Minimally Supervised Regression using Topological Projections in Self-Organizing Maps

Zimeng Lyu*, Alexander Ororbia*, Rui Li* and Travis Desell*
*Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, USA
zimenglyu@mail.rit.edu, agovcs@rit.edu, rxlics@rit.edu, tjdvse@rit.edu

***Abstract*—Parameter prediction is essential for many applications, facilitating insightful interpretation and decision-making. However, in many real life domains, such as power systems, medicine, and engineering, it can be very expensive to acquire ground truth labels for certain datasets as they may require extensive and expensive laboratory testing. In this work, we introduce a semi-supervised learning approach based on topological projections in self-organizing maps (SOMs), which significantly reduces the required number of labeled data points to perform parameter prediction, effectively exploiting information contained in large unlabeled datasets. While few-shot learning has seen significant advances in recent years, the majority of existing approaches focus on classification tasks, making our regression-based method particularly novel for continuous parameter estimation problems. Our proposed method first trains SOMs on unlabeled data followed by only using a minimal number of available labeled data points to assign targets to key best matching units (BMU). The values estimated for newly-encountered data points are computed utilizing the average of the N closest labeled data points in the SOM's U-matrix in tandem with a topological shortest path distance calculation scheme. The effectiveness of our approach has been validated through practical application in power engineering, specifically for estimating critical coal property values in coal power plants, where traditional laboratory testing is both time-consuming and costly. Our results indicate that the proposed minimally supervised model significantly outperforms traditional regression techniques, including linear and polynomial regression, Gaussian process regression, K-nearest neighbors, as well as deep neural network models and related clustering schemes.**

***Index Terms*—component, formatting, style, styling, insert.**

I. Introduction

Minimally supervised learning has become an important framework in data science, allowing practitioners to generate meaningful insights without relying on extensively labeled datasets [1]. In healthcare, for instance, these methods can help identify patient risk factors or expedite disease diagnosis even with limited expert annotations [2]. For classification tasks, the lack of sufficient labels can be approximated with weakly supervised learning or crowdsourced labels. By contrast, regression tasks often demand precise, continuous values tied to specialized measurements such as sensor readings or expert evaluations. Consequently, it becomes more difficult to rely on heuristic labels or approximate annotations. Unlike classification, where missing labels can sometimes be inferred or estimated, regression values generally cannot be reverse-engineered from features alone.

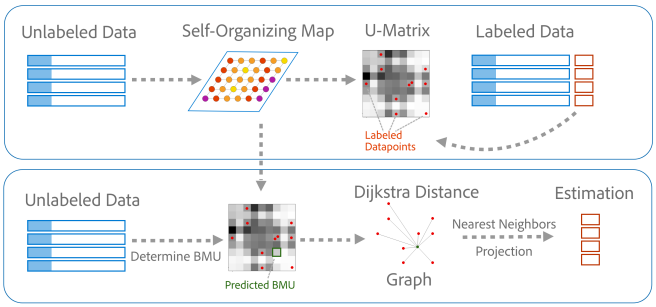


Fig. 1: The proposed minimally supervised SOM modeling framework.

A specific application of parameter estimation is often seen in power industries [3], [4]. These facilities utilize parameter estimation to estimate future demand, forecast fuel properties, anticipate maintenance needs, and assess potential environmental impacts. By predicting these values, plants can optimize operational efficiency, plan budgets, and better comply with environmental regulations.

Serving as the motivating example and data source for this study, a coal power plant needs to analyze coal property decompositions from a sensor which provides 512 spectra readings per minute as coal flows through it in order to make real-time operational decisions. This power plant has 12 cyclones, of which only two were installed with expensive sensors that record time series of these 512 channels of spectra readings. From the time that the sensors were installed in 2020, only three field tests were conducted to collect samples that were sent to a lab for analysis. Concretely, these required human specialists to travel to the power plant in order to find and collect representative coal samples and then send those samples to a lab for analysis; the lab analyzed the coal samples and sent the results back after a few weeks. Each field test was only able to acquire around 20 coal property analysis data points, in each field trip for each cyclone, at significant cost. This work examines how such a limited number of ground truth samples can be used to effectively perform regression tasks on tens of thousands of unlabeled data points.

Self-organizing maps (SOMs) [5] are a type of artificial neural network that learn in an unsupervised fashion and create

a low-dimensional space representing the training data [6], [7]. SOMs are well-known for their unique ability to maintain the topological and metric relationships of the original high-dimensional data, making them a powerful tool for visualizing and interpreting complex datasets, particularly those with many attributes or observed variables [8]–[10].

The main hypothesis of this work is that an SOM can be trained using a large amount of unlabeled data –learning how to properly represent the topology of the input data space – and be readily used for cases where labeled data is limited. This has resulted in our design of a novel approach for semi-supervised learning – one that we also call "minimally-supervised" – that utilizes topological projections within the SOM framework. For the coal data example, only 67 labeled data points were available for a dataset of over 20K samples. In this algorithm, the SOM is first trained on unlabeled data for clustering and then the few available labeled data points are mapped to the trained SOM's topology. Finally, we make use of the distance relationship between an unknown data point's best matching unit (BMU) and its closest N labeled data points' BMUs to dynamically craft parameter value predictions. The performance of the proposed method is empirically tested on coal fired power plant spectra data as well as appliance energy consumption data; our results show that our minimally-supervised learning approach works better than other powerful supervised and unsupervised methods.

