Evaluating the Effectiveness and Stability of the Constrained Hybrid Metaheuristic Algorithm in Probabilistic Neural Networks Training

Szymon Kucharczyk[1,3][0009−0002−2413−6984],
Piotr A. Kowalski[1,2][0000−0003−4041−6900],
Jacek Mańdziuk[4,5][0000−0003−0947−028X]

[1] Faculty of Physics and Applied Computer Science, AGH University of Krakow,
[2] Systems Research Institute, Polish Academy of Sciences,
[3] AGH Doctoral School, AGH University of Krakow,
[4] Faculty of Mathematics and Information Science, Warsaw University of Technology,
[5] Faculty of Computer Science, AGH University of Krakow,
kucharcz@agh.edu.pl, pkowal@agh.edu.pl, jacek.mandziuk@pw.edu.pl

Abstract. Probabilistic Neural Networks (PNNs) are memory-based networks that have been used successfully for classification and regression tasks. Training of PNNs may be performed by analytical methods, e.g., plug-in or by heuristic methods, e.g., Particle Swarm Optimization. Heuristic methods are superior to traditional PNN training techniques because of their nonparametric behavior, independence of the PNN kernel selection, and ability to optimize method parameters for a given problem. One of the recently proposed training algorithms - *constrained Hybrid Metaheuristic* (cHM), overperformed other analytical and heuristic procedures on a variety of datasets. Here, we present a further evaluation of the cHM method for training PNNs for classification tasks. In particular, we study its effectiveness and stability with different hyperparameters across 10 datasets. The results show that the cHM training procedure is generally stable and the parameter selection does not significantly impact the PNN training accuracy for 8 of 10 tested datasets.

Keywords: Probabilistic Neural Networks · learning procedure · metaheuristic · hybrid metaheuristic · synergy· hyperparameter optimization

1 Introduction

Probabilistic Neural Networks constitute a family of neural models that explicitly embed probabilistic reasoning into learning, making them well-suited for classification and regression tasks [18]. Their core idea is to approximate class-conditional probability distributions from data, usually through kernel density estimation (KDE), and then assign a label by selecting the class that maximizes the estimated likelihood. In contrast to conventional neural networks that learn decision boundaries via iterative weight optimization (e.g., backpropagation),

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

PNNs assess how probable it is that a given input originates from each class distribution and make decisions directly from these distributional comparisons.

PNNs are valued in settings where uncertainty and noise are intrinsic, because probability distributions naturally express ambiguity and support robust decisions. At the same time, their probabilistic structure promotes interpretability by linking outputs to explicit density estimates and prototype contributions, which aligns well with explainability requirements [9] and uncertainty-aware modelling [8]. Despite remaining computational challenges, continued work on optimization and hybridization indicates that PNNs are likely to remain a relevant component of explainable and probabilistic AI.

1.1 Hyperparameter Optimization

Hyperparameter optimization (HPO) is a critical factor in metaheuristic algorithms. Not only does it provide control over the behavior of the algorithm, but it also helps maximize metaheuristic performance for a given problem [3]. The HPO techniques for metaheuristics can be divided into three categories. Simple generate-evaluate methods generate candidate hyperparameter sets and evaluate them all. Iterative generate-evaluate methods perform the generation and evaluation phases repeatedly, testing a new set of hyperparameters in each iteration. A different approach to HPO uses hyperheuristic methods that rely on self-adaptation and hybridization of a few algorithms, aiming at their optimal selection and parameterization. Popular approaches of this type include self-adaptation and hybridization of the Particle Swarm Optimization (PSO) [15, 14]. Also, probabilistic models have become very popular in the area of HPO thanks to their ability to model uncertainty in the objective function [1, 17].

