Cluster-wise NN-ERA

Dennis Padilla

March 2023

1 Introduction

Social science research is experiencing a paradigm shift due to recent advancements in technology and computational power. These advancements have given rise to a new set of tools known as machine learning (ML) methods (Hindman, 2015; Yarkoni and Westfall, 2017). ML is an approach that creates machine intelligence through algorithms and data (Ghahramani, 2015). Social scientists are deploying ML techniques to address the challenges of identifying nonlinear relationships, producing parsimonious models, and increasing model reproducibility (Grimmer et al., 2021). Their application is helping scientists draw insights from data in a way that has been limited in the past. Currently, social scientists are deploying ML methods as standalone techniques or in tandem with other theoretical approaches (e.g., Reier Forradellas et al., 2020; Vijayakumar et al., 2022). Regardless of the method of deployment, they deliver a richer understanding of the social phenomenon of interest.

By design, ML methods are capable of making accurate predictions on data through complex pattern recognition. In social science research, social scientists would combine ML with the structural equation modelling (SEM) framework to take advantage of this capability. The function of the SEM framework is to define a theoretical relational model consisting of observable and latent variables. The researcher then assesses the model's plausibility according to observed data (Jöreskog, 1970; Wright, 1934). Combining SEM with ML, social scientists could capture both the causality relationship among variables using the theoretical model and whether the model can produce accurate predictions on out-of-sample data. To date, social scientists are deploying the SEM-ML approach as a step-wise procedure or in a unified framework (Aktepe et al., 2015; Albahri et al., 2022; Zobair et al., 2021).

In the step-wise approach, social scientists use ML as a means of data exploration or model testing. In the former, when exploring data, social scientists may employ one or a combination of several exploratory ML methods to build a model. They then utilize insights from the data-driven approach to formulate a theoretical model using the SEM framework. This iterative approach can reduce the time to develop new theories given that ML methods are not restricted to linear relationships (MacCallum and Austin, 2000). In the latter, where a

theoretical model is defined, social scientists will deploy a set of predictive ML methods that test whether the model is capable of making accurate predictions on out-of-sample data (Abbasi et al., 2021; Ebrahimi et al., 2022). In social science, the function of out-of-sample prediction is not yet commonplace, but will likely become more widespread as more social scientists begin utilizing ML methods.

In the unified SEM-ML approach, social scientists benefit from both the theoretical and empirical aspects of the two models. To date, the neural network extended redundancy analysis framework is an approach that attempts to combine the two models in one objective function (NN-ERA; Vijayakumar et al., 2022 As an exploratory approach, this introduces nonlinearity into the model while maintaining the theoretical structure imposed by the researcher. Further details on the NN-ERA framework will be discussed in the following sections.

In SEM-ML applications, social scientists should also consider the use of a set of techniques known as cluster analysis. This set of practices is becoming increasingly important as data gets collected in more variety, in increasing volumes, and at increasing velocities, a phase of data collection known as Big Data (Elgendy and Elragal, 2014). The utility of cluster analysis stems from their ability to identify heterogeneous groups within data (Brusco et al., 2017; Madhulatha, 2012; Seifert, 1997). Such an approach helps segment data into smaller subsets in Big Data samples, allowing for insights to be drawn during the process of data collection.

Social scientists already acknowledge the importance of clustered data given the popularity of methods such as multi-level modelling and latent class analysis (Enders and Tofighi, 2007; McCutcheon, 19877). However, when data is unclassified, or if it may exist in other unspecified clusters, social scientists may not consider using a clustering approach. One example of the application of cluster analysis in psychology would be analyses that consider the social behaviour of groups. These research questions would benefit from the use of cluster analysis to identify how different groups respond to social phenomena. This exploratory approach can provide insights to refine or build new theories regarding the clusters under analysis.

Therefore, the motivation of this research is to combine cluster analysis with the state-of-art SEM framework to allow for further exploration within a theoretical model. Specifically, the paper intends to offer insight into the use and application of clustering in psychological research. Through the development of a clustering extension to the NN-ERA framework, the thesis aims to identify two main insights. Firstly, the research intends to assess whether theorized relationships can be identified between clusters in a non-linear framework and secondly, this research aims to build on the literature regarding the use cases of machine learning techniques in psychological research.

2 Background Literature

The cluster-wise NN-ERA approach is a combination of several earlier models, including neural networks, cluster analysis and a special case of the SEM framework. In this section, we offer an overview of these components, as well as further texts for the interested reader.

