Technical Report

The Theory of Probabilistic Hierarchical Supervised Ensemble Learning

Ziauddin Ursani, Dmytro Antypov, Katie Atkinson, Judith Clymo, Matthew Dyer, Matthew Rosseinsky, Sven Schewe, Andrij Vasylenko
University of Liverpool

Abstract: This technical report serves as a supplement to provide data support to our research article [1]. Therefore, this should be read in conjunction with that article and as such, this report is not self-contained. This report is about the work developed for the material science project[1] aimed at design and discovery of materials for the development of human society and its environment. The work is specifically related to development of Explainable Artificial Intelligence (XAI) tools for the purpose. The Theory of Probabilistic Hierarchical Supervised Ensemble Learning (TPHSEL) is our first step in this direction. Having said that, we have also tested its generalisability outside the material science datasets.

The TPHSEL is built upon the theory of probabilistic hierarchical supervised learning (TPHSL), which says that any complex dataset can be learnt through hierarchy of simple models [1-8]. This is achieved by a constrained probability equation. In this equation a higher probability class is constrained by a certain margin called set partition, which it must overcome to win the instance in its fold, as shown in expression 1.

$$P_{max} > P_{max2} + \delta \quad (1)$$

Where:

P_{max} = largest value within the probability distribution of membership of the instance across the available classes of the dataset.
P_{max2} = 2nd largest value within the probability distribution of membership of the instance across the available classes of the dataset.
δ = Set partition – A trained parameter.

The expression 1 says that the largest probability must be larger by a certain margin δ than the 2nd runner up to win the medal of the instance. If this constraint is not overcome, then the instance will remain unclassified in the current hierarchy. Its classification will be considered again in the next hierarchy under some other model, but it must pass the test of expression 1 to get classified. This hierarchical process continues until realisation of classification of the instance. Figure-1 is a pictorial representation of expression 1.

Figure-1 shows two classes with their centroids at points X1 and X2, whereas centres of 51-dimentional balls are at Y1 and Y2. The two classes are equidistant from a

---

[1] Digital Navigation of Chemical Space (DNCS), Engineering and Physical Sciences Research Council (EPSRC), program grant (EP/V026887/1).

hyperplane Z. The Δ represents the distance between the two class boundaries, which can be mapped onto the set partition δ of expression 1. There are six instances a, b, c, d, e, and f. Class 1 contains instances 'd' and 'e', while class 2 contains instances 'a' and 'f'. However, instances b and c are not contained by any of the classes because of class separation caused by the set partition δ . Therefore, these instances will be tested on the model in the next hierarchy. They will continue to jump to the next hierarchy until classified.

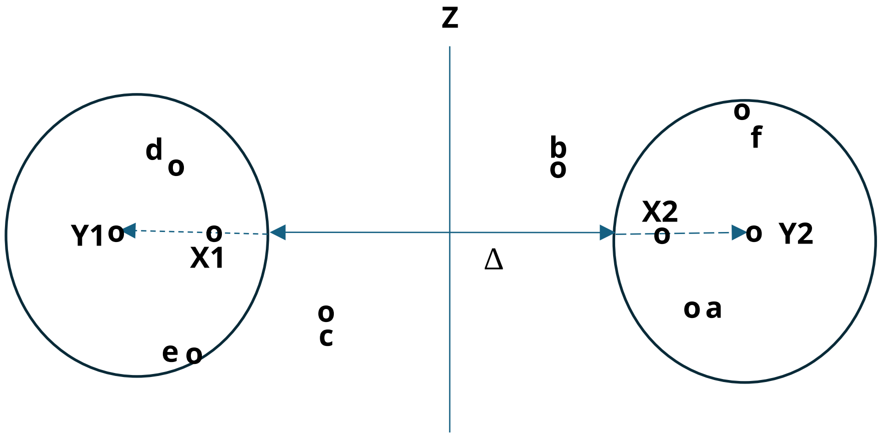

Fig. 1. Classification with TPHSL

The TPHSL was ensembled, resulting in the theory of probabilistic hierarchical supervised ensemble learning (TPHSEL). This was done to improve the quality of solutions. To test its validity, it was applied on a dataset from the materials science domain, the Lithium-Ion Conductivity Dataset [9]. The dataset contained 405 entries of chemical compositions consisting of 252 non-conductors and 153 conductors. The results are compared with the Compositionally Restricted Attention-Based network (CrabNet) [10], which is a deep learning method especially designed to predict material properties. The results are produced in Table-1.