In this work, the first type of HPO was selected, particularly a GridSearch optimization [2]. This brute-force technique starts with a definition of all possible parameter values to test. Next, the Cartesian product C of all these parameters is produced. Then, the GridSearch algorithm evaluates all parameters from C in the same way. In the end, the evaluation run with the highest performance metric is considered to be the optimal value of the hyperparameters for the given algorithm. Generally, this method can suffer from the curse of dimensionality [2] when the parts of the Cartesian product are large. However, for this study, the parameter set space was selected to omit the issue and test a reasonable space of possible cHM parameter values.

1.2 Motivation and Contribution

The cHM algorithm [7] demonstrated superior performance in PNN training, surpassing both single metaheuristics and traditional statistical methods in terms of accuracy. As a computational intelligence approach, cHM's performance can be further enhanced through HPO. Investigating the impact of HPO on this algorithm offers valuable insights into optimal parameter selection. Moreover, such an analysis shows the algorithm's stability across varying hyperparameter configurations when applied to diverse datasets.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

While this study investigates the cHM parameterizations across multiple datasets, it does not aim to compare the computational cost or predictive accuracy of the proposed approach with other PNN training techniques, including conventional and heuristic methods. Instead, the analysis concentrates on the internal parameters of the hybrid method and examines their impact on the overall performance.

The main contributions of this paper are three-fold: **(1)** We conduct a thorough HPO for the cHM method across 10 datasets; **(2)** The best cHM parameters for each dataset are proposed; **(3)** The dependence of the cHM algorithm on the number of iterations, probing, and fit constraints is analyzed.

1.3 Related work

The training process of PNNs, particularly in determining smoothing parameters, can be divided into two main categories, each with distinct characteristics. The first category includes deterministic methods, e.g. the plug-in [19] procedure. These approaches focus on minimizing the error between the estimated and probability density functions, ensuring that the estimated density closely matches the actual data distribution. As deterministic methods, they consistently produce the same results for the same input data and are particularly effective when the primary objective is accurate density estimation. The second category comprises non-deterministic approaches, including meta-heuristic algorithms like Genetic Algorithms, PSO, Simulated Annealing (SA) and constrained Hybrid Metaheuristic algorithm, as well as reinforcement learning [11] techniques. These methods are tailored to optimize classification performance by adjusting the smoothing parameter to maximize class separability. Although less stable than statistical methods, they offer greater flexibility and the potential to achieve superior results in individual training runs, especially in complex classification tasks where optimal class distinction is crucial. By leveraging their adaptability, non-deterministic methods can outperform statistical techniques under specific conditions. They can deliver exceptional classification accuracy when carefully tuned, making them valuable in scenarios where class separability is a priority [7, 10].

2 Methods

When using heuristics, particularly swarm-based algorithms, to train PNNs, each individual within the population is represented as a vector of proposed smoothing parameters. Specifically, an individual corresponds to a vector that contains parameters necessary to configure a PNN for a given dataset [10].

2.1 constrained Hybrid Metaheuristic

Figure 1 presents the flow diagram of the PNN training with the cHM algorithm. Generally, the training algorithm combines k single (inner) metaheuristics with

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

a shared population to ensemble them into one hybrid training method [7]. It consists of 2 stages: probing and fit. Before the cHM algorithm starts, its internal parameters are initialized:

- $maxFE_{probing}$ ($maxFE_p$) - number of maximum cost function evaluations during the probing phase [20]. This parameter is incremented for each individual for each test sample in the dataset.
- $maxFE_{fit}$ ($maxFE_f$) - maximal number of cost function evaluations during the fit phase [20]. The incrementation strategy is similar to the $maxFE_p$ one.
- n - number of probing and fit phases.
- Random initialization of the population θ, the same for each inner metaheuristic.

Next, the algorithm is executed for n iterations. Firstly, the probing phase is run. In this step, each kth metaheuristic is used to train a PNN until the number of cost function evaluations does not exceed the $maxFE_p$ criteria for each of the k methods. After probing, the population θ_{best} from the best-performing method, in terms of the cost function value, is passed to the fit phase. In the fit phase, the metaheuristic that produced θ_{best} is used for further PNN training. The training continues until the number of cost function evaluations does not exceed the $maxFE_f$ or the PNN training convergence condition is met. In the end, after n probing and fit phases, the best individual from the best population is taken as a final smoothing parameter vector for the PNN construction [7].

