30], a GA-based algorithm.
- **Paper II**: Elitist GA with one point crossover and bit-flip mutation.
- **Paper III**: CHC is used for instance selection, while both CHC and PSO with star topology are used for feature selection.
- **Paper IV**: PSO with star topology.
- **Paper V**: A Genetic Programming (GP) CHC-based Algorithm .

3.3 Designing fitness function

The fitness or objective function defines the criteria by which the quality of potential solutions within the population is assessed.

- **Paper I**:

  $$\mu^* = \arg\max_{\mu} \mathcal{F}_I(\mu;\mathcal{A},X^{tr},X^{va},\mathcal{U}). \tag{3.1}$$

  The objective function $\mathcal{F}_I$ accepts a candidate feature subset μ and returns the corresponding accuracy $\mathcal{U}$ on the validation split X^{va} using the surrogate model $\mathcal{M}'$, μ^* is the feature subset that leads to maximum validation accuracy according to $\mathcal{M}$.

- **Paper II**:

  $$\mu^* = \arg\max_{\mu} \mathcal{F}_{II}(\mu;\mathcal{A},X^{tr},X^{va},\mathcal{P}). \tag{3.2}$$

  The objective function $\mathcal{F}_{II}$ accepts a candidate feature subset μ and returns $\mathcal{P}$ the leave-one-domain-out cross-validation performance of model $\mathcal{M}$ trained in a wrapper setting using feature subset μ, μ^* is the feature subset that lead to maximum leave-one-domain-out cross-validation performance according to $\mathcal{M}$.

- **Paper III**: I

  $$\iota^* = \arg\max_{\iota} \mathcal{F}_I(\iota;\mathcal{A},X^{tr},X^{va},\rho). \tag{3.3}$$

The objective function $\mathcal{F}_I$ accepts a candidate instance subset ι and returns the corresponding Spearman rank correlation between evaluations conducted using model $\mathcal{M}$ and evaluations performed using the meta-model $\mathcal{M}'$, ι^* is the instance subset that leads to the surrogate model $\mathcal{M}^*$ with prefect correlation with the original model $\mathcal{M}$.

- **Paper IV**:

$$\lambda^* = \arg\max_{\lambda} \mathcal{F}_{IV}(\lambda; \mathcal{A}, X^{tr}, X^{va}, \mathcal{U}). \tag{3.4}$$

The objective function $\mathcal{F}$ accepts a candidate hyperparameters setting λ and returns the corresponding accuracy $\mathcal{U}$ on the validation split X^{va}, λ^* is the hyperparameter setting that leads to a maximum accuracy on validation split.

- **Paper V**:

$$\mathcal{L}^* = \arg\max_{\mathcal{L}} \mathcal{F}_{V}(\mathcal{L}; \mathcal{A}, X^{tr}, X^{va}, \mathcal{CI}). \tag{3.5}$$

The objective function $\mathcal{F}$ accepts a candidate survival regression loss function $\mathcal{L}$ and returns the corresponding validation C-index $\mathcal{CI}$ performance, $\mathcal{L}^*$ is the loss function that leads to a maximum C-index performance on the validation set.

3.4 Defining evolution control

It has been observed that when a surrogate model $\mathcal{M}'$ is employed for fitness evaluations, there is a high probability of the evolutionary algorithm converging towards a false optimum [99]. A false optimum is a solution that represents the optimal point according to the approximate model but does not align with the true optimum of the original fitness function.

Hence, in many instances, it is crucial to combine the approximate model $\mathcal{M}'$ with the original model $\mathcal{M}$. This concept can be viewed as a matter of model management or evolution control. Evolution control implies that when employing evolutionary computation with approximate models, the original fitness function is employed to assess certain individuals or all individuals in specific generations. We design evolution control strategies in our work to ensure that first, the surrogate model $\mathcal{M}'$ will not mislead the optimization to a false optimum. And second, the computational cost should be reduced as much as possible.

- **Paper I**: The best individual found by surrogate $\mathcal{M}'$ is reevaluated using $\mathcal{M}$, after every fixed number of (z) generations.

- **Paper II**: No evolution control is needed, as we have used the original model $\mathcal{M}$ for all fitness evaluations.
- **Paper III**: All individuals in the population evaluated with $\mathcal{M}'$ are reevaluated using $\mathcal{M}$ after every fixed number of (z) generations.
- **Paper IV**: Only the best individual found in the search by $\mathcal{M}'$ is reevaluated using $\mathcal{M}$.
- **Paper V**: The best individual found by surrogate $\mathcal{M}'$ is reevaluated using $\mathcal{M}$, after every fixed number of (z) generations.

3.5 Evaluating results

Our approaches of evaluating the results differ based on whether the objective of the paper is to answer **RQ1** or **RQ2**. As the main goal of **RQ2** is to produce novel EC algorithms, we compare our proposed algorithms empirically against SOTA methods and baseline alternatives using benchmark datasets. We also carry sensitivity analysis for the main algorithm parameters and amortized analysis to quantify the computational savings of our algorithms.

In papers addressing **RQ1**, we carry ablation procedures to quantify the significance of the findings. We also conduct correlation analysis to identify novel insights and interesting configurations.