2.1 Neural Networks

Neural networks are an ML approach that takes inspiration from the human brain (McCulloch and Pitts, 1943). They are composed of units known as nodes that are connected via weights. Networks are composed of a series of nodes, known as layers. Layers of a basic neural network model can be categorized as input, hidden, or output layers. As their names suggest, input layers and output layers are responsible for the passing of information to and from the user, whereas hidden layers are responsible for applying activation functions, or nonlinear mapping, to data. In the network architecture, each layer of nodes passes information to the next layer through their associated weights. This structure of nodes, weights, and layers form the basis of all neural network architectures.

The simplest neural network architecture, known as the shallow neural network, is a composition of an input layer, a single hidden layer, and an output layer. Architectures beyond the three-layer design are beyond the scope of this paper, but Goodfellow et al., 2016 provide additional information on other deep neural network architectures.

2.1.1 Learning Approaches

Currently, shallow neural networks are trained using supervised, unsupervised, or reinforcement learning to identify insights from data (Alloghani et al., 2020; Caruana and Niculescu-Mizil, 2006). The choice or combination of these learning approaches depends on the application of the network and the type of data available.

Supervised learning is the most common approach to train shallow neural networks as it involves labelled data to train models. Labelled data contains a true output or class that the model is meant to predict. In these approaches, models are dependent on human insights to provide correctly labelled data or to feature engineer variables for the model to learn from. In the supervised setting, data is partitioned into a training set and a test set. The model is initially trained on the training set and may undergo modifications to improve its performance. To evaluate the accuracy of the model, it is then tested on the unseen data of the test set. Commonly, models that are trained via supervised learning are applied to solve classification or prediction challenges (Nasteski, 2017; Vaswani et al., 2017).

Alternatively, unsupervised learning allows models to identify patterns in the data without the intervention of humans. Specifically, this entails training the

model on data that does not specify a true score or classification. Unsupervised learning alleviates the requirement for labelled data and the need to manually perform feature engineering. This form of learning, however, produces insights that are unknown to the researcher and because of this unsupervised learning methods may be less accurate as estimates cannot be compared to ground truth. Applications of models trained via unsupervised learning include clustering, dimensionality reduction, and visualization (Usama et al., 2019).

A re-emerging learning approach known as reinforcement learning is a gamified, trial-and-error approach to model training. It involves an agent, an environment, and a reward, where the goal of the learning approach is to maximize the reward outcome. Contrary to supervised and unsupervised learning, reinforcement learning creates data through its experience in an environment. Models trained via reinforcement learning are being applied to challenges related to computer vision and natural language processing (Le et al., 2022).

2.1.2 Algorithm

To improve the performance of the network, scientists utilize an algorithm known as feed-forward backpropagation. The intuition behind the approach is to assess how well the model performs and then update the model parameters to improve performance.

2.1.2.1 Forward Propagation

Forward propagation refers to the direction of data being passed in a network to estimate output. We let n represent the number of data points, let m represent the number of input variables, let h represent the number of hidden layer nodes, and let o represent the number of output layer nodes. To compute forward propagation from input to output, consider

$$\mathbf{Z^{(2)}} = \mathbf{X}\mathbf{W^{(1)}} \tag{1}$$

where $\mathbf{X}$ represents an n by m input matrix and $\mathbf{W^{(1)}}$ represents an m by h matrix of weights connecting the jth neuron of the previous layer to the kth neuron in the next layer. At the hidden layer, an activation function, ϕ, is applied, such that

$$a^{(2)} = \phi(\mathbf{Z^{(2)}}) \tag{2}$$

Where $a^{(2)}$ represents the output matrix of the activation function of dimensions n by h. Propagating to the output layer, the process of summation and activation function computation is repeated. Here, the output from the hidden layer is multiplied by the second layer weight matrix, $\mathbf{W^{(2)}}$, of dimensions h by o, where

$$\mathbf{Z^{(3)}} = a^{(2)}\mathbf{W^{(2)}} \tag{3}$$

And applying the activation function yields the estimate of the output variable $\hat{y}$.

$$\hat{y} = \phi(\mathbf{Z^{(3)}}) \tag{4}$$

2.1.2.2 Backpropagation

Backpropagation refers to the process of using gradient descent to update model parameters (Curry, 1944; Pedregal, 2003). The algorithm traverses backwards through a network and uses the error, e at each layer to update the weights of the previous layer. We define the error function, $E(X, \theta)$ as the error between the desired output, y, and the estimated output, $\hat{y}$, of the neural network with model parameters, θ. We also define the learning rate, α, as the hyperparameter of the model. Gradient descent updates the model parameters according to

$$\theta^{i+1} = \theta^i - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta} \tag{5}$$

where t represents the parameters of the network at iteration t in gradient descent.