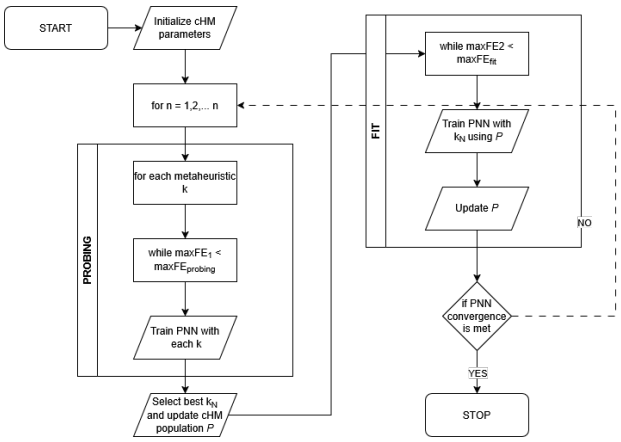

Fig. 1: Block diagram of PNN training using the cHM algorithm.

In this work, a group of five metaheuristic methods is evaluated as the inner-optimization methods of the cHM algorithm: PSO [5], the Bat algorithm (BAT) [12], Bacterial Foraging Optimization (BFO) [16], SA [6], and the Flower Pollination Algorithm (FPA) [10]. These global optimization techniques are well-established in the literature [13]. Following [7], we adopted the standard versions of these metaheuristics.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

2.2 Datasets and experiment parameters

In this work, PNNs are constructed using the Cauchy kernel, incorporating distinct smoothing parameters for each feature vector within the dataset. The cHM algorithm is utilized to train the PNNs for classification tasks. The cHM method is initialized with a randomly generated population of individuals, where each individual represents a set of smoothing parameters required for building the PNN. The initial population comprises 20 individuals with parameter values constrained to real numbers in the range [0,10]. The same population is employed to initialize each inner optimizer within the cHM procedure. Following [7], the error rate is used as the cost function.

When training PNNs using computational intelligence techniques, the smoothing parameter values during optimization must be constrained due to PNN architecture requirements. The possible values for these parameters are restricted to the interval [0,10000] of real numbers. If a smoothing parameter is found to be negative, the reflection technique [4] is used to ensure that the parameter value is adjusted to be within the permissible positive range.

The specific parameters of the cHM algorithm are summarized in Section 3, while the parameters used for each metaheuristic within the cHM framework are detailed in [7]. The cost function convergence threshold is set to $1e-8$.

The performance of PNN training with the cHM algorithm is evaluated on 10 well-known datasets taken UCI repository. Each dataset is stratified into train and test subsets, with the size of the test set comprising 20% of the total dataset. The training process is repeated 10 times for each experiment configuration, for each dataset, enabling the calculation of cumulative (average) metrics, including accuracy, precision, and recall.

3 Experimental evaluation

The main focus of this research is performing the hyperparameter optimization of cHM for PNN training for various classification tasks. The cHM with different sets of algorithm parameters is used to train PNN and evaluate its performance for classification tasks. The $maxFE_p$ and $maxFE_f$ values for HPO are established proportionally to population size and training set cardinality [7].

3.1 HPO

Table 1 demonstrates $maxFE_p$ and $maxFE_f$ parameter ranges used in the cHM for HPO for each dataset. The value of cHM n was taken from the set of [3,4,5] for each dataset. This gives a sample of 48 parameter sets for each dataset, tested for PNN training using the cHM algorithm (4 probing values * 4 fit values * 3 n values).