In Table-1 column 1 lists the name of the method, number of simulations are given in column 2, whereas column 3 provides method of x-validation, accuracy results are printed in column 4, and finally Mathews corelation coefficient is figured in column 5.

TABLE I

Experimental Outcomes on Lithium-Ion Conductivity Dataset

Method	**# of Sims**	**x-validation**	**Accuracy**	**MCC**
(1)	**(2)**	**(3)**	**(4)**	**(5)**
TPHSEL	10	5-fold	0.818	0.611
CrabNet	5	5-fold	0.814	0.633

It can be seen in Table-1 that TPHSEL has performed slightly better in terms of accuracy but slightly worse w.r.t. MCC score. However, we found one interesting phenomenon when we looked at results of standard ML parameters against vote difference between the two classes. We found the quality of results proportional to vote difference. Figure-2 is testimony to this statement.

It can be seen from Figure-2 that as vote difference increases the quality of ML performance parameters also increase. Keeping this in mind and since the conductor class is the class of our interest, we also built the graph of accuracy against the vote in the favour of conductor class. The results are shown in Figure-3. In Figure-3 we see accuracy as a confidence level that we have in the material for belonging to the conductor class. Therefore, this is not a crisp decision on whether material is conductor or non-conductor but tells us rather probabilistically that the material in question is a conductor. The graph in Figure-3, shows that our confidence in the material being a conductor increases as vote in favour of conductor class increases.