2.2 Extended Redundancy Analysis

Extended redundancy analysis allows for the exploration of the directional relationship between multiple sets of variables (ERA; Takane and Hwang, 2005 The ERA is a special case of the SEM framework and is composed of observable variables and latent variables. The framework is considered a special case of the SEM framework as it assumes formative relationships between observable and latent variables. The ERA framework can be described as follows. Let $Z^{(1)}$ represent an n by r matrix of dependent variables, and let $Z^{(2)}$ represent an n by t matrix of independent variables. The model can be expressed as

$$\mathbf{Z^{(1)}} = \mathbf{Z^{(2)}WA} + \mathbf{E} \tag{6}$$

$$rank(\mathbf{WA}) \leq min(r, t) \tag{7}$$

where $\mathbf{W}$ is a t by d matrix of component weights, $\mathbf{A}$ is a d by r matrix of component loadings, and $\mathbf{E}$ is an n by r matrix of errors. To account for multiple sets of variables, equation (6) can be expressed as

$$\mathbf{Z^{(1)}} = [\mathbf{Z_2}, \mathbf{Z_3}]\mathbf{WA} + \mathbf{E} \tag{8}$$

where the weight matrix, $\mathbf{W}$, restricts component loadings to the specified component factor such that weights correspond to their respective factor and are zero elsewhere in the matrix.

2.2.1 Algorithm

To estimate the parameters of the model, the ERA uses an alternating-least squares algorithm such that the weight matrix, $\mathbf{W}$, and component loading matrix, $\mathbf{A}$, parameters are updated in an alternating fashion until convergence. For further information on the alternating least squared method for parameter estimation of the ERA, we refer the reader to Takane and Hwang, 2005 and de Leeuw et al., 1976.

2.3 Clustering

ML models that are trained via unsupervised learning can be applied to clustering problems (Madhulatha, 2012; Saxena et al., 2017). The model takes a data-driven approach to segment data into heterogeneous subgroups according to a specified metric of data similarity. The three main approaches to clustering include hierarchical clustering, density-based clustering, and centroid-based clustering.

2.3.1 Hierarchical Clustering

Hierarchical clustering is an algorithm that groups points into sets of clusters, where clusters are distinct from one another, and the points within a cluster are more similar compared to points of another cluster (Murtagh and Contreras, 2012; Saxena et al., 2017). Hierarchical clusters form distinct clusters through a top-down or a bottom-up approach. The top-down approach begins by considering one cluster containing all the points and is said to converge when all points reach individual clusters. On the other hand, bottom-up hierarchical clustering begins by considering all points as unique clusters, then reaches convergence when all points agglomerate to one cluster.

Hierarchical clustering determines the clusters according to a similarity metric and linkage criteria. It is often the case that the similarity metric and linkage criteria are determined according to the field of research, but commonly scientists perform hierarchical clustering according to a Euclidean distance similarity metric and an average linkage method criteria. The hierarchical clustering approach is well suited for hierarchical data, such as taxonomies or webpage data. For the interested reader, we suggest Yim and Ramdeen, 2015 further information on hierarchical clustering, similarity metrics, and linkage criteria.

2.3.2 Density-based Clustering

Density-based clustering segments data into clusters according to a point's surrounding density. The approach utilizes two hyperparameters to determine density, the radius of a neighbourhood, ϵ, and the minimum number of points within a neighbourhood to define a core point. According to these hyperparameters, points are categorized either as core points, border points, and outliers. Through this approach, non-ellipsoid clusters and outlier points can be identified. The most common density-based clustering method is known as

Density-Based Spatial Clustering of Applications with Noise (DBSCAN). We refer the interested reader to Ester et al., 1996 for more details on density-based approaches.

2.3.3 Centroid-based Clustering

Centroid-based clustering partitions data according to a specified number of clusters. Like hierarchical clustering, centroid-based approaches utilize a distance metric to determine the association of points to a centroid. The centroid-based approach is an iterative algorithm that determines the distance of each point to each cluster centroid, then updates the centroid according to the average distance of the points closest to the centroid. The most common centroid-based clustering technique is k-means clustering (MacQueen, 1967).

Compared to the hierarchical and density-based clustering approaches, the centroid-based approach scales better to larger datasets and outperforms both techniques with time complexity $O(ntk)$, compared to the time complexity of DBSCAN ($O(n^2)$), and hierarchical clustering ($O(n^3)$) (Jang and Jiang, 2019; Murtagh and Contreras, 2012; Whittingham and Ashenden, 2021).

Compared specifically to the hierarchical approach, centroid-based clustering is capable of handling cluster re-assignment, a limitation of hierarchical methods (Madhulatha, 2012). In comparison to density-based approaches, centroid-based approaches are easier to implement given the need for a single hyperparameter, k (Madhulatha, 2012).

Given the advantages of the centroid-based approach over other clustering techniques, we aim to integrate the common centroid-based approach, k-means clustering, into the hybrid NN-ERA framework. The proposed method will assess the feasibility of a unified approach and determine whether novel insights can be uncovered through the hybrid theoretical machine learning model.