- **Paper I**:
  - Empirical analysis based on 14 datasets from the UCI ML repository comparing accuracy and run-time of the SAGA algorithm against a classical wrapper.
  - Amortized analysis of the computational cost showed that the upper-bound run-time of the SAGA algorithm is, in the worst case, equal to the wrapper, and it scales better for different choices of inductive algorithms.
  - Sensitivity analysis of the main parameters of SAGA controlling population reduction and migration strategies.
- **Paper II**:
  - Empirical analysis based on hybrid energy buses datasets, comparing GADIF against a number of wrapper and filter feature selection methods.
- **Paper III**:

  - Empirical analysis based on 13 datasets from UCI ML repository comparing accuracy and run-time of the CHC$_{QX}$ and PSO$_{QX}$ algorithm against classical wrappers CHC and PSO.
  - Amortized analysis of the computational cost showed that the cost of running our algorithm CHC$_{QX}$ is cheaper than CHC, as long as two algorithms run for at least 13 generations.
  - Sensitivity analysis of the main parameters of CHC$_{QX}$, including population size and evolution control frequency.

- **Paper IV**:

  - Empirical analysis based on 4 benchmark vision datasets.
  - Ablation study to quantify the contributions of different randomness techniques in different settings.
  - Correlation analysis to identify interesting configurations and interactions between randomness techniques.

- **Paper V**:

  - Empirical analysis based on 19 benchmark survival analysis datasets.
  - Ablation study to confirm the importance of the non-zero gradient for the censored cases.

4. Concluding Remarks

As the dance between EC and ML algorithms continues to evolve, it seems like both paradigms can greatly benefit from working jointly. The task of designing intelligent algorithms can be viewed as a bilevel optimization process in this mixture. In this process, EC methods search for an optimal initial configuration in the outer optimization, while the inner optimization focuses on analytically determining or using a gradient-based approach to fine-tune the parameters of the learning algorithm. Recent studies in the field of evolving parameterized activation functions and loss functions have demonstrated the advantages of combining elements from both EC and ML paradigms, resulting in improved algorithm performance [26; 29].

4.1 Conclusions

In this thesis, we have explored two research directions in EDL. First, we have proposed novel EC methods that allow estimation of the performance of the ML model $\mathcal{M}$ using a small subset of training instances. Our work extends the applicability of using EC to optimize $\mathcal{M}$ to learning tasks with big datasets. We have provided theoretical proofs that the run-time of our FS algorithms in **Paper I** and in **Paper III** are, in the worse case, equal to a counterpart FS wrapper. Empirically, our algorithms were **three times faster** than a wrapper, while arriving at feature subset solutions of **significantly higher accuracy**. Our ideas are not specific to a certain type of EC e.g., GA, we have tested the applicability to Swarm Intelligence (SI) with the results of PSO.

Our results show the benefit of using an imperfect (qualitative) approximation of the fitness evaluations for early explorations in ML optimizations. Our approximations were successful at quickly identifying interesting regions of the search space, **greatly reducing the number of expensive full-trainings of** $\mathcal{M}$.

We extend the applicability of EC to novel optimization problems in DL, not previously studied in the literature. Our **transfer-learning** algorithm in **Paper II** penalizes learning from features that are unlikely to generalize to domains with no labeled data. We showcase the task of modeling SOH for a population of hybrid energy buses, where our algorithm identified better feature subsets for unseen scenarios/domains e.g., different operations and configurations.

In our study of randomness techniques in training DNNs in **Paper IV**, we categorized existing randomness techniques based on type: data, network, and optimization; and based on purpose: regularization, data size, convergence, and training time. Our categorization allowed us to identify gaps in the coverage of techniques, as we propose two novel techniques: Loss Noise (LN) and Gradient Dropout (GD). Our empirical results showed that both methods lead to significant test error reductions in a variety of learning tasks.

The results of our ablation study showed that choosing the weight initialization scale, the starting learning rate, and how much to decay it during training, are the choices that lead to consistent test error reductions. The role of image augmentation is also important to improve performance, but the augmentation techniques are dataset-dependent, an augmentation policy must be optimized for the dataset.

Our results suggest that optimizers demonstrate different training dynamics. Vanilla SGD and Adam both showed the "expected" preference of starting with a high level of noise while decaying it as the training progressed. Interestingly, we observed that Adam showed a clear preference for steep decay of the learning rate, This observation contradicts a common intuition in the DL community that decaying the learning rate is not as necessary for adaptive optimizers due to their ability to use different step sizes for different parameters. Another surprising result is the one of SGD with Momentum that kept an almost constant scale of noise throughout training.

The analysis reveals that SGD with Momentum should be the preferred optimizer for scenarios involving large-batch training and few-shot learning. Conversely, when the primary goal is achieving the best performance, especially on challenging datasets, Vanilla SGD outperforms other options. It's worth noting, however, that proper hyperparameter selection is critical for Vanilla SGD due to its sensitivity to factors like initialization, learning rate, and batch size. Therefore, practitioners aiming for peak performance should steer clear of using Adam, as our study found its generalization performance to be significantly poorer compared to the SGD family of optimizers, especially on more demanding datasets. Nonetheless, one advantage of Adam is its flexibility in working with a wider range of learning rates and weight initialization values, which can be valuable when computational resources are limited for fine-tuning these parameters.

Another interesting finding was the superior performance of DropConnect compared to the more popular variant, Dropout, for fully connected networks. This finding suggests that DropConnect should be the preferred method for few-shoot and transfer learning scenarios where a pre-trained feature extractors are used.