II. Related Work

Semi-supervised learning, or minimally-supervised learning, primarily deals with the problem of learning from a limited number of labeled samples. It is widely used for classification-related tasks, such as natural language processing (NLP) [11], image and video analysis [12], and anomaly detection [13]. For regression problems, semi-supervised learning is mostly used for computer vision-related tasks. Dai *et al.* proposed uncertainty-consistent variational model ensembling using Bayesian networks to improve uncertainty value estimates [14] where as Rezagholiradeh *et al.* used semi-supervised generative adversarial networks for regression [15]. Semi-supervised learning for regression, however, is significantly less common. A.H. de Souza Júnior *et al.* proposed a "minimal learning machine" for supervised, distance-based (nonlinear) regression and classification on multidimensional response spaces [16]. On the other hand, Levatić *et al.* designed and employed a form of semi-supervised multi-target regression (MTR) [17]. Kostopoulos *et al.* utilized semi-supervised learning to predict the final grade of students who take online courses [18] while Meattini *et al.* proposed a minimally-supervised regression model for robot hand grasping control [19].

Similarly, many few shot learning algorithms focus on learning an embedded feature space for faster adaptation and better generalization. Vinyals *et al.* proposed matching networks, which used an attention mechanism to calculate the weights of similarity between the embeddings of a query set and a support set [20]. Snell *et al.* proposed prototypical networks to compute the mean of the examples in the embedding space and to determine the class of input data through distance metrics [21]. Sung *et al.* utilized relation networks to obtain a relation score of new samples in embedded space for classification [22]. Ravi *et al.* proposed a long short-term memory (LSTM) based model to learn an optimization algorithm for meta learning in a few-shot setting [23]. More recently, Chen *et al.* explored meta-learning on a pre-trained network using its evaluation distance metric [24] whereas Chi *et al.* proposed MetaFSCIL to incrementally learn more classes from base class [25].

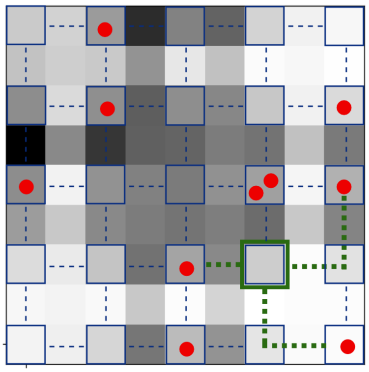


Fig. 2: An example topological projection of an unlabeled data point to a trained SOM topology with mapped labeled data.

With respect to the SOM (Kohonen map) the original efforts behind its proposal and construction focused on unsupervised clustering [5], [6], [26], [27]. However, later work showed that the SOM could be adapted for semi-supervised learning, combining it with a K-nearest neighbors (KNN) classifier to tackle different categorization tasks [28]–[30]. Nevertheless, related work using SOMs for regression is limited. For regression problems, Riese *et al.* proposed a supervised and semi-supervised SOM algorithm that was capable of unsupervised, supervised, and semi-supervised classification as well as regression on high-dimensional data [31]. Notably, the SOM has also be seen integration/fusion with other regression methods, such as support vector regression [32] [33] and geographically-weighted regression [34] for domain-specific applications. Gholami *et al.* employed an SOM for regression to simulate groundwater quality, comparing its performance with artificial neural networks (ANN) and co-active neuro-fuzzy inference systems (CANFIS) [35]. Nimmanterdwong *et al.* proposed the use of an SOM to cluster biomass datasets, integrating it with a regression model to accurately predict lignocellulosic biomass structural components from ultimate/proximate analysis [36]. Despite these few relevant efforts, to our knowledge, this is the first work to utilize the SOM topology itself to directly perform parameter prediction, particularly in the context of minimally available annotated target samples.

III. SOM Topological Projections

An SOM is a neural system that learns and adapts in an unsupervised fashion; it is a model that is used primarily for visualization and dimensionality reduction [37]. SOMs are able to transform high-dimensional data samples into a

Algorithm 1 SOM Topological Projection

function SOM(X-*unlabeled*, X-*labeled*, Y-*labeled*)
  $som = SOMClustering.fit(X\text{-}unlabeled)$ {*Cluster unlabeled data with SOM*}
  $U\text{-}Matrix = som.getU\text{-}Matrix()$
  $DistanceGraph = DijkstraGraph(UMatrix)$ {*Build Dijkstra Shortest Paths Distance Graph*}
  $PairwiseDistance = DistanceGraph.getPairwiseDistance()$ {*Shortest distance between any two nodes*}
  $Labeled\text{-}BMUs = som.transform(X\text{-}labeled)$ {*Fit labeled data on trained SOM*}
  $ClosestNeighbors = findClosestNeighbor(PairwiseDistance, Labeled\text{-}BMUs, N)$
{*Find N closest neighbors and their pairwise distances for each node*}
  $EstimatedValues = makeTable(ClosestNeighbors, Y\text{-}labeled, N)$ {*Calculate estimated values for each node*}
  for x in X-*unlabeled* **do**
    $x\text{-}BMU = som.transform(x)$ {Get unknown data point's BMU}
    $y\text{-}estimated = EstimatedValues[x\text{-}BMU]$ {Look up its estimated values}
  end for
end function
function MAKEWEIGHTEDNEIGHBORTABLE($ClosestNeighbors$, Y-*labeled*, N)
  for x in X-*unlabeled* **do**
    for n in N **do**
      $weight = 1/ClosestNeighbors[n]$ {*Weight is the inverse of pairwise distance*}
      $distance = Y\text{-}labeled[n]$
      $SumWeight+ = weight$
      $SumDistance+ = distance$
    end for
    $Estimation[x] = SumDistance/SumWeight$ {Get weighted average of closest neighbors}
  end for
  $returnEstimation$
end function
function MAKELINESEARCHTABLE($ClosestNeighbors$, Y-*labeled*, $LineSearchMethod$)
  for x in X-*unlabeled* **do**
    if $LineSearchMethod == Linear$ **then**
      $model = LinearRegressor$
    end if
    if $LineSearchMethod == Polynomial$ **then**
      $model = PolynomialRegressor$
    end if
    $model.fit$(ClosestNeighbors,Y-labeled)
    $Estimation[x] = model.predict[0]$ {The estimated value is when x=0 in regression model}
  end for
  $returnEstimation$
end function=0