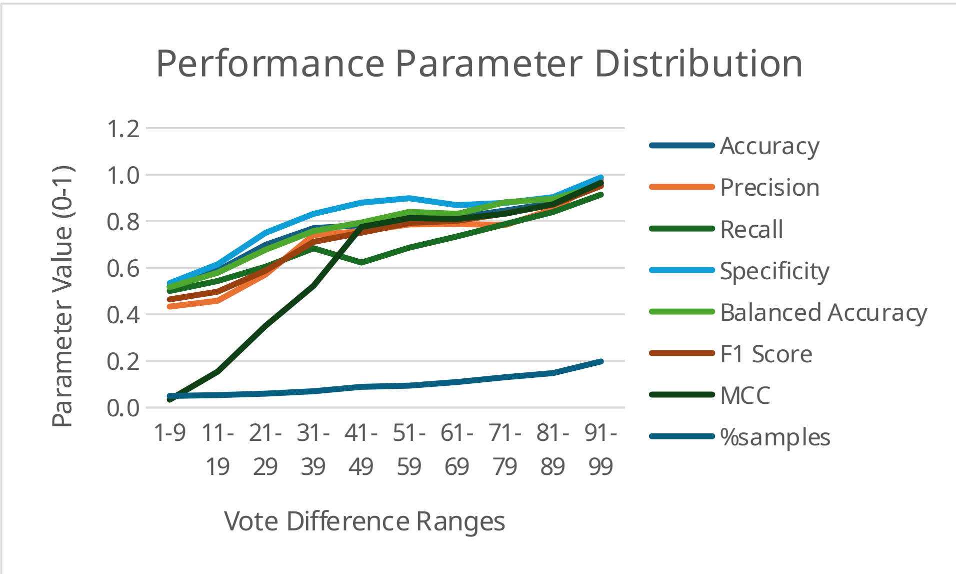

Fig. 2. Machine Learning Parameters across clusters of vote difference

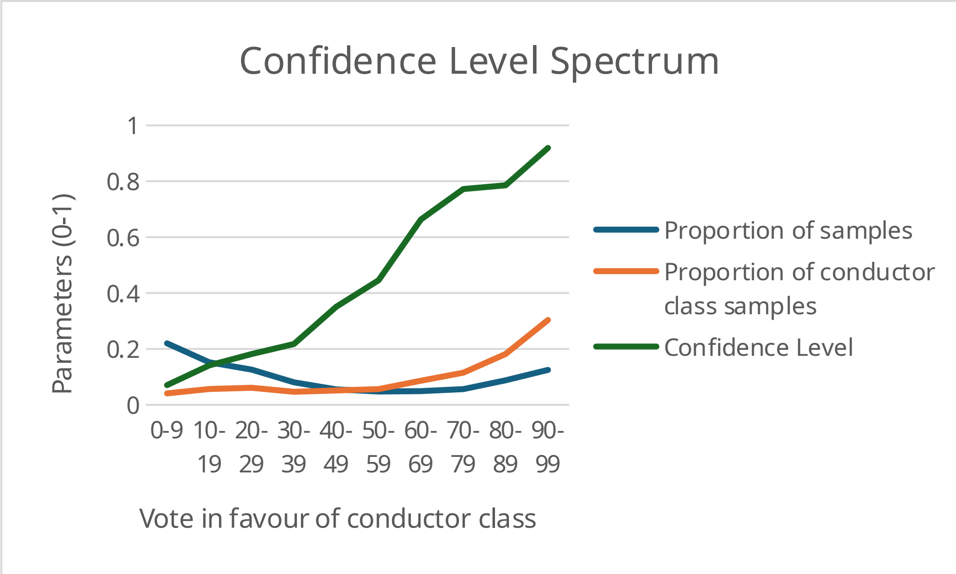

Fig. 3. Confidence level across clusters of the vote in favour of the conductor class

Encouraged by results of TPHSEL, we also applied it on the 21 general machine learning datasets from UCI machine learning repository [11]. The details of datasets are given in Table II.

In Table II, column 1 belongs to serial number, whereas dataset name and its acronym are listed in columns 2-3 respectively. Finally, dimensions of the dataset in terms of number of instances, number of features, and number of classes are mentioned in columns 4-6 respectively.

The TPHSEL was applied on the datasets of Table II. The code was run on a local HPC system with only one node and one core assigned to the program. Only 9.6 GB memory per core is allocated. The results are presented in Table III.

In Table III, column1 presents serial number, the name of the dataset is in column 2, whereas columns 3-5 provide Average accuracy, standard deviation and execution time in second respectively.

Table II
Information of General ML Datasets

Sr. #	**Name**	**Acronym**	**# of instances**	**# of features**	**# of classes**
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
1	Acute Inflammations Nephritis	AIN	120	6	2
2	Acute Inflammations Urinary	AIU	120	6	2
3	Balance Scale	BS	625	4	3
4	Banknote Authentication	BA	1372	4	2
5	Breast Cancer Wisconsin Diagnostic	BCWD	569	30	2

6	Car Evaluation	CE	1728	6	4
7	Chess King Rook vs King Pawn	CKRKP	3196	36	2
8	Climate Model Simulation Crashes	CMSC	540	18	2
9	Iris	-	150	4	3
10	Monks 1	-	432	6	2
11	Monks 2	-	432	6	2
12	Monks 3	-	432	6	2
13	Seeds	-	210	7	3
14	Seismic Bumps	SB	2584	18	2
15	Thyroid Disease ANN Thyroid	TDAT	7200	21	3
16	Thyroid Disease New Thyroid	TDNT	215	5	3
17	User Knowledge Modelling	UKM	403	5	4
18	Wall Following Robot Navigation Sensor Readings 2	WFRNSR2	5456	2	4
19	Wall Following Robot Navigation Sensor Readings 4	WFRNSR4	5456	4	4
20	Wall Following Robot Navigation Sensor Readings 24	WFRNSR24	5456	24	4
21	Wine	-	178	13	3

It can be seen in Table III that TPHSEL has achieved an average accuracy of 96.75% with average standard deviation of less than 0.5 in the average execution time of less than an hour. The results of TPHSEL were compared with random forest (RF) [12], optimal classification trees (OCT) [13], and support vector machines (SVM) [14]. The experimental setup of all the methods is given in Table IV.