In Table 1, the results of the HPO experiment are summarized. For each dataset, the cHM parameters with the highest average accuracy value from 10 runs are presented. To exemplify, the table presents the maximum test accuracy

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

metric from the algorithm run with the highest average test accuracy for given cHM hyperparameters. In addition, Table 1 shows the ratio of $maxFE$'s probing and fit constraints together with the n value for the best run. Generally, cHM usually converged after 3 or 4 iterations, except for the Banknote dataset, where cHM performed best with 5 iterations. The results also indicate that there is no obvious choice for the probing and fit maximum number of function evaluation constraints ($maxFE_p$ and $maxFE_f$). The mean value of the probing/fit (p/f) ratio was 0.291, and the median was 0.3. The ratio distribution suggests there is no clear indication of an optimal ratio value. The detailed results of the correlation between individual cHM parameters and PNN accuracy performance on the test set are presented in Section 3.2.

Table 2(a) presents accuracy, precision, and recall metrics on average for all 10 experiment repetitions for all hyperparameters. In addition, the average standard deviation of each metric value is presented. For Iris, Wine, Cancer, ILPD, Parkinson and Climate datasets the standard deviation of each metric was smaller than 10% of the average metric value over all HPO runs. It may indicate that the cHM has similar training performance with various hyperparameters for these datasets. For the Glass, Ecoli, Heart and Banknote datasets, the average standard deviation across all HPO runs was significantly higher. This may imply the impact of cHM hyperparameters and algorithm randomness on the PNN training performance. The results for these three datasets require further studies of parameter selection for these datasets.

Table 1: HPO experiment. Tested and finally selected parameter values with the corresponding optimization results. The $maxFE_p$ and $maxFE_f$ values are in thousands of feature evaluations, e.g., 48 equals 48 000 FE.

Dataset	Ranges		Selected cHM parameters			Performance			Ratio
	$maxFE_p$	$maxFE_f$	$maxFE_p$	$maxFE_f$	n	avg acc	avg prec	avg recall	pf_ratio
Iris	[24, 48, 72, 120]	[120, 240, 480, 720]	72000	240000	3	0.937	0.938	0.937	0.3
Cancer	[91, 182, 273, 455]	[455, 910, 1820, 2730]	273000	910000	4	0.959	0.957	0.954	0.3
Wine	[28.4, 56.8, 85.2, 142]	[142, 284, 568, 852]	85200	284000	3	0.892	0.897	0.893	0.3
ILPD	[92.6, 185.2, 277.8, 463]	[463, 926, 1852, 2778]	185200	1852000	3	0.668	0.604	0.611	0.1
Glass	[34.2, 68.4, 102.6, 171]	[171, 342, 684, 1026]	171000	684000	4	0.588	0.576	0.504	0.25
Parkinson	[31.2, 62.4, 93.6, 156]	[156, 312, 624, 936]	31200	312000	4	0.931	0.905	0.917	0.1
Ecoli	[53, 106, 159, 265]	[265, 530, 1060, 1590]	265000	530000	4	0.742	0.648	0.526	0.5
Heart	[47.4, 94.8, 142.2, 237]	[237, 474, 948, 1422]	237000	1422000	4	0.442	0.290	0.251	0.167
Climate	[86.4, 172.8, 259.2, 432]	[432, 864, 1728, 2592]	259200	432000	3	0.858	0.505	0.504	0.6
Banknote	[219.4, 438.8, 658.2, 1097]	[1097, 2194, 4388, 6582]	2194000	1097000	5	0.997	0.996	0.997	0.5

Table 2: (a) Test set performance metrics from HPO. (b) Correlation results between cHM parameters (n and $maxFE_p$) and average test accuracy of trained PNN. $t_c = 2.013$ was used in all experiments.