low-dimensional map (usually two-dimensional) which can be represented visually through what is known as a U-Matrix – a learned map that preserves the topological and metric relationships of the original data points. This makes SOMs particularly useful for visualizing complex datasets and recognizing clusters or patterns within them.

In Kohonen maps, units are typically arranged using a square (each unit has four neighbors) or a hexagonal (each unit has six neighbors) map. As the SOM is trained, samples are matched to their BMU and the BMU, and its neighboring units within a certain radius, are moved towards the sample; the neighbors are pulled to a lesser degree depending on how far away from the BMU they are within the map. This results in neighboring units typically being closer to each other in the map while still allowing the overall map to represent the topology of the sample space.

This work utilizes the map topology to project values for unlabeled samples given the integration of a small number of labeled data points assigned to the map. Figure 1 presents a flowchart of our proposed method and Algorithm 1 formally depicts the methodology. Concretely, the SOM is first trained using unlabeled data and the labeled data points are mapped to their best matching units (BMUs) within the SOM (marked as red dots in the U-Matrix depicted in Figure 2). Parameters for unlabeled data points during inference can then be estimated by projecting values from their nearest neighbors in the SOM (see Figure 2). Squares with blue borders are the SOM units and squares between SOM nodes represent the distance between units, visualized in varying shades of gray. Darker colors represent a greater measured distance between the nodes on either side of the cell. The red dots represent the small set of labeled datapoints that are fitted into the trained SOM.

Following this, also shown in Figure 2, an example of a new unlabeled data point A is matched to its best matching unit (the green cell). Dijkstra's algorithm is then used to calculate (given the distance between neighboring units in the SOM graph) the shortest paths' pairwise distance between the green node and all of the nodes that contain labeled data (note that this step can be calculated once for each unit in the SOM and have the distance values cached to significantly improve inference performance). The N nearest neighbors – in this case, $N = 3$ – are selected (paths shown by green dotted lines) and are then used for final parameter prediction.

This method is flexible given that any distance metric could be employed in order to calculate distance values in the topological projection; in this work, we use the Euclidean

distance as it provided best results. Furthermore, the topological projection utilized to predict parameter values can also be carried out in different ways. This work investigates using a weighted average of the nearest neighbors as well as a regression based on nearest neighbor distance measurements:

a) Weighted Average Projection: Given the N nearest labeled neighbors, with parameters n_p, a weighted average estimation for each parameter e_p can be predicted given the distance from each unlabeled data point's BMU to each of that BMU's neighbors, $d(BMU, n)$:

$$e_p = \frac{\sum_{n=1}^{N} n_p \cdot \frac{1}{d(BMU,n)}}{\sum_{n=1}^{N} \frac{1}{d(BMU,n)}}.$$

b) Regression Projection: Alternately, each neighbor distance and labeled parameter value can be used as x/y pairs to train regression estimators. If the x value is the distance between the unknown node and its neighbors, then $f(x)$ is the parameter value, which is known based on the neighbors. This facilitates the use of regression in the context of conducting parameter prediction, where the estimated value is $f(x = 0)$. This work investigates estimating parameters for both linear and polynomial regressors.

IV. Experimental Setup

Datasets. This work utilizes two real-world datasets from the energy domain. The first one was the primary motivation for this work and was collected from a coal fired power plant's cyclones. The unlabeled input data is per-minute coal sample spectra readings, each with 512 channels, which were collected from the years 2021 through 2023. This unlabeled data consists of more than 20K data points. Of these, only 67 data points were taken as samples and sent to a lab for coal property analysis in order to generate labeled values for training a predictor. The 13 labeled coal properties for prediction provided by the laboratory analysis were: *BaseAcidRatio, AshContent, BTU, H2OContent, NaContent, FeContent, AlContent, CaContent, Kcontent, MgContent, SiContent, SOContent, TiContent*.

The second dataset is appliance energy usage in a low energy house [38]. The appliance energy usage can be affected by things such as the temperature and humidity of the environment in the low energy house. The energy dataset was recorded every 10 minutes and also has approximately 20K data points. The dataset contains 27 input parameters, including energy use of light fixtures, the humidity and temperature of different rooms in the house, as well as weather data from outside of the house. This dataset serves as a benchmark given that all data points contain our prediction parameter target (appliance energy usage), which lets us remove this parameter target from varying amounts of data points (simulating unlabeled data) further permitting us to examine how well our algorithm performs with more or less labeled data (given its semi-supervised nature), as compared to other benchmark methods.