4.2 A look into the future

It is also important to understand whether the long run-time problem of EDL will continue to be relevant in the future. We analyze the recent progress and trends of the three pillars: compute, data and algorithms to answer this question.

Our computing power was growing exponentially – even accelerating over the last decade, but is on the verge of slowing down. Historically, the progress of our computing power for traditional CPUs followed the Moore's law [100] predicting that the number of transistors on a microchip would double every 18 months. However, GPUs appeared to be growing at a faster rate than traditional CPUs; the Huang's law [101] suggests that GPU performance was at least tripling every two years, a rate much faster than Moore's law. However, the growth in the compute is bounded by the practical limits of physics, and is expected to plateau as quickly as in 2025 [102].

We may observe the growth in our datasets sizes for the vision benchmark datasets in Figure 4.1. The hand-written digits dataset MNIST was the largest labeled image dataset of its time in 1998, with 60K training instances, almost 10 years later, in 2009, the ImageNet dataset was released with more than 14M images. Today, the Google dataset JFT-300M [103] has over 300M images. Our benchmark datasets sizes has grown in size by more than 4 000 times within the last 20 years, almost doubling every 18 months. A similar exponential rate of growth is reported in papers from the big data literature [104; 105]. As this rate of growth is showing no sign of slowing down, we could expect the big techs to soon work with datasets of over billion images.

The current direction of advancing DL algorithms is pushing towards more data-hungry algorithms. A clear example is the Vision Transformers (ViT), showing superior performance compared to CNN for datasets with more than a hundred million training samples. A result explained based on the intuition that the lack of the inductive biases of ViT compared to CNNs is causing the higher performance. CNNs are designed with certain architectural assumptions that introduce bias in how they perceive and process visual information. An example of such bias is the one toward local features due to their use of convolutional filters with fixed receptive fields. ViTs, on the other hand, rely on the multi-head self-attention mechanism allowing them to capture long-range dependencies in the data without any pre-defined local receptive fields, as used in CNNs [7].

As these trends continue, in the near future, we will have larger datasets, and more data-hungry algorithms, but a slower rate of progress of the computing power. Extrapolating forward, the need to come up with efficient EC algorithms, or even training-free alternatives [106] for model evaluations is crucial to reduce the reliance on increases in computing power [107].

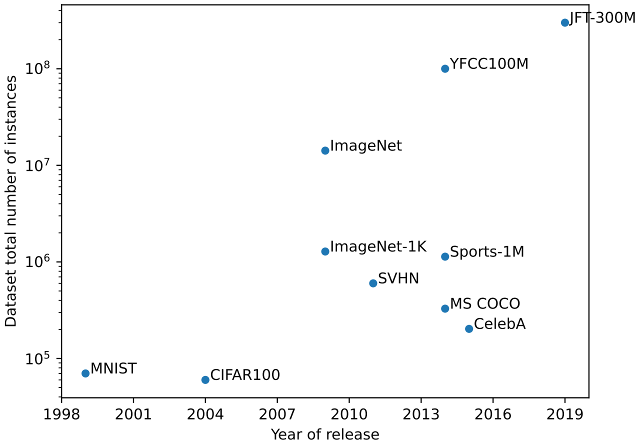


Figure 4.1: Computer vision benchmark datasets by size in terms of number of instances, ordered chronically from left to right based on their year of release.

4.3 Future Work

A number of DNN optimizations discussed in this thesis **Paper 2**, **Paper IV** and **Paper V** can be carried in a coevolution setting. The use of coevolution could potentially maximize the benefit of potential interactions between different design decisions. It would be interesting to analyze such interactions between e.g., loss and activation functions.

The real-world applications of DL rarely consider one measure of success e.g., accuracy. In our work of optimizing activation functions in **Paper 2**, we evaluated candidate activations according to their accuracy. It would be interesting to carry out the search while considering another competing objective e.g., energy consumption in a Multi-Objective Optimization (MOO) setting. The search could potentially identify high-performing yet computationally cheap functions to use with a simpler model with fewer layers outperforming an over-parameterized deep DNN with ReLU units, similar ideas of using activation functions to achieve more computationally efficient DNN architectures are an active research direction in the literature of trainable activation functions [108].

The Genetic Programming (GP) algorithm we have used to discover loss and activation functions in **Paper 2** and **Paper V**, can be used to discover schedules for other design choices in DNN. Many of the hyper-parameters are held constant throughout the training e.g., label smoothing, and the GP-based algorithm could be used to test the null hypothesis "holding a constant value of label smoothing throughout the training is optimal". Also, we could use it to identify task-specific schedules for hyper-parameters that generally benefit from decaying e.g, learning rate or increasing batch size, while not knowing exactly what schedule to use e.g., exponential or step-function [109].