3 Proposed Method

The cluster-wise NN-ERA approach integrates k-means clustering to identify heterogeneous groups within a theoretical framework. The proposed method is motivated by the growing application of Big Data in social science research. However, the use of larger datasets brings new challenges regarding computational complexity and violations of assumptions regarding the normality of the data. The addition of the k-means algorithm addresses both of these challenges being both computationally light and scalable, as well as being able to identify heterogeneous subgroups in data (Cheung and Jak, 2016). Moreover, the integration of the two methods encourages a theory-building process based on cluster-specific differences according to the proposed theory. Prior to describing the cluster-wise NN-ERA approach, we briefly describe the NN-ERA and the k-means clustering algorithms in more detail.

3.1 NN-ERA

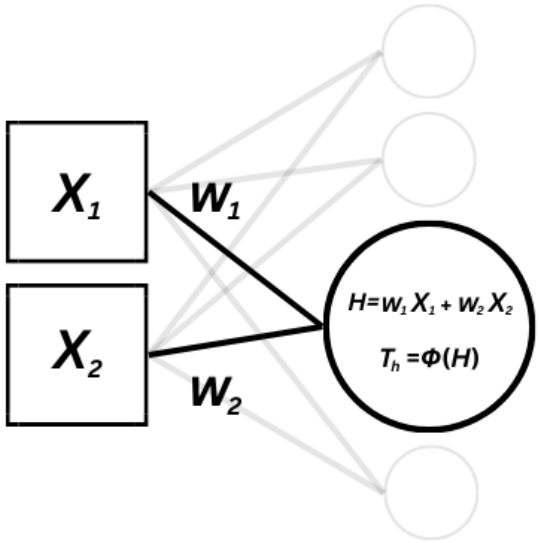

Figure 1: Computation at the hidden layer of the NN-ERA.

The NN-ERA framework is a unification of the ERA and the neural network model (Vijayakumar et al., 2022 The unified approach considers nonlinearities and interactions between the observed and latent variables. As an integration of the neural network approach, the NN-ERA utilizes the feed-forward backpropagation algorithm to estimate model parameters. This lends the model to be advantageous over similar approaches as it is capable of grouping sets of variables onto a single latent variable beyond the conventional linear approaches of previous ERA and SEM models.

The algorithm of the NN-ERA framework follows a similar approach to the feed-forward backpropagation algorithm. The feed-forward component can be described as follows:

$$\mathbf{Y} = \mathbf{F}\mathbf{W_o} \tag{9}$$

where $\mathbf{Y}$ represents an n by q matrix of predicted outputs, where $\mathbf{F}$ represents an n by p matrix of component scores and where $\mathbf{W_o}$ represents a p by q matrix of weights connecting the latent layer to the output layer. $\mathbf{F}$ can be further expanded as:

$$\mathbf{F} = \phi(\phi(\mathbf{X}\mathbf{W_h})\mathbf{W_f}) \tag{10}$$

where ϕ represents a non-linear activation function that is specified by the researcher, where $\mathbf{X}$ represents an n by m matrix of observable variables, where

$\mathbf{W_h}$ represents an m by o matrix of weights connecting inputs to the hidden layer and where $\mathbf{W_f}$ represents an o by p matrix of weights connecting the hidden layer to the latent layer. The most common nonlinear activations include ReLU, leaky ReLU, and sigmoid functions (Sharma et al., 2020). For the interested reader, we refer them to Szandała, 2021 for further information on activation functions in ML and DL models.

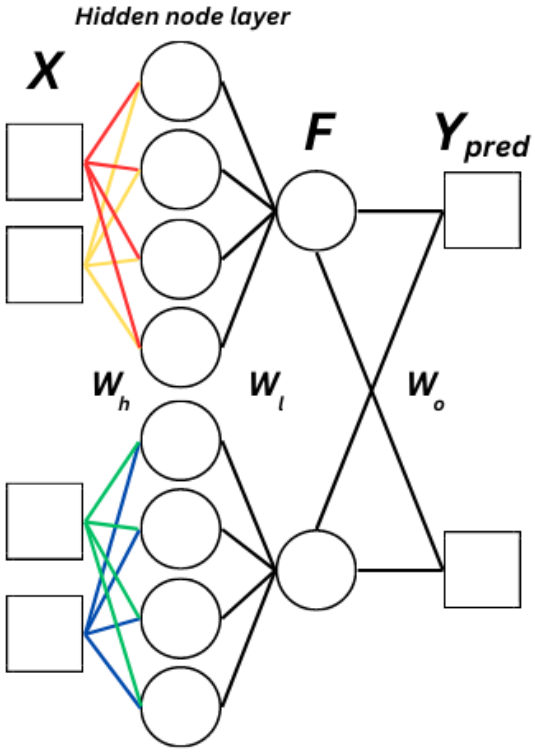

Figure 2: Illustration of forward propagation in the NN-ERA framework.