Experimental Setting and Preparation. This work explores using both regression (supervised) methods as well as unsupervised clustering methods. For the supervised regression tasks, the labeled data (67 datapoints for the coal data and varying quantities for the appliance data) were divided into training and test datasets. The training consisted of 80% of the entire labeled dataset (53 samples for the coal data), and each test dataset had the remaining 20% of the labeled dataset (14 datapoints for the coal data). The supervised methods that we compared against included: linear regression, polynomial regression, a deep neural network (DNN) regressor, Gaussian process regression (GPR), and K-Nearest neighbors (KNN). With respect to unsupervised learning, we compared the clustering performance between density-based spatial clustering of applications with noise (DBSCAN) and the SOM. Unfortunately, most of the semi-supervised learning models are designed for computer vision or NLP applications, and the code for other semi-supervised learning efforts has not been provided [16], [39], thus limiting the comparisons that we were able to make.

Each experiment was repeated 10 times (each experimental trial was seeded uniquely) on a MacBook Pro with Intel Core i9 processors. Code is available at https://github.com/zimenglyu/Minimally-Supervised-SOM. We used k-fold ($k = 5$) cross validation for hyperparameter tuning and optimal model configuration. The best model chosen was then tested on the testing dataset. The input dataset is high-dimensional for the coal data (512 parameters); therefore, all of those regression tasks used the top principal components (PCs) which represented 80% of the variance from principal components analysis (PCA) used to conduct dimensionality reduction. For the coal data, the top 31 PCs represented 80% of the data's total variance.

As data normalization methods can affect model training results, given that they might alter the original data distribution, all experiments were performed using min-max scaling (or feature scaling), standardization (z-score normalization), and robust normalization. Min-max scaling does not change the data distribution, however, it can be heavily affected by large outliers. Standard normalization assumes that the feature distribution is approximately Gaussian and is less sensitive to outliers as compared to min-max scaling. Robust normalization utilizes the median and interquartile range to transform the data and is the most robust to data outliers out of the three normalization schemes.

V. Results

SOM Hyperparameters. Apart from the SOM training hyperparameters, e.g., learning rate, radius, training epochs, etc., which were tuned in the experiments, we found that two major hyperparameters significantly affected the performance of our proposed method. The first parameter that our scheme is sensitive to is the SOM's grid size, as we found that this significantly affects the clustering results as well as how spread out the labeled data points are. Figure 3 presents how the same labeled data points are distributed within SOMs with size 10×10 and 30×30 (from the perspective of the coal data set). The two SOMs are trained with the same training hyper parameters and, in the figure, the red dots mark labeled

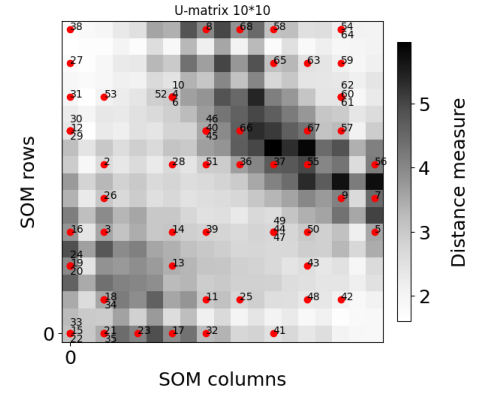
TABLE I: SOM validation RMSE under various data normalizers, topologies, and number of nearest neighbors N used for estimation.

	N=3	N=5	N=7	N=10	N=12	N=15
Minmax Normalization						
10x10	7.64e2	7.28e2	7.09e2	7.29e2	7.38e2	7.48e2
15x15	8.71e2	8.20e2	8.33e2	8.33e2	8.31e2	8.06e2
20x20	8.37e2	7.87e2	7.31e2	7.29e2	7.28e2	7.31e2
25x25	6.50e2	6.58e2	**6.44e2**	6.69e2	6.79e2	6.65e2
30x30	8.79e2	7.56e2	7.97e2	8.61e2	8.61e2	8.58e2
Standard Normalization						
10x10	8.14e2	7.61e2	7.77e2	7.74e2	7.69e2	7.81e2
15x15	7.71e2	7.43e2	7.59e2	7.70e2	7.78e2	8.15e2
20x20	8.07e2	7.85e2	7.50e2	7.57e2	7.72e2	7.64e2
25x25	7.05e2	7.25e2	7.23e2	7.11e2	7.43e2	7.27e2
30x30	6.68e2	7.87e2	7.43e2	**6.66e2**	6.76e2	6.96e2
Robust Normalization						
10x10	7.76e2	7.67e2	7.93e2	7.88e2	7.77e2	7.86e2
15x15	8.06e2	7.92e2	8.04e2	8.00e2	7.98e2	7.96e2
20x20	7.08e2	**6.83e2**	6.91e2	7.17e2	7.27e2	7.36e2
25x25	6.90e2	7.04e2	7.11e2	7.42e2	7.30e2	7.44e2
30x30	9.74e2	8.77e2	8.15e2	8.76e2	8.59e2	8.57e2

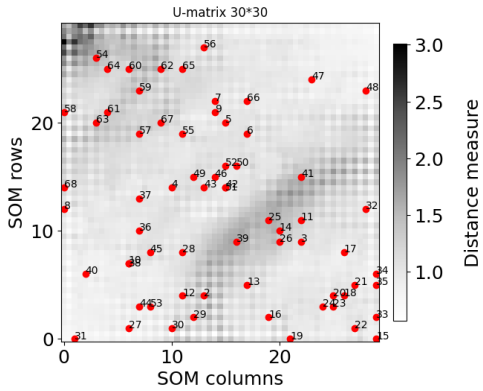
data points. The numbers next to the red dots are the identifier (ID) of each labeled data point. In the 10×10 SOM U-Matrix, many labeled data points are clustered together, such as data point $ID = 4$, 6, 10 and 52, where data points are more spread out in the 30×30 SOM U-Matrix. The more spread out the labeled data points are, the better the parameter value estimation result tended to be, since the SOM more accurately reflects the sample space topology. The second parameter our method is sensitive to is the number of neighbors used for weighted average parameter value estimation. Table I further depicts that increasing the number of neighbors used in the topological projection does not necessarily yield better value estimation results. Our results indicate that it is more ideal to use the labeled data points near the unknown point for the weighted average estimation.