A possible extension of the work in **Paper I** and in **Paper IV** is to use an approximate model $\mathcal{M}'$ to estimate the performance of model $\mathcal{M}$ for the learning task $\mathcal{T}$, where $\mathcal{M}'$ belong to a different learning algorithm $\mathcal{A} \neq \mathcal{A}'$. A motivation is to choose $\mathcal{A}'$ based on the characteristic of the dataset of $\mathcal{T}$ to ensure that evaluations with $\mathcal{M}'$ require less time than ones with $\mathcal{M}$. As an example, a DT model would be a good approximation choice for datasets with a large number of instances, given its linear complexity $O(nk^2)$ towards the number of training instances. kNN is the opposite as it scales linearly with more features $O(n^2k)$, but would struggle with a large number of instances. An important aspect is to consider the differences in the inductive biases between $\mathcal{A}$ and $\mathcal{A}'$. Intuitively, DT could be a useful approximate model $\mathcal{M}'$ to select features for model $\mathcal{M}$ as Random Forest (RF), as both learning algorithms share similarities in their assumptions. It is unclear whether the same DT approximate model would be of any use to select feature subsets for a learning task where $\mathcal{M}$ is SVM. The same thinking applies to DL, as Multi-layer Perceptrons (MLPs) are often used as test-beds for new ideas and techniques. Again, the usefulness of an MLP-based $\mathcal{M}'$ to optimize e.g., the image augmentation policy of ViT-based model $\mathcal{M}$, depends on $\mathcal{M}'$ ability to rank solutions similarly to $\mathcal{M}$.

References

[1] MOBYEN UDDIN AHMED, MOHAMMED GHAITH ALTARABICHI, SHAHINA BEGUM, FREDRIK GINSBERG, ROBERT GLAES, MAGNUS ÖSTGREN, HAMIDUR RAHMAN, AND MAGNUS SORENSEN. **A vision-based indoor navigation system for individuals with visual impairment**. *International Journal of Artificial Intelligence*, **17**(2):188–201, 2019. vi

[2] MOHAMMED GHAITH ALTARABICHI, MOBYEN UDDIN AHMED, MARIA RITA CICERI, STEFANIA BALZAROTTI, FEDERICA BIASSONI, DEBORA LOMBARDI, AND PAOLO PEREGO. **Reaction Time Variability Association with Unsafe Driving**. In *Transport Research Arena TRA2020, 27 Apr 2020, Helsinki, Finland*, 2020. vi

[3] MOHAMMED GHAITH ALTARABICHI, YUANTAO FAN, SEPIDEH PASHAMI, SŁAWOMIR NOWACZYK, AND THORSTEINN RÖGNVALDSSON. **Predicting state of health and end of life for batteries in hybrid energy buses**. In *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference :*, pages 1231–1231, 2020. vi

[4] MOHAMMED GHAITH ALTARABICHI, PEYMAN SHEIKHOLHARAM MASHHADI, YUANTAO FAN, SEPIDEH PASHAMI, SŁAWOMIR NOWACZYK, PABLO DEL MORAL, MAHMOUD RAHAT, AND THORSTEINN RÖGNVALDSSON. **Stacking ensembles of heterogenous classifiers for fault detection in evolving environments**. In *30th European Safety and Reliability Conference, ESREL 2020 and 15th Probabilistic Safety Assessment and Management Conference, PSAM15 2020, Venice, Italy, 1-5 November, 2020*, pages 1068–1068. Research Publishing Services, 2020. vii

[5] ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND GEOFFREY E HINTON. **Imagenet classification with deep convolutional neural networks**. *Advances in neural information processing systems*, **25**, 2012. 1

[6] KAIMING HE, XIANGYU ZHANG, SHAOQING REN, AND JIAN SUN. **Deep residual learning for image recognition**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 5

[7] ALEXEY DOSOVITSKIY, LUCAS BEYER, ALEXANDER KOLESNIKOV, DIRK WEISSENBORN, XIAOHUA ZHAI, THOMAS UNTERTHINER, MOSTAFA DEHGHANI, MATTHIAS MINDERER, GEORG HEIGOLD, SYLVAIN GELLY, ET AL. **An image is worth 16x16 words: Transformers for image recognition at scale**. *arXiv preprint arXiv:2010.11929*, 2020. 1, 27

[8] ILYA SUTSKEVER, ORIOL VINYALS, AND QUOC V LE. **Sequence to sequence learning with neural networks**. *Advances in neural information processing systems*, **27**, 2014. 1

[9] ASHISH VASWANI, NOAM SHAZEER, NIKI PARMAR, JAKOB USZKOREIT, LLION JONES, AIDAN N GOMEZ, ŁUKASZ KAISER, AND ILLIA POLOSUKHIN. **Attention is all you need**. *Advances in neural information processing systems*, **30**, 2017. 1

[10] JACOB DEVLIN, MING-WEI CHANG, KENTON LEE, AND KRISTINA TOUTANOVA. **Bert: Pre-training of deep bidirectional transformers for language understanding**. *arXiv preprint arXiv:1810.04805*, 2018. 1

[11] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. **Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs**. *jama*, **316**(22):2402–2410, 2016. 1

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. **Playing atari with deep reinforcement learning**. *arXiv preprint arXiv:1312.5602*, 2013. 1

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. **Human-level control through deep reinforcement learning**. *nature*, **518**(7540):529–533, 2015. 1

[14] Risto Miikkulainen, Jason Liang, Elliot Meyerson, Aditya Rawal, Daniel Fink, Olivier Francon, Bala Raju, Hormoz Shahrzad, Arshak Navruzyan, Nigel Duffy, et al. **Evolving deep neural networks**. In *Artificial intelligence in the age of neural networks and brain computing*, pages 293–312. Elsevier, 2019. 1, 6