Similarly, the NN-ERA utilizes the gradient descent algorithm, highlighted in equation (5) to optimize the model parameters according to the error of the model, $\mathbf{E}$.

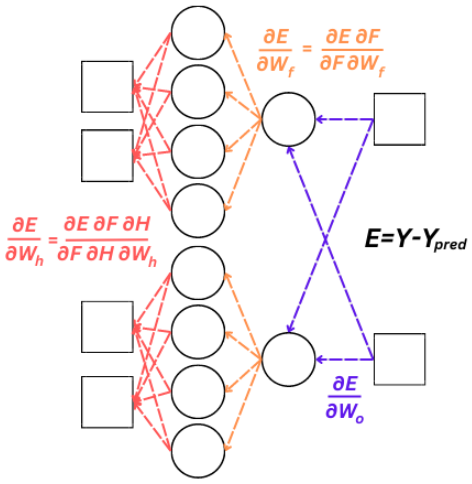

Figure 3: Illustration of backpropagation in the NN-ERA framework.

3.2 k-Means Clustering

The k-means algorithm is an unsupervised clustering algorithm that partitions data into k specified groups MacQueen, 1967. Of the several types of clustering algorithms, k-means clustering is the most popular due to its simple implementation and its guaranteed convergence. Similar to the neural network and NN-ERA, the k-means technique follows an iterative algorithm with the objective to minimize the within-cluster variance across clusters. The k-means objective function can be described as follows:

$$\arg\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} ||x - \mu_c||^2 \tag{11}$$

where C represents the number of clusters, i represents the ith cluster and μ represents the mean of the points of cluster C. The mean of the points of any cluster C is computed as:

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{12}$$

Albeit useful, the k-means approach does not consider any theoretical relationship between variables. As a data-driven approach, no insights are drawn about the influence of the variables on the clusters. The motivation of the proposed model is to leverage the ability to identify heterogenous clusters and do so while considering the influences of a theoretical model upon the groups.

3.3 Cluster-wise NN-ERA

Algorithm 1 Cluster-wise NN-ERA

1. Initialize general NN-ERA.
2. Compute general NN-ERA forward propagation.
3. Initialize cluster centroids according to predicted output.

while improvement > 0.0001 **do**

4a. Compute general NN-ERA forward propagation.
4b. Perform cluster assignment.
4c. Update cluster centroids.
4d. Perform cluster-wise NN-ERA.
4e. Perform general NN-ERA backpropagation.

end while

The cluster-wise NN-ERA combines the k-means algorithm with the theoretical basis of the NN-ERA. The unified technique follows an iterative algorithm that considers the predicted output of the NN-ERA to identify k clusters centroids. The points associated with each centroid are then computed through cluster-wise NN-ERA models to extract their respective theoretical relationships. The proposed model unifies the clustering and NN-ERA framework through a single objective function which can be described as

$$\arg\min_{C} \sum_{i=1}^{C} \sum_{e \in C_i} ||x - \mu_c||^2 \tag{13}$$

which follows a similar objective function as the k-means approach, but based on the error of the NN-ERA, $\mathbf{E_i}$. The error can be expressed as

$$E_i = y - \hat{y} \tag{14}$$

where y represents the matrix of true output values and $\hat{y}$ represents the predicted output of the model. Additionally, to compute the cluster centroid, μ_c, the computation can be described as

$$\mu_c = \frac{1}{|C_i|} \sum_{e \in C_i} e. \tag{15}$$

Given the integration of the NN-ERA, the proposed method utilizes activation functions at the model's hidden layer to capture nonlinear relationships between observed and latent variables. Moreover, the proposed model also contains four nodes at the hidden layer for each latent variable. Entirely, the cluster-wise NN-ERA approach is not only capable of identifying cluster-specific theoretical differences, but also capable of identifying patterns previously limited with past models.

4 Simulation Study

4.1 Simulation Procedure

To assess the feasibility of the proposed method, a simulation study was conducted. To gauge the performance of the unified method, the model was compared to the step-wise NN-ERA approach. In both cases, the cluster-wise NN-ERA structure was identical to the step-wise approach such that two output variables were estimated according to two latent variables and each of the latent variables were loaded based on four hidden layer nodes and two observed variables. As a step-wise procedure, however, k-means clustering component was initially performed and clustering information was extracted. The theoretical relationship of each cluster was then estimated using separate NN-ERA models. In comparison, the unified approach updates cluster identification while estimating the theoretical relationship of each cluster. Additionally, the unified approach also considers centroids based on the error of the general NN-ERA to identify cluster belonging.