Table I shows the average prediction root mean squared error (RMSE) using different SOM sizes and number of nearest neighbors over 10 repeated runs using different data normalization methods for the coal dataset. The best RMSE for each of the different data normalization methods are marked in bold font. Different data normalization schemes affect the final optimal SOM model due to their impact on the outliers as well as distribution of the original data [40]. Nevertheless, the best RMSE values for the three normalization methods are fairly close; however, using min-max scaling ultimately gives the best result.

Coal Dataset Results. Weighted averaging (WAVG) as well as linear and polynomial regression were tested using our scheme's topological projection on the coal dataset. The experimental SOM sizes and number of nearest neighbors tested were the same as those in Table I. Table II shows the best RMSE for all experiments using weighted neighbors and both regression methods. The weighted neighbor approach performs the best as compared to linear and polynomial regression under all three data normalization methods. As a result, we used this scheme for the remaining experiments.



(a) U-Matrix size 10x10



(b) U-Matrix size 30x30

Fig. 3: Labeled data point mappings in SOMs trained on the unlabeled coal data with different grid sizes.

TABLE II: SOM weighted average versus line search.

	Minmax	Standard	Robust
WAVG	6.44e2	6.66e2	6.83e2
Linear	7.75e2	7.62e2	7.90e2
Polynomial	1.77e3	1.48e3	1.65e3

TABLE III: Random guessing baseline RMSE.

	$X \sim U$	$X \sim N$		$X \sim U$	$X \sim N$
B/A	0.05	0.03	Ca	7.58e5	7.91e5
Ash	1.17	0.68	K	1.01e5	1.25e5
BTU	9.26e4	6.22e4	Mg	1.76e5	1.02e5
H2O	7.22	3.62	Si	4.91e7	2.63e7
Na	1.27e6	1.11e6	SO	9.39e6	5.72e6
Fe	3.55e6	1.79e6	Ti	9.85e3	8.90e3
Al	3.74e6	3.11e6	Average	5.25e6	3.01e6

Topological Regression vs Regression Methods. Our proposed method uses very few labeled data points for unknown data sample estimation. Generally, other supervised methods perform quite poorly when training data examples are limited and the input data is high dimensional. In these experiments, to get a sense of how these baseline methods perform, we tested supervised regression method such as linear regression, polynomial regression, GPR, DNN, and KNN.

Tables III and IV show the validation prediction RMSE in the original scale of each coal (data) property, across all the experimental methodologies. Experiments are conducted using all three data normalization methods: Minmax, Standard, and

TABLE IV: Min-max RMSE results.

	2*Linear	2*Polynomial	GPR		2*DNN	2*KNN	SOM			
			RBF	Matern			RAND	AVG	LS	WAVG
B/A	0.09	0.02	0.13	0.03	0.05	**0.01**	2.03e2	1.56e2	0.26	0.11
Ash	0.64	0.61	1.35	0.22	0.96	**0.19**	2.43e3	2.01e3	1.32	0.57
BTU	1.16e5	4.40e4	3.42e4	2.86e4	9.86e4	1.22e4	9.52e5	8.60e5	3.79e2	**1.45e2**
H2O	7.02	3.59	9.38	2.41	4.06	**1.07**	1.04e4	9.83e3	3.38	1.31
Na	2.60e6	2.75e6	4.11e6	1.25e6	2.01e6	2.32e5	7.59e6	6.84e6	1.53e3	**5.19e2**
Fe	3.10e6	1.94e6	2.12e6	1.99e6	4.70e6	1.07e6	1.27e7	1.14e7	1.84e3	**8.27e2**
Al	3.80e6	3.33e6	9.10e6	1.05e6	2.06e6	6.08e5	1.82e7	1.63e7	2.68e3	**9.75e2**
Ca	8.50e5	7.22e5	7.54e5	3.30e5	1.04e6	2.04e5	8.79e6	8.00e6	1.23e3	**5.56e2**
K	1.58e5	1.10e5	2.88e5	5.22e4	1.05e5	4.72e4	3.80e6	3.50e6	3.46e2	**1.51e2**
Mg	1.37e5	4.26e4	2.29e5	5.08e4	1.45e5	2022e4	5.39e6	4.95e6	4.60e2	**1.88e2**
Si	2.66e7	2.90e7	6.55e7	6.35e6	4.07e7	4.10e6	1.08e8	9.88e7	8.82e3	**3.48e3**
SO	1.46e7	4.51e6	5.04e6	4.56e6	1.45e7	2.20e6	4.32e7	3.98e7	3.46e3	**1.48e3**
Ti	1.26e4	8.43e3	2.48e4	3.05e3	6.05e3	9.63e2	1.43e6	1.33e6	1.17e2	**45.16**
Average	4.00e6	3.27e6	6.71e6	1.20e6	5.03e6	6.54e5	1.62e7	1.48e7	1.61e3	**6.44e2**

TABLE V: Energy data RMSE measurements.