[15] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, **86**(11):2278–2324, 1998. 1

[16] Sepp Hochreiter and Jürgen Schmidhuber. **Long short-term memory**. *Neural computation*, **9**(8):1735–1780, 1997. 1

[17] Vinod Nair and Geoffrey E Hinton. **Rectified linear units improve restricted boltzmann machines**. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 1

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. **Dropout: a simple way to prevent neural networks from overfitting**. *The journal of machine learning research*, **15**(1):1929–1958, 2014. 1

[19] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. **Deep information propagation**. *arXiv preprint arXiv:1611.01232*, 2016. 1, 4, 5

[20] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019. 4, 17, 19

[21] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. **On the impact of the activation function on deep neural networks training**. In *International conference on machine learning*, pages 2672–2680. PMLR, 2019. 1, 4, 5

[22] Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. **A survey on evolutionary computation approaches to feature selection**. *IEEE Transactions on evolutionary computation*, **20**(4):606–626, 2015. 1, 3, 6, 15

[23] Yuqiao Liu, Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Kay Chen Tan. **A survey on evolutionary neural architecture search**. *IEEE transactions on neural networks and learning systems*, 2021. 1, 2, 4, 6

[24] James Bergstra and Yoshua Bengio. **Random search for hyper-parameter optimization.** *Journal of machine learning research*, **13**(2), 2012. 1, 5, 17

[25] Garrett Bingham, William Macke, and Risto Miikkulainen. **Evolutionary optimization of deep learning activation functions**. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 289–296, 2020. 1, 2, 4, 5, 20

[26] Garrett Bingham and Risto Miikkulainen. **Discovering parametric activation functions**. *Neural Networks*, **148**:48–65, 2022. 1, 2, 25

[27] Santiago Gonzalez and Risto Miikkulainen. **Improved training speed, accuracy, and data utilization through loss function optimization**. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020. 1, 2, 6

[28] Santiago Gonzalez and Risto Miikkulainen. **Optimizing loss functions through multivariate taylor polynomial parameterization**. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 305–313, 2021.

[29] Christian Raymond, Qi Chen, Bing Xue, and Mengjie Zhang. **Fast and Efficient Local-Search for Genetic Programming Based Loss Function Learning**. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1184–1193, 2023. 1, 2, 25

[30] Larry J Eshelman. **The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination**. In *Foundations of genetic algorithms*, **1**, pages 265–283. Elsevier, 1991. 2, 21

[31] Prabhat Hajela. **Genetic search-an approach to the nonconvex optimization problem**. *AIAA journal*, **28**(7):1205–1210, 1990. 2

[32] Manzil Zaheer, Sashank Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. **Adaptive methods for nonconvex optimization**. *Advances in neural information processing systems*, **31**, 2018. 2

[33] Abdullah Konak, David W Coit, and Alice E Smith. **Multi-objective optimization using genetic algorithms: A tutorial**. *Reliability engineering & system safety*, **91**(9):992–1007, 2006. 2

[34] Zhi-Hui Zhan, Lin Shi, Kay Chen Tan, and Jun Zhang. **A survey on evolutionary computation for complex continuous optimization**. *Artificial Intelligence Review*, pages 1–52, 2022. 2

[35] Zhi-Hui Zhan, Jun Zhang, Ying Lin, Jian-Yu Li, Ting Huang, Xiao-Qi Guo, Feng-Feng Wei, Sam Kwong, Xin-Yi Zhang, and Rui You. **Matrix-based evolutionary computation**. *IEEE Transactions on Emerging Topics in Computational Intelligence*, **6**(2):315–328, 2021. 2, 3

[36] Jason Liang, Elliot Meyerson, Babak Hodjat, Dan Fink, Karl Mutch, and Risto Miikkulainen. **Evolutionary neural automl for deep learning**. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 401–409, 2019. 2

[37] Esteban Real, Chen Liang, David So, and Quoc Le. **Automl-zero: Evolving machine learning algorithms from scratch**. In *International conference on machine learning*, pages 8007–8019. PMLR, 2020. 2

[38] Kenneth O Stanley and Risto Miikkulainen. **Evolving neural networks through augmenting topologies**. *Evolutionary computation*, **10**(2):99–127, 2002. 2

[39] Kenneth O Stanley, Jeff Clune, Joel Lehman, and Risto Miikkulainen. **Designing neural networks through neuroevolution**. *Nature Machine Intelligence*, **1**(1):24–35, 2019. 2

[40] Pedro Carvalho, Nuno Lourenço, Filipe Assunção, and Penousal Machado. **AutoLR: an evolutionary approach to learning rate policies**. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 672–680, 2020. 2

[41] Jiang Su and Harry Zhang. **A fast decision tree learning algorithm**. In *Aaai*, **6**, pages 500–505, 2006. 3

[42] FRANK Z BRILL, DONALD E BROWN, AND WORTHY N MARTIN. **Fast generic selection of features for neural network classifiers**. *IEEE Transactions on Neural Networks*, **3**(2):324–328, 1992. 3, 16

[43] LÉON BOTTOU AND CHIH-JEN LIN. **Support vector machine solvers**. *Large scale kernel machines*, **3**(1):301–320, 2007. 3

[44] CHIH-FONG TSAI, WILLIAM EBERLE, AND CHI-YUAN CHU. **Genetic algorithms in feature and instance selection**. *Knowledge-Based Systems*, **39**:240–247, 2013. 3