4.2 Data Simulation

The cluster-wise and step-wise approaches follow a model structure containing two observed variables and four hidden nodes per latent variable, with a total of two latent variables. From the latent variables, the models estimate two output variables. The data for the simulation was generated according to three variables: sample size (100, 300, 1000), cluster size (50%, 75%), and data variance (low, high). Of each set, 500 iterations were generated, producing 6000 simulated datasets for comparison.

4.3 Metrics

Across data settings, the models were compared according to their mean squared error (MSE), relative bias, and classification rate. The mean squared error was calculated as follows

$$MSE = \frac{1}{n}\sum_{i=1}^{N}(y - \hat{y})^2 \tag{16}$$

where n represents the number of data points, y represents the true output and $\hat{y}$ represents the predicted output of the model. MSE provides insights regarding the average amount of error in the model and is always non-negative, where values closer to zero are considered better. The relative bias was computed as follows

$$RB = \frac{\hat{y} - y}{y} * 100 \tag{17}$$

which signifies the percent comparison between the model's predicted estimates and the ground truth. Whereas the classification rate of the models was

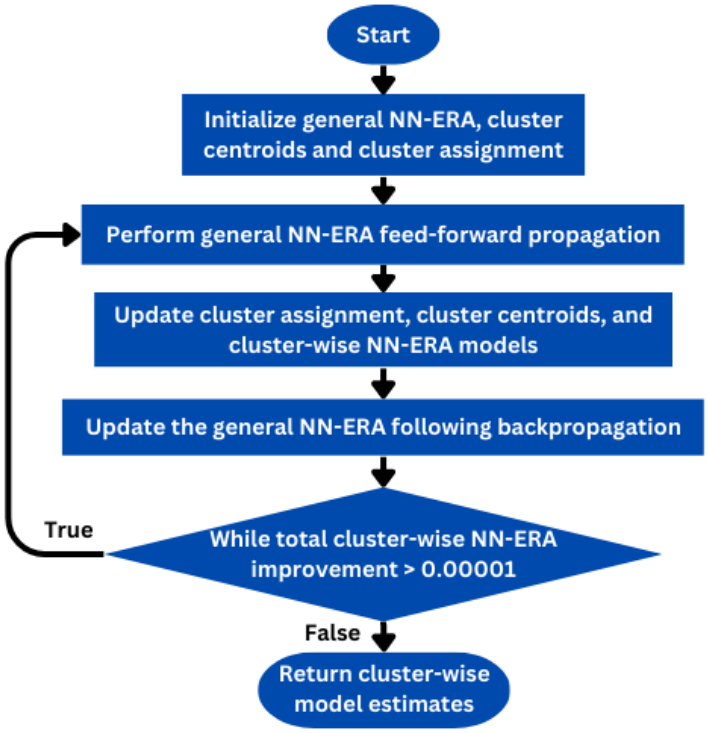

Figure 4: Illustration of the cluster-wise NN-ERA algorithm.

determined by how well the model correctly predicted the cluster belonging to each data point to its respective cluster.

5 Results

5.1 Mean squared error

Across data settings, the proposed model outperforms the step-wise approach, producing a lower MSE suggesting that the unified approach is able to create a better fit to the data. Comparing the 50% cluster separation, in the simulated data of 100 samples, the proposed method produced an MSE of $M = 3.641, SD = 0.725$ and $M = 3.779, SD = 0.734$ for the low and high variance settings, in comparison to the step-wise approach of MSEs of $M = 5.195, SD = 0.545$ and $M = 5.105, SD = 0.558$ respectively. Whereas in the simulated data of 300 samples, the unified method produced MSEs of $M = 3.639, SD = 0.481$ and $M = 3.793, SD = 0.509$ for the low and high variance settings, while the step-wise approach produced MSEs of $M = 5.225, SD = 0.327$ and $M = 5.14, SD = 0.298$ respectively. In the simulated data of 1000

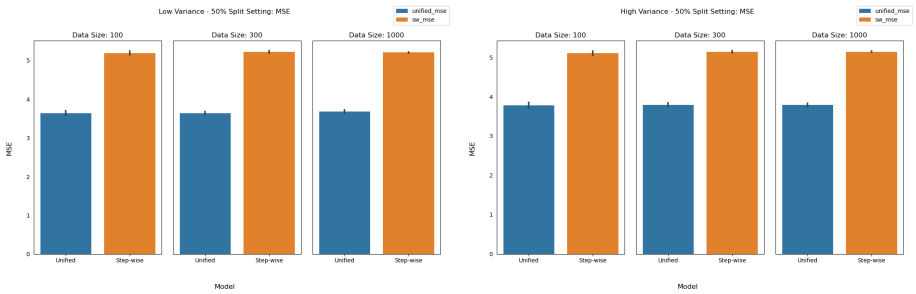

Figure 5: MSE in the 50% cluster separation data settings.

samples, the unified method produced MSEs of $M = 3.683, SD = 0.502$ and $M = 3.786, SD = 0.47$ for the low and high variance settings, in comparison to the MSEs of the step-wise approach of $M = 5.213, SD = 0.174$ and $M = 5.141, SD = 0.18$ respectively.