	50			200		
	Minmax	Standard	Robust	Minmax	Standard	Robust
Linear	1.72e4	1.72e4	1.72e4	7.50e3	7.50e3	7.50e3
Poly	7.85e3	5.49e3	1.01e4	1.03e4	9.93e3	1.10e4
DNN	5.35e3	5.58e3	5.25e3	6.52e3	6.36e3	6.44e3
GPR(RBF)	5.93e3	5.10e3	6.23e3	5.81e3	5.48e3	6.25e3
GPR(Matern)	5.06e3	4.83e3	6.09e3	5.78e3	5.91e3	5.75e3
SOM WAVG	**8.32e1**	**8.57e1**	**8.04e1**	**9.01e1**	**9.25e1**	**9.14e1**

TABLE VI: GPR and DNN testing RMSE using 1600 training samples.

	Minmax	Standard	Robust
GPR(RBF)	6.81e3	6.85e3	7.82e3
GPR(Matern)	6.68e3	6.74e3	7.54e3
DNN	8.38e3	1.20e4	8.24e3

Robust. Table IV shows the results using Minmax normalization; and it was found that using Standard and Robust normalization produce similar results as Minmax. Table III provides a baseline, with the first column $X \sim U$ showing RSME using random predictions based on the uniform distribution of each coal property's value range, and the second column $X \sim N$ shows the random predictions using the normal distribution of each coal property data value. As the original dataset contains an extremely limited number of labeled data points for validation of the regression methods, the validation RMSE of those methods can be very high. We present random guessing RMSE to show if a method is valid at making predictions (i.e., do our methods improve over randomly selecting a units values). We further compare the SOM topological projections to powerful methods including Gaussian processes (GPs) and DNNs.

a) Classical Methods: We tested two classical methods, linear regression and polynomial regression. Notice that the prediction performance of these methods is close to random guessing. Polynomial regression performs slightly better than linear regression with min–max normalization while linear regression performs better using the other two normalization schemes. Since the RMSE for each coal property is reported on its unnormalized scale, the overall average is skewed by those properties with larger value ranges, we do not over-speculate on the average RMSE provided in the bottom row.

b) Gaussian Process Regression (GPR): is a non-parametric, Bayesian approach to regression that employs a probabilistic framework for learning and inference [41], which has shown strong performance on small sized datasets. We used two kernel functions for GPR: $Constant{*}RBF$ (a radial basis function kernel) and $Constant * Matern$ (a Matern generalized RBF kernel).

c) Deep Neural Networks (DNNs): offer a powerful, nonlinear approach to tackling high-dimensional regression problems. After applying PCA to the input data features, we obtain 31 (transformed) features, upon which a simple DNN is further applied to conduct regression/estimation. The structure of the DNN utilized three dense layers of widths (31, 15, and 10) each utilizing a ReLU activation function. The DNN performs better with the min-max scaling method, but generally worse than the GPR methods. As DNNs generally require large training datasets to obtain good-quality performance, this was expected.

d) K-Nearest Neighbors (KNNs): is a simple but widely-used method for regression tasks. The naïve KNN predicts values by taking the average values of its "k" nearest neighbors. The results show that KNN outperforms other methods in predicting 2-3 coal properties for the coal dataset; however, it generally still performs worse than our proposed method.

e) The Self-Organizing Map: To show that the distance on the U-Matrix plays an important role in parameter value estimation, we also tested the use of non-weighted average values of randomly selected neighbors, non-weighted average values of closest neighbors, as well as linear regression on closest neighbors (polynomial regression is not shown due to the poor performance that we found it exhibited). The results in Table IV demonstrate that using the average value of random neighbors (*SOM RAND*) performs the worst (as expected), using average value of closest neighbors (*SOM AVG*) does better than random neighbors, and linear regression (*SOM LS*) among closest neighbors is better than average closest neighbors. Note that using the weighted average of closest neighbors (*SOM WAVG*) does the best over all proposed methods as well as other regression methods. In general, we find that *SOM WAVG* performs orders of magnitude better than the other methods for almost all coal properties.

A. Unsupervised DBSCAN

It is possible to apply the idea of using the result of clustering and the distance values among adjunct clusters for value estimation in other schemes beyond the SOM. In light

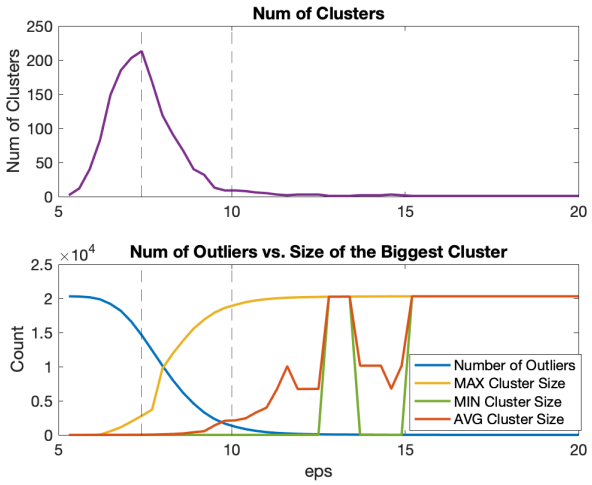


Fig. 4: DBSCAN with PCA.

of this, we explored using another unsupervised clustering method for the weighted average value estimation, i.e., DBSCAN (density-based spatial clustering of applications with noise) [42] which is an unsupervised clustering algorithm known for its ability to effectively group data points in high-dimensional spaces. DBSCAN identifies dense clusters, distinguishing them from sparser noise regions, making it highly suitable for high-dimensional datasets.