Dataset	Accuracy		Precision		Recall	
	Mean	Std	Mean	Std	Mean	Std
Iris	0.916	0.027	0.918	0.026	0.916	0.027
Cancer	0.948	0.010	0.948	0.010	0.941	0.012
Wine	0.831	0.084	0.840	0.078	0.833	0.083
ILPD	0.650	0.022	0.587	0.025	0.594	0.028
Glass	0.529	0.088	0.488	0.121	0.434	0.096
Parkinson	0.911	0.033	0.879	0.042	0.900	0.043
Ecoli	0.702	0.079	0.576	0.135	0.455	0.088
Heart	0.401	0.049	0.246	0.043	0.217	0.035
Climate	0.849	0.014	0.489	0.036	0.492	0.033
Banknote	0.984	0.028	0.984	0.028	0.985	0.028

(a)

Dataset	n				$maxFE_p$			
	$corr_p$	t_s	$corr_s$	p_s	$corr_p$	t_s	$corr_s$	p_s
Iris	-0.065	-0.444	-0.033	0.826	-0.224	-1.561	-0.175	0.233
Cancer	0.183	1.263	0.23	0.115	-0.024	-0.166	-0.032	0.830
Wine	-0.267	-1.881	-0.268	0.065	-0.042	-0.286	-0.023	0.877
ILPD	-0.136	-0.932	-0.114	0.439	-0.116	-0.789	-0.125	0.399
Glass	0.146	1.004	0.172	0.242	0.239	1.671	0.256	0.079
Parkinson	-0.328	-2.354	-0.310	0.032	-0.300	-2.134	-0.333	0.021
Ecoli	-0.155	-1.063	-0.168	0.255	0.124	0.844	0.136	0.357
Heart	0.019	0.126	0.066	0.654	0.132	0.903	0.044	0.764
Climate	0.202	1.399	0.182	0.216	-0.082	-0.555	-0.093	0.529
Banknote	-0.006	-0.042	-0.060	0.686	0.364	2.653	0.315	0.029

(b)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

3.2 cHM parameters impact on PNN training performance

To determine the dependence of cHM hyperparameters on the performance of PNN training, Pearson and Spearman correlation coefficients were calculated to evaluate the linear and monotonic relationship between variables. For this purpose, $maxFE_p$, $maxFE_f$ and n were assumed to be independent of each other. The correlation coefficients were calculated separately for each dataset, with a degree of freedom equal to 46, which comes from $n_{samples} - 2$. Pearson's α and Spearman's $p-value$ were set to 0.05. The t_s and t_c stand for T-statistics statistical and critical values for the Pearson correlation, respectively. The p_s represents a p value for the Spearman correlation. The null hypothesis stated that there is a relationship between each parameter and the average test accuracy metric value for PNN trained with the cHM algorithm. If the absolute value of t_s was higher than t_c, the null hypothesis cannot be rejected, and there might be a linear relationship between variables according to the Pearson correlation theory. If p_s is lower than $p-value$ for the Spearman correlation, the null hypothesis may not be rejected, and the examined variables might have a monotonic relationship.

Table 2(b) demonstrates the correlation coefficients of Pearson $corr_p$ and Spearman $corr_s$ between the cHM parameter n and the average test accuracy of the PNN trained with the cHM method. The results indicate that there might be a relationship between the cHMs n value and the average test accuracy of PNN classification for the Parkinson dataset. For other datasets, calculations did not show significant evidence to assume such a relationship. Similarly, Table 2 b) presents that a relationship between the average test accuracy metric of PNN and the cHMs $maxFE_p$ for Parkinson and Banknote datasets cannot be rejected.

4 Conclusions

In this paper, Hyperparameter Optimization of the cHM algorithm for PNN training is conducted on 10 classification datasets. Although no clear optimal values for the probing and fit constraints are identified, the best performance is typically achieved within 3–4 iterations (n). The parameters $maxFE_p$ and $maxFE_f$ should be selected based on dataset characteristics, using Table 1 or via dedicated HPO. The results in this table indicate low metric variability, confirming the robustness of the cHM algorithm across different parameter settings. Correlation analysis (Table 2 b)) revealed potential relationships between cHM parameters and performance only for the Parkinson and Banknote datasets. Future work may include evaluating the impact of varying the number of inner optimizers and improving population transfer between cHM phases.