In Table IV, column 1 gives the serial number; name of method is in column 2. Column 3 mentions number of simulations whereas the last column provides details about the train-test split.

It can be seen from Table IV that TPHSEL has the least training data in comparison to other methods. Finally, the comparative results are presented in Table V. Column 1, gives name of the dataset. Columns 2-5 give average accuracy of TPHSEL, OCT, RF and SVM respectively. The last column presents the best results among OCT, RF and SVM.

From the comparative results given in Table V, it can be seen that TPHSEL has got better average accuracy by the margin of 1.76% as compared to the results of the best of the three methods OCT, RF and SVM.

Table III
Experimental Results

Sr. #	Name	Avg. Acc.	Std. Dev.	Ex. Time

(1)	(2)	(3)	(4)	(5)
1	AIN	100.00	0.00	0.16
2	AIU	100.00	0.00	0.20
3	BS	99.89	0.21	21.25
4	BA	99.51	0.19	723.84
5	BCWD	96.73	0.53	898.37
6	CE	96.49	0.45	4398.97
7	CKRKP	98.53	0.20	37557.10
8	CMSC	92.15	0.91	556.93
9	Iris	95.47	0.42	3.88
10	Monks 1	100.00	0.00	111.34
11	Monks 2	91.32	1.38	200.99
12	Monks 3	99.00	0.74	114.61
13	Seeds	92.95	1.76	47.08
14	SB	91.18	0.31	48735.80
15	TDAT	98.71	0.11	159796.00
16	TDNT	96.33	0.83	12.70
17	UKM	94.99	0.64	190.55
18	WFRNSR2	99.93	0.04	12286.70
19	WFRNSR4	99.96	0.03	18919.30
20	WFRNSR24	94.42	0.45	464780.00
21	Wine	94.27	1.12	39.28
-	**Average**	**96.75**	**0.49**	**35685.48**

Table IV
Experimental Setup of Methods under Comparison

Sr. #	Method	# of Sims	Configuration
(1)	(2)	(3)	(4)
1	OCT	5	50%-25%-25% (train-valid-test)
2	RF	10	10-fold x-validation
3	SVM	100	80%-20% (train-test)
4	TPHSEL	10	3-fold x-validation

Conclusions and Future work:

This report presents a theory of probabilistic hierarchical supervised ensemble learning (TPHSEL). The theory proposes a new paradigm of learning i.e., hierarchical learning. The TPHSEL has not only produced competitive results on 21 general machine learning datasets, but it has also matched its performance with a black box method called CrabNet on the very difficult Lithium-Ion conductivity dataset from studies in materials science. The work has also produced a relationship between the quality of standard ML parameters and the pattern of voting to classes within an ensemble. This relationship would help to develop new comparison criteria among the algorithms. This in turn will also help to assign a confidence level to classification decisions and thus will be helpful in picking the instances that are most promising for the objective at hand. We believe that future extensions lie in large sized ensembles whereby statistical analysis of their results will have potential to aid achieving various objectives in the same way as we obtain different statistical objectives based on a big data.