DBSCAN requires two main parameters: eps and $min\_samples$. eps is the maximum distance between two samples for one to be considered as in the neighborhood of the other. $min\_samples$ is the number of samples in a neighborhood for a point to be considered as a core point; this includes the point itself. The plot below show the number of clusters and cluster size using $min\_samples = 3$ and different eps. Figure 4 show the relationship of using different eps and the number of clusters, size of the biggest, average, and smallest cluster, and the number of outliers on the original data and data with 80% variance preserved utilizing PCA.

The vertical dash line shows the potential eps for clustering and its corresponding number of clusters and the number of outliers. When it reaches the highest number of total clusters, the amount of outliers are around 20K and all clusters are extremely small. When the number of outliers drops significantly, the biggest cluster contains almost all the datapoints, which left very little room for (effective) value estimation. The results from DBSCAN shows that not all unsupervised learning clustering methods work with the proposed semi-supervised scheme in comparison to utilizing SOMs.

B. Energy Dataset

As the coal dataset has a very limited number of labeled data points, we utilize the energy dataset to study the effectiveness of SOM topological projections using different quantities of labeled data points for training. We utilized 50, 100, and 200 randomly selected labeled data points, with regression models using 80% for training, and 20% for validation. For the SOM topological projection, we utilized the remaining data points as unlabeled data, removing the prediction parameter target. Table V presents the weighted average value projection compared to the other regression methods. From these results, we observe that the SOM does significantly better than the other regression methods.

To further explore the regression capability of GPR and DNN on even larger training dataset, GPR and DNN models were trained on datasets containing 1600 data points and tested on sets with 200 data points. The validation results are shown in Table VI. Comparing results in Table V and VI, even training with significantly more training data does not necessary improve the performance of complex regression models in our problem contexts. Note that the SOM topological projections, even when using a greatly limited set labeled data patterns (even just 50 labeled samples), still outperforms these methods by several orders of magnitude.

VI. Conclusion

This paper proposed a minimally-supervised learning algorithm based on topological projections in a self-organizing map (SOM). Our scheme clusters the unlabeled data within an SOM and then utilizes the distance between SOM units to create a graph that can be used to compute the shortest paths in the SOM topology space between nodes via Dijkstra's algorithm. Labeled data points are mapped to the SOM and predictions of unlabeled data points can be made via topological projections based on the closest labeled nearest neighbor units from the best matching unit of the unlabeled data.

We investigated utilizing both a weighted average and regression methods to perform parameter prediction given the nearest neighbors, finding that the weighted average method performed best. Our results also show that our proposed SOM method performs orders of magnitude better than other supervised learning regression methods, such as linear and polynomial regression, Gaussian process regression, and deep neural networks, as well as an unsupervised learning scheme, i.e., DBSCAN. We also show that our methods still significantly outperform supervised methods even when the supervised models are provided with much more labeled training data.

Our methodology provides a promising approach for semi-supervised regression tasks, particularly offering an effective scheme for large, mostly unlabeled, datasets, particularly in instances where very few data points are annotated. The scheme is effective with high dimensional data and further results can be easily visualized through the use of U-Matrices. Our work also provides interesting avenues for future research endeavors, such as efforts that study others methods for performing the topological projections inherent to our computational framework. Additionally, as our neural method is easily visualized via a U-Matrix, this system could be used to facilitate explainablity in model predictions.

References

[1] A. Abdulaziz, J. Zhou, M. Fang, S. McLaughlin, A. Di Fulvio, and Y. Altmann, "A variational autoencoder for minimally-supervised pulse

shape discrimination," *Annals of Nuclear Energy*, vol. 204, p. 110496, 2024.

[2] Z. Huang, G. Long, B. Wessler, and M. C. Hughes, "A new semi-supervised learning benchmark for classifying view and diagnosing aortic stenosis from echocardiograms," in *Machine Learning for Healthcare Conference*. PMLR, 2021, pp. 614–647.

[3] J. Smrekar, D. Pandit, M. Fast, M. Assadi, and S. De, "Prediction of power output of a coal-fired power plant by artificial neural network," *Neural Computing and Applications*, vol. 19, pp. 725–740, 2010.

[4] D. Adams, D.-H. Oh, D.-W. Kim, C.-H. Lee, and M. Oh, "Prediction of sox–nox emission from a coal-fired cfb power plant with machine learning: Plant data learned by deep neural network and least square support vector machine," *Journal of Cleaner Production*, vol. 270, p. 122310, 2020.

[5] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological cybernetics*, vol. 43, no. 1, pp. 59–69, 1982.

[6] H. Vaidya, T. Desell, A. Mali, and A. Ororbia, "Neuro-mimetic task-free unsupervised online learning with continual self-organizing maps," *arXiv preprint arXiv:2402.12465*, 2024.

[7] A. G. Ororbia, "Brain-inspired machine intelligence: A survey of neurobiologically-plausible credit assignment," *arXiv preprint arXiv:2312.09257*, 2023.

[8] T. Kohonen, "Essentials of the self-organizing map," *Neural networks*, vol. 37, pp. 52–65, 2013.

[9] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Transactions on neural networks*, vol. 11, no. 3, pp. 586–600, 2000.

[10] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, no. 10, pp. 1358–1384, 1996.

[11] C. Qiang, H. Li, H. Ni, H. Qu, R. Fu, T. Wang, L. Wang, and J. Dang, "Minimally-supervised speech synthesis with conditional diffusion model and language model: A comparative study of semantic coding," *arXiv preprint arXiv:2307.15484*, 2023.