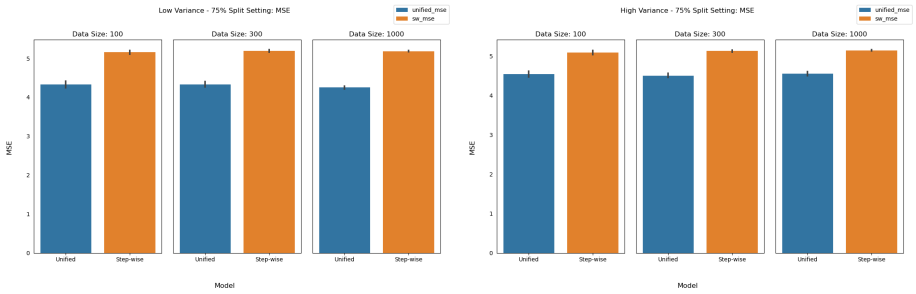

Figure 6: MSE in the 75% cluster separation settings.

However, it should be noted that the performance of the proposed model appears to be closer to the step-wise approach in cases when the cluster sizes vary (ie, in the 75% data split setting). In the simulated data of 100 samples with 75% cluster separation, the unified method produced an MSE of $M = 4.328, SD = 1.034$ and $M = 4.539, SD = 0.856$ for the low and high variance settings, in comparison to the step-wise approach of MSEs of $M = 5.16, SD = 0.573$ and $M = 5.141, SD = 0.18$ respectively. In the same cluster separation of 300 samples, the unified method produced an MSE of $M = 4.328, SD = 0.751$ and $M = 4.503, SD = 0.597$ for the low and high variance settings, while the step-wise approach produces MSEs of $M = 5.191, SD = 0.344$ and $M = 5.126, SD = 0.332$ respectively. In the largest sample size in the 75% cluster separation, the unified method produced an MSE of $M = 4.254, SD = 0.442$ and $M = 4.552, SD = 0.625$ for the low and high variance settings, in comparison to the step-wise approach of MSEs of $M = 5.182, SD = 0.188$ and $M = 5.142, SD = 0.18$ respectively.

5.2 Relative Bias

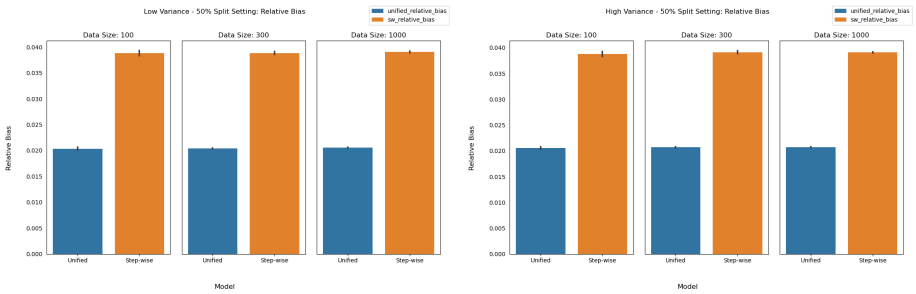

Figure 7: Relative bias in the 50% cluster separation data settings.

In the 50% cluster separation setting of 100 samples, the average relative bias across 500 iterations was $M = 0.02, SD = 0.003$ and $M = 0.021, SD = 0.003$ in the low and high variance settings. The average relative bias of the step-wise approach in an identical setting was $M = 0.039, SD = 0.006$ and $M = 0.039, SD = 0.006$ in the low and high variance settings respectively. In the setting of 300 samples, the average relative bias of the proposed model was $M = 0.02, SD = 0.002$ and $M = 0.021, SD = 0.002$ for the low and high variance data settings respectively. In comparison, the average relative bias of the step-wise approach was $M = 0.039, SD = 0.003$ and $M = 0.039, SD = 0.003$. In the largest sample, the cluster-wise NN-ERA produced an average relative bias of $M = 0.021, SD = 0.001$ and $M = 0.021, SD = 0.001$, while the step-wise approach produced an average relative bias of $M = 0.039, SD = 0.002$ and $M = 0.039, SD = 0.002$.