[45] LOUIS E YELLE. **The learning curve: Historical review and comprehensive survey**. *Decision sciences*, **10**(2):302–328, 1979. 4

[46] JIA DENG, WEI DONG, RICHARD SOCHER, LI-JIA LI, KAI LI, AND LI FEI-FEI. **Imagenet: A large-scale hierarchical image database**. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 4

[47] DONG LING TONG AND ROBERT MINTRAM. **Genetic Algorithm-Neural Network (GANN): a study of neural network activation functions and depth of genetic algorithm search applied to feature selection**. *International Journal of Machine Learning and Cybernetics*, **1**:75–87, 2010. 5

[48] RICHARD BELLMAN AND ROBERT KALABA. **On adaptive control processes**. *IRE Transactions on Automatic Control*, **4**(2):1–9, 1959. 5, 19

[49] JASPER SNOEK, HUGO LAROCHELLE, AND RYAN P ADAMS. **Practical bayesian optimization of machine learning algorithms**. *Advances in neural information processing systems*, **25**, 2012. 5, 17

[50] JIA WU, XIU-YUN CHEN, HAO ZHANG, LI-DONG XIONG, HANG LEI, AND SI-HAO DENG. **Hyperparameter optimization for machine learning models based on Bayesian optimization**. *Journal of Electronic Science and Technology*, **17**(1):26–40, 2019. 5, 17

[51] IRWAN BELLO, BARRET ZOPH, VIJAY VASUDEVAN, AND QUOC V LE. **Neural optimizer search with reinforcement learning**. In *International Conference on Machine Learning*, pages 459–468. PMLR, 2017. 5, 20

[52] PRAJIT RAMACHANDRAN, BARRET ZOPH, AND QUOC V LE. **Searching for activation functions**. *arXiv preprint arXiv:1710.05941*, 2017. 5, 6, 19, 20

[53] KAIMING HE, XIANGYU ZHANG, SHAOQING REN, AND JIAN SUN. **Delving deep into rectifiers: Surpassing human-level performance on imagenet classification**. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 5

[54] ARVIND NEELAKANTAN, LUKE VILNIS, QUOC V LE, ILYA SUTSKEVER, LUKASZ KAISER, KAROL KURACH, AND JAMES MARTENS. **Adding gradient noise improves learning for very deep networks**. *arXiv preprint arXiv:1511.06807*, 2015. 5

[55] SERGEY IOFFE AND CHRISTIAN SZEGEDY. **Batch normalization: Accelerating deep network training by reducing internal covariate shift**. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 5

[56] HANXIAO LIU, KAREN SIMONYAN, ORIOL VINYALS, CHRISANTHA FERNANDO, AND KORAY KAVUKCUOGLU. **Hierarchical representations for efficient architecture search**. *arXiv preprint arXiv:1711.00436*, 2017. 6

[57] BERND BISCHL, MARTIN BINDER, MICHEL LANG, TOBIAS PIELOK, JAKOB RICHTER, STEFAN COORS, JANEK THOMAS, THERESA ULLMANN, MARC BECKER, ANNE-LAURE BOULESTEIX, ET AL. **Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges**. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **13**(2):e1484, 2023. 6

[58] ZHI-HUI ZHAN, JIAN-YU LI, AND JUN ZHANG. **Evolutionary deep learning: A survey**. *Neurocomputing*, **483**:42–58, 2022. 6, 15

[59] JOEL LEHMAN, JEFF CLUNE, DUSAN MISEVIC, CHRISTOPH ADAMI, LEE ALTENBERG, JULIE BEAULIEU, PETER J BENTLEY, SAMUEL BERNARD, GUILLAUME BESLON, DAVID M BRYSON, ET AL. **The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities**. *Artificial life*, **26**(2):274–306, 2020. 6

[60] ARILD NØKLAND. **Direct feedback alignment provides learning in deep neural networks**. *Advances in neural information processing systems*, **29**, 2016. 7

[61] CAGLAR GULCEHRE, MARCIN MOCZULSKI, MISHA DENIL, AND YOSHUA BENGIO. **Noisy activation functions**. In *International conference on machine learning*, pages 3059–3068. PMLR, 2016. 7

[62] XAVIER GLOROT, ANTOINE BORDES, AND YOSHUA BENGIO. **Deep sparse rectifier neural networks**. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011. 7

[63] MOHAMMED GHAITH ALTARABICHI, SŁAWOMIR NOWACZYK, SEPIDEH PASHAMI, AND PEYMAN SHEIKHOLHARAM MASHHADI. **Surrogate-assisted genetic algorithm for wrapper feature selection**. In *2021 IEEE congress on evolutionary computation (CEC)*, pages 776–785. IEEE, 2021. 8

[64] MOHAMMED GHAITH ALTARABICHI, YUANTAO FAN, SEPIDEH PASHAMI, PEYMAN SHEIKHOLHARAM MASHHADI, AND SŁAWOMIR NOWACZYK. **Extracting Invariant Features for Predicting State of Health of Batteries in Hybrid Energy Buses**. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–6. IEEE, 2021. 9

[65] MOHAMMED GHAITH ALTARABICHI, SŁAWOMIR NOWACZYK, SEPIDEH PASHAMI, AND PEYMAN SHEIKHOLHARAM MASHHADI. **Fast Genetic Algorithm for feature selection—A qualitative approximation approach**. *Expert systems with applications*, **211**, 2023. 10