[12] D. Wang, Y. Zhang, K. Zhang, and L. Wang, "Focalmix: Semi-supervised learning for 3d medical image detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3951–3960.

[13] D. Azzalini, L. Bonali, and F. Amigoni, "A minimally supervised approach based on variational autoencoders for anomaly detection in autonomous robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2985–2992, 2021.

[14] W. Dai, X. Li, and K.-T. Cheng, "Semi-supervised deep regression with uncertainty consistency and variational model ensembling via bayesian neural networks," *arXiv preprint arXiv:2302.07579*, 2023.

[15] M. Rezagholiradeh and M. A. Haidar, "Reg-gan: Semi-supervised learning based on generative adversarial networks for regression," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 2806–2810.

[16] A. H. de Souza Junior, F. Corona, G. A. Barreto, Y. Miche, and A. Lendasse, "Minimal learning machine: A novel supervised distance-based approach for regression and classification," *Neurocomputing*, vol. 164, pp. 34–44, 2015.

[17] J. Levatić, M. Ceci, D. Kocev, and S. Džeroski, "Semi-supervised learning for multi-target regression," in *New Frontiers in Mining Complex Patterns: Third International Workshop, NFMCP 2014, Held in Conjunction with ECML-PKDD 2014, Nancy, France, September 19, 2014, Revised Selected Papers 3*. Springer, 2015, pp. 3–18.

[18] G. Kostopoulos, S. Kotsiantis, N. Fazakis, G. Koutsonikos, and C. Pierrakeas, "A semi-supervised regression algorithm for grade prediction of students in distance learning courses," *International Journal on Artificial Intelligence Tools*, vol. 28, no. 04, p. 1940001, 2019.

[19] R. Meattini, A. Bernardini, G. Palli, and C. Melchiorri, "semg-based minimally supervised regression using soft-dtw neural networks for robot hand grasping control," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 144–10 151, 2022.

[20] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," *Advances in neural information processing systems*, vol. 29, 2016.

[21] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[22] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1199–1208.

[23] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International conference on learning representations*, 2016.

[24] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 9062–9071.

[25] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "Metafscil: A meta-learning approach for few-shot class incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 14 166–14 175.

[26] A. J. Amos, K. Lee, T. S. Gupta, and B. S. Malau-Aduli, "Validating the knowledge represented by a self-organizing map with an expert-derived knowledge structure," *BMC Medical Education*, vol. 24, no. 1, p. 416, 2024.

[27] M. Jalali, M. Shademani, M. Paripour, and M. Jalali, "Assessment of water quality for mountainous high-elevated spring waters using self-organized maps," *Groundwater for Sustainable Development*, vol. 24, p. 101082, 2024.

[28] I. Jahan, F. Mohamed, V. Blazek, L. Prokop, S. Misak, and V. Snasel, "Power quality parameters forecasting based on som maps with knn algorithm and decision tree," in *2023 23rd International Scientific Conference on Electric Power Engineering (EPE)*. IEEE, 2023, pp. 1–6.

[29] G. Li, F. Liu, and H. Yang, "Research on feature extraction method of ship radiated noise with k-nearest neighbor mutual information variational mode decomposition, neural network estimation time entropy and self-organizing map neural network," *Measurement*, vol. 199, p. 111446, 2022.

[30] L. Suchenwirth, W. Stümer, T. Schmidt, M. Förster, and B. Kleinschmit, "Large-scale mapping of carbon stocks in riparian forests with self-organizing maps and the k-nearest-neighbor algorithm," *Forests*, vol. 5, no. 7, pp. 1635–1652, 2014.

[31] F. M. Riese, S. Keller, and S. Hinz, "Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data," *Remote Sensing*, vol. 12, no. 1, p. 7, 2019.

[32] J. Che, J. Wang, and G. Wang, "An adaptive fuzzy combination model based on self-organizing map and support vector regression for electric load forecasting," *Energy*, vol. 37, no. 1, pp. 657–664, 2012.

[33] Z. Dong, D. Yang, T. Reindl, and W. M. Walsh, "A novel hybrid approach based on self-organizing maps, support vector regression and particle swarm optimization to forecast solar irradiance," *Energy*, vol. 82, pp. 570–577, 2015.

[34] Z. Wang, J. Xiao, L. Wang, T. Liang, Q. Guo, Y. Guan, and J. Rinklebe, "Elucidating the differentiation of soil heavy metals under different land uses with geographically weighted regression and self-organizing map," *Environmental Pollution*, vol. 260, p. 114065, 2020.

[35] V. Gholami, M. Khaleghi, S. Pirasteh, and M. J. Booij, "Comparison of self-organizing map, artificial neural network, and co-active neuro-fuzzy inference system methods in simulating groundwater quality: geospatial artificial intelligence," *Water Resources Management*, vol. 36, no. 2, pp. 451–469, 2022.

[36] P. Nimmanterdwong, B. Chalermsinsuwan, and P. Piumsomboon, "Prediction of lignocellulosic biomass structural components from ultimate/proximate analysis," *Energy*, vol. 222, p. 119945, 2021.

[37] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[38] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and buildings*, vol. 140, pp. 81–97, 2017.

[39] Y.-F. Li, H.-W. Zha, and Z.-H. Zhou, "Learning safe prediction for semi-supervised regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 2017.

[40] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, 2020.

[41] C. Williams and C. Rasmussen, "Gaussian processes for regression," *Advances in neural information processing systems*, vol. 8, 1995.

[42] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *kdd*, vol. 96, 1996, pp. 226–231.