Table V
Comparison of Results

Dataset	**TPHSEL**	**OCT**	**RF**	**SVM**	**Best of 3**
(1)	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**
AIN	100.00	100.00	100.00	-	100.00
AIU	100.00	100.00	100.00	-	100.00
BS	99.89	89.60	82.76	-	89.60
BA	99.51	98.70	99.34	-	99.34
BCWD	96.73	94.00	-	-	94.00
CE	96.49	87.50	94.70	-	94.70
CKRKP	98.53	95.60	99.46	-	99.46
CMSC	92.15	92.90	-	-	92.90
Iris	95.47	95.10	95.07	-	95.10
Monks 1	100.00	93.50	100.00	-	100.00
Monks 2	91.32	75.80	72.86	-	75.80
Monks 3	99.00	94.20	98.92	-	98.92
Seeds	92.95	91.30	93.71	-	93.71
SB	91.18	93.30	93.39	-	93.39
TDAT	98.71	95.60	-	-	95.60
TDNT	96.33	95.80	-	-	95.80
UKM	94.99	91.31	90.79	-	90.79
WFRNSR2	99.93	100.00	-	97.81	100.00
WFRNSR4	99.96	-	-	96.28	96.28
WFRNSR24	94.42	-	-	91.10	91.10
Wine	94.27	94.20	97.74	-	97.74
Average	**96.75**	**-**	**-**	**-**	**94.99**

References

1. Z. Ursani, D. Antypov, K. Atkinson, J. Clymo, M. Dyer, M. Rosseinsky, S. Schewe, A. Vasylenko. The theory of probabilistic hierarchical supervised ensemble learning. Accepted in *International Conference on Machine Learning and Applications 2024*. Florida USA.
2. Z.Ursani. The theory of probabilistic hierarchical supervised learning for classification. In 2023 *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, pages 0934–0938. IEEE,2023.

3. Z.Ursani and A.A.Ursani. The theory of probabilistic hierarchical learning for classification. *Annals of Emerging Technologies in Computing* (AETiC), 7(1): 61–74, 2023.
4. Z.Ursani and J.Dicks. Introducing the theory of probabilistic hierarchical learning for classification. In *Advances and Trends in Artificial Intelligence. From Theory to Practice: 32nd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2019*, Graz, Austria, July 9–11, 2019, Proceedings 32, pages 628–641. Springer,2019.
5. Z.Ursani and D.W.Come. A hierarchical set-partitioning nonlinear discriminant classifier trained by an evolutionary algorithm. In *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, pages15–20. IEEE,2018.
6. Z.Ursani and D.W.Corne. A hierarchical nonlinear discriminant classifier trained through an evolutionary algorithm. In *Big Data Cloud and Applications: Third International Conference*, BDCA 2018, Kenitra, Morocco, April4–5, 2018, Revised Selected Papers 3, pages 273–501 Springer,2018.
7. Z.Ursani and D.W.Corne. A novel nonlinear discriminant classifier trained by an evolutionary algorithm. In *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*, pages 336–340, 2018.
8. Z.Ursani and D.W.Corne. Use of reliability engineering concepts in machine learning for classification. In *2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 30–34. IEEE, 2017.
9. C. J. Hargreaves, M. W. Gaultois, L. M. Daniels, E. J. Watts, V. A. Kurlin, M. Moran, Y. Dang, R. Morris, A. Morscher, K. Thompson, et al. A database of experimentally measured lithium solid electrolyte conductivities evaluated with machine learning. npj *Computational Materials*, 9 (1):9, 2023.
10. A. Y.-T. Wang, S. K. Kauwe, R. J. Murdock, and T. D. Sparks. Compositionally restricted attention-based network for materials property predictions. Npj Computational Materials, 7(1):77, 2021.[10]
11. A. Asuncion and D. Newman. UCI machine learning repository, 2007.
12. J. Abellan, C. J. Mantas, J. G. Castellano, and S. Moral-Garcia. Increasing diversity in random forest learning algorithm via imprecise probabilities. Expert Systems with Applications, 97:228–243, 2018.
13. D. Bertsimas and J. Dunn. Optimal classification trees. Machine Learning, 106:1039–1082, 2017.
14. R. B. Drumond, R. F. Albuquerque, G. A. Barreto, and A. H. Souza. Pattern classification based on regional models. Applied Soft Computing, 129:109592, 2022.