Acknowledgments. SK and PK were partially supported by the program „Excellence Initiative – research university" for the AGH University of Krakow and by a Grant for Statutory Activity from the Faculty of Physics and Applied Computer Science of the AGH. JM was partially supported by the National Science Centre, Poland, grant number 2023/49/B/ST6/01404.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52

References

1. Arsenault, B.: Learning to optimize. Articulon (2018), https://articulon.bradleyarsenault.me/article/learning-to-optimize
2. Feurer, M., Hutte, F.: Chapter 1. Hyperparameter Optimization. Springer (2019)
3. Huang, C., Li, Y., Yao, X.: A survey of automatic parameter tuning methods for metaheuristics. IEEE TEVC **24**(2), 201–216 (2020)
4. Innocente, M., Sienz, J.: Constraint-handling techniques for particle swarm optimization algorithms (01 2021). https://doi.org/10.48550/arXiv.2101.10933
5. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of ICNN'95. vol. 4, pp. 1942–1948 vol.4 (1995)
6. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science (New York, N.Y.) **220**, 671–80 (06 1983)
7. Kowalski, P.A., Kucharczyk, S., Mańdziuk, J.: Constrained hybrid metaheuristic algorithm for probabilistic neural networks learning. Information Sciences **713**, 122185 (2025). https://doi.org/10.1016/j.ins.2025.122185
8. Kowalski, P.A., Kulczycki, P.: Interval probabilistic neural network. Neural Computing and Applications **28**, 817–834 (2017)
9. Kowalski, P.A., Kusy, M.: Determining significance of input neurons for probabilistic neural network by sensitivity analysis procedure. Computational Intelligence **34**(3), 895–916 (2018). https://doi.org/https://doi.org/10.1111/coin.12149
10. Kowalski, P.A., Wadas, K.: Triggering probabilistic neural networks with flower pollination algorithm. Computational Intelligence and Mathematics for Tackling Complex Problems (2019)
11. Kusy, M., Zajdel, R.: Application of reinforcement learning algorithms for the adaptive computation of the smoothing parameter for probabilistic neural network. IEEE Trans. Neural Netw. Learn. Syst. **9**(26), 2163–2175 (2015)
12. Naik S.M., Jagannath R.P.K., K.V.: Bat algorithm-based weighted laplacian probabilistic neural network. Neural Comput & Applic **32**, 1157–1171 (2020)
13. Okulewicz, M., Mańdziuk, J.: The impact of particular components of the PSO-based algorithm solving the dynamic vehicle routing problem. Applied Soft Computing **58**, 586–604 (2017). https://doi.org/10.1016/j.asoc.2017.04.070
14. Okulewicz, M., Zaborski, M., Mańdziuk, J.: Generalized self-adapting particle swarm optimization algorithm with archive of samples (2020), https://arxiv.org/abs/2002.12485
15. Okulewicz, M., Zaborski, M., Mańdziuk, J.: Self-Adapting Particle Swarm Optimization for continuous black box optimization. Applied Soft Computing **131**, 109722 (2022). https://doi.org/https://doi.org/10.1016/j.asoc.2022.109722
16. Passino, K.: Biomimicry of bacterial foraging for distributed optimization and control. ieee control systems magazine 22(3), 52-67. Control Systems, IEEE **22**, 52 – 67 (07 2002). https://doi.org/10.1109/MCS.2002.1004010
17. Sieradzki, S., Mańdziuk, J.: Modified adaptive tree-structured parzen estimator for hyperparameter optimization (2025), https://arxiv.org/abs/2502.00871
18. Specht, D.F.: Probabilistic neural networks. Neural Networks **3**(1), 109–118 (1990)
19. Wand, M.P., Jones, M.C., et al.: Multivariate plug-in bandwidth selection. Computational Statistics **9**(2), 97–116 (1994)
20. Wu, G., Mallipeddi, R., Suganthan, P.: Problem definitions and evaluation criteria for the cec 2017 competition and special session on constrained single objective real-parameter optimization (10 2016)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_52