[66] MOHAMMED GHAITH ALTARABICHI, SŁAWOMIR NOWACZYK, SEPIDEH PASHAMI, AND PEYMAN SHEIKHOLHARAM MASHHADI. **Fast Genetic Algorithm for feature selection—A qualitative approximation approach**. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, pages 11–12, 2023. 10

[67] STEPHAN DEMPE AND ALAIN ZEMKOHO. **Bilevel optimization**. In *Springer optimization and its applications*, **161**. Springer, 2020. 15

[68] MARK ANDREW HALL ET AL. **Correlation-based feature selection for machine learning**. 1999. 16

[69] KANCHAN JHA AND SRIPARNA SAHA. **Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique**. *Applied Soft Computing*, **98**:106823, 2021. 16

[70] HONGFANG ZHOU, XIQIAN WANG, AND ROUROU ZHU. **Feature selection based on mutual information with correlation coefficient**. *Applied Intelligence*, **52**(5):5457–5474, 2022. 16

[71] LIN SUN, TENGYU YIN, WEIPING DING, YUHUA QIAN, AND JIUCHENG XU. **Multilabel feature selection using ML-ReliefF and neighborhood mutual information for multilabel neighborhood decision systems**. *Information Sciences*, **537**:401–424, 2020. 16

[72] QUANQUAN GU, ZHENHUI LI, AND JIAWEI HAN. **Generalized fisher score for feature selection**. *arXiv preprint arXiv:1202.3725*, 2012. 16

[73] HUAN LIU, RUDY SETIONO, ET AL. **A probabilistic approach to feature selection-a filter solution**. In *ICML*, **96**, pages 319–327. Citeseer, 1996. 16

[74] PIER LUCA LANZI. **Fast feature selection with genetic algorithms: a filter approach**. In *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, pages 537–540. IEEE, 1997. 16

[75] MANOSIJ GHOSH, SUKDEV ADHIKARY, KUSHAL KANTI GHOSH, ARITRA SARDAR, SHEMIM BEGUM, AND RAM SARKAR. **Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods**. *Medical & biological engineering & computing*, **57**(1):159–176, 2019. 16

[76] YISHI ZHANG, SHUJUAN LI, TENG WANG, AND ZIGANG ZHANG. **Divergence-based feature selection for separate classes**. *Neurocomputing*, **101**:32–42, 2013. 16

[77] ALAN JOVIĆ, KARLA BRKIĆ, AND NIKOLA BOGUNOVIĆ. **A review of feature selection methods with applications**. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. Ieee, 2015. 16

[78] NAOUAL EL ABOUDI AND LAILA BENHLIMA. **Review on wrapper feature selection approaches**. In *2016 International Conference on Engineering & MIS (ICEMIS)*, pages 1–5. IEEE, 2016. 16

[79] ANDREA BOMMERT, XUDONG SUN, BERND BISCHL, JÖRG RAHNENFÜHRER, AND MICHEL LANG. **Benchmark for filter methods for feature selection in high-dimensional classification data**. *Computational Statistics & Data Analysis*, **143**:106839, 2020. 16

[80] FENG TAN, XUEZHENG FU, YANQING ZHANG, AND ANU G BOURGEOIS. **A genetic algorithm-based method for feature subset selection**. *Soft Computing*, **12**(2):111–120, 2008. 16

[81] STJEPAN ORESKI AND GORAN ORESKI. **Genetic algorithm-based heuristic for feature selection in credit risk assessment**. *Expert systems with applications*, **41**(4):2052–2064, 2014.

[82] POOJA RANI, RAJNEESH KUMAR, ANURAG JAIN, AND SUNIL KUMAR CHAWLA. **A hybrid approach for feature selection based on genetic algorithm and recursive feature elimination**. *International Journal of Information System Modeling and Design (IJISMD)*, **12**(2):17–38, 2021.

[83] HAO SUN, JING JIN, REN XU, AND ANDRZEJ CICHOCKI. **Feature selection combining filter and wrapper methods for motor-imagery based brain–computer interfaces**. *International Journal of Neural Systems*, **31**(09):2150040, 2021.

[84] XIAN-FANG SONG, YONG ZHANG, DUN-WEI GONG, AND XIAO-ZHI GAO. **A fast hybrid feature selection based on correlation-guided clustering and particle swarm optimization for high-dimensional data**. *IEEE Transactions on Cybernetics*, 2021. 16

[85] HUONG THANH LE, LUAN VAN TRAN, XUAN HOAI NGUYEN, AND THI HIEN NGUYEN. **Optimizing genetic algorithm in feature selection for named entity recognition**. In *Proceedings of the Sixth International Symposium on Information and Communication Technology*, pages 11–16, 2015. 16

[86] DANIEL PERALTA, SARA DEL RÍO, SERGIO RAMÍREZ-GALLEGO, ISAAC TRIGUERO, JOSE M BENITEZ, AND FRANCISCO HERRERA. **Evolutionary feature selection for big data classification: A mapreduce approach**. *Mathematical Problems in Engineering*, **2015**, 2015.

[87] JOAQUÍN DERRAC, SALVADOR GARCÍA, AND FRANCISCO HERRERA. **A first study on the use of coevolutionary algorithms for instance and feature selection**. In *International conference on hybrid artificial intelligence systems*, pages 557–564. Springer, 2009. 16

[88] MARC CLAESEN AND BART DE MOOR. **Hyperparameter search in machine learning**. *arXiv preprint arXiv:1502.02127*, 2015. 17

[89] Petro Liashchynskyi and Pavlo Liashchynskyi. **Grid search, random search, genetic algorithm: A big comparison for NAS. arXiv 2019**. *arXiv preprint arXiv:1912.06059*. 17

[90] Li Yang and Abdallah Shami. **On hyperparameter optimization of machine learning algorithms: Theory and practice**. *Neurocomputing*, **415**:295–316, 2020. 17

[91] Marc-André Zöller and Marco F Huber. **Benchmark and survey of automated machine learning frameworks**. *Journal of artificial intelligence research*, **70**:409–472, 2021. 17

[92] Michael Meissner, Michael Schmuker, and Gisbert Schneider. **Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training**. *BMC bioinformatics*, **7**(1):1–11, 2006. 17

[93] Xueli Xiao, Ming Yan, Sunitha Basodi, Chunyan Ji, and Yi Pan. **Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm**. *arXiv preprint arXiv:2006.12703*, 2020. 17

[94] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. **The irace package: Iterated racing for automatic algorithm configuration**. *Operations Research Perspectives*, **3**:43–58, 2016. 17

[95] Pablo Ribalta Lorenzo, Jakub Nalepa, Michal Kawulok, Luciano Sanchez Ramos, and José Ranilla Pastor. **Particle swarm optimization for hyper-parameter selection in deep neural networks**. In *Proceedings of the genetic and evolutionary computation conference*, pages 481–488, 2017. 17

[96] David H Wolpert and William G Macready. **No free lunch theorems for optimization**. *IEEE transactions on evolutionary computation*, **1**(1):67–82, 1997. 20

[97] DR Ramdania, M Irfan, F Alfarisi, and D Nuraiman. **Comparison of genetic algorithms and Particle Swarm Optimization (PSO) algorithms in course scheduling**. In *Journal of Physics: Conference Series*, **1402**, page 022079. IOP Publishing, 2019. 20

[98] FD Wihartiko, H Wijayanti, and F Virgantari. **Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem**. In *IOP conference series: materials science and engineering*, **332**, page 012020. IOP Publishing, 2018. 20

[99] Yaochu Jin. **A comprehensive survey of fitness approximation in evolutionary computation**. *Soft computing*, **9**(1):3–12, 2005. 22

[100] Robert R Schaller. **Moore's law: past, present and future**. *IEEE spectrum*, **34**(6):52–59, 1997. 27

[101] Tekla S Perry. **Move over, Moore's law. Make way for Huang's law [Spectral Lines]**. *IEEE Spectrum*, **55**(5):7–7, 2018. 27

[102] John Shalf. **The future of computing beyond Moore's Law**. *Philosophical Transactions of the Royal Society A*, **378**(2166):20190061, 2020. 27

[103] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. **Revisiting unreasonable effectiveness of data in deep learning era**. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 27

[104] Changqing Ji, Yu Li, Wenming Qiu, Yingwei Jin, Yujie Xu, Uchechukwu Awada, Keqiu Li, and Wenyu Qu. **Big data processing: Big challenges and opportunities**. *Journal of Interconnection Networks*, **13**(03n04):1250009, 2012. 27

[105] Min Chen, Shiwen Mao, and Yunhao Liu. **Big data: A survey**. *Mobile networks and applications*, **19**:171–209, 2014. 27

[106] Joe Mellor, Jack Turner, Amos Storkey, and Elliot J Crowley. **Neural architecture search without training**. In *International Conference on Machine Learning*, pages 7588–7598. PMLR, 2021. 28

[107] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. **The computational limits of deep learning**. *arXiv preprint arXiv:2007.05558*, 2020. 28

[108] Andrea Apicella, Francesco Donnarumma, Francesco Isgrò, and Roberto Prevete. **A survey on modern trainable activation functions**. *Neural Networks*, **138**:14–32, 2021. 28

[109] Sergey Zagoruyko and Nikos Komodakis. **Wide residual networks**. *arXiv preprint arXiv:1605.07146*, 2016. 29

Mohammed Ghaith Altarabichi

“Mohammed Ghaith holds a Master’s degree in Computer Science from HKR University in Sweden, and will soon defend his PhD with Halmstad University in Sweden. His research is focused on using Evolutionary Computation (EC) methods to design and optimize Deep Learning (DL) models. Mohammed Ghaith has developed feature selection, instance selection, and hyper-parameter tuning algorithms to improve the performance of DL models. Additionally, he has explored optimizing other design decisions of DL algorithms like loss functions for survival analysis and used EC algorithms to uncover insights about complex interactions between design choices for DL models in computer vision. His work is published in top journals (Expert Systems With Applications), conferences (IEEE CEC, ACM GECCO, IEEE DSAA) within the field of AI and he has received a number of recognitions (2nd place in the ESREL 2020 conference AI competition) and awards (Global Swede 2017) for research excellence.“

School of Information Technology

ISBN: 978-91-89587-32-8 (printed)
Halmstad University Dissertations, 2024

HALMSTAD
UNIVERSITY