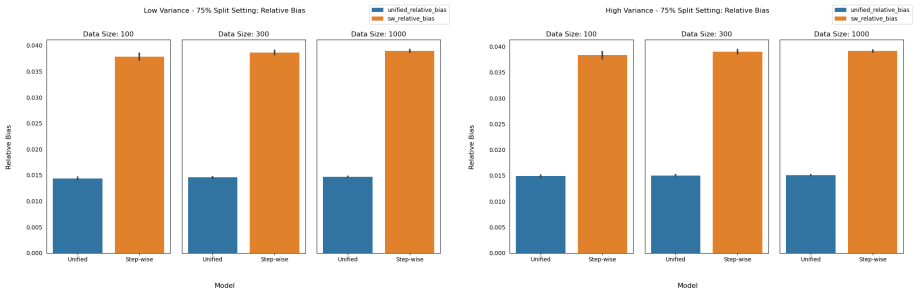

Figure 8: Relative bias in the 75% cluster separation data settings.

In the 100 sample size, 75% cluster separation setting, the average relative biases of the approaches were $M = 0.014, SD = 0.002$ and $M = 0.015, SD = 0.002$ for the unified approach and $M = 0.038, SD = 0.007$ and $M = 0.038, SD = 0.008$ for the step-wise approach in the low and high variance settings respectively. In the 300 sample setting, the average relative biases of the unified

approach were $M = 0.015, SD = 0.001$ and $M = 0.015, SD = 0.001$ for the low and high variance settings. In the step-wise approach, the average relative biases for the low and high variance settings were $M = 0.039, SD = 0.005$ and $M = 0.039, SD = 0.004$. In the 1000 sample set, the proposed method produced average relative biases of $M = 0.015, SD = 0.001$ and $M = 0.015, SD = 0.001$ for the low and high variance settings, compared to $M = 0.039, SD = 0.003$ and $M = 0.039, SD = 0.002$ of the step-wise approach.

5.3 Classification Rate

Across the twelve data settings and averaged over 500 data simulations, both the proposed cluster-wise NN-ERA and the step-wise approaches classified 100% of the data points to their respective clusters.

6 General Discussion and Conclusion

The results of the simulation study suggest that the proposed cluster-wise NN-ERA method is a feasible means of integrating clustering approaches to the hybrid NN-ERA framework. Across the simulation settings, the proposed method was able to achieve the same level of classification accuracy as the step-wise approach. Similarly, the proposed method produced closer approximations of the data than the step-wise approach according to the MSE and relative bias scores.

The development of the proposed method adds to the growing literature on hybrid models that integrate machine learning into theoretical frameworks (Jacobucci et al., 2016; Serang et al., 2021). Additionally, the cluster-wise NN-ERA also adds to the literature on interpretable machine learning models (Du et al., 2019; Murdoch et al., 2019). Given the integration of both unsupervised and supervised learning in a theoretical framework, it is worth considering how other models, such as attention, can be integrated. Nevertheless, the proposed research is not without limitations.

The current study is limited to the generated simulation data. Future research should consider applying the cluster-wise NN-ERA method to empirical data to better evaluate the model's capabilities. Within the simulation, the current study is also limited in the number of clusters analyzed, warranting an evaluation of the proposed method according to several k clusters. Based on the preliminary results, the NN-ERA framework could be integrated with other clustering approaches to address the limitations of the k-means approach. Given that the k-means algorithm does not consider probabilistic memberships, future research could investigate the integration of probabilistic clustering approaches such as fuzzy clustering. Moreover, to address the limitation of spherical clusters in the k-means approach, density-based methods such as DBSCAN could also be applied. Entirely, a clustering ensemble could be applied to address the limitations of each of the clustering methods and to reach a consensus on the underlying data structure.

In the proposed research, we developed a hybrid machine learning method that integrates the k-means algorithm with the NN-ERA framework. The model was comparable to the step-wise k-means NN-ERA approach regarding classification and outperformed the step-wise approach according to the MSE and relative bias metrics. The method allows for the identification of heterogeneous clusters within a theoretical framework. Through the proposed method, social science researchers can explore the underlying structure of the data whilst maintaining the theoretical relationships imposed by the researcher. From such insights, we hope that this model can be applied to accelerate the development of new generalizable social science theories.

References

Abbasi, G. A., Tiew, L. Y., Tang, J., Goh, Y.-N., & Thurasamy, R. (2021). The adoption of cryptocurrency as a disruptive force: Deep learning-based dual stage structural equation modelling and artificial neural network analysis (D. Dragan, Ed.). *PLOS ONE*, *16*(3), e0247582. https://doi.org/10.1371/journal.pone.0247582

Aktepe, A., Ersöz, S., & Toklu, B. (2015). Customer satisfaction and loyalty analysis with classification algorithms and Structural Equation Modeling. *Computers & Industrial Engineering*, *86*, 95–106. https://doi.org/10.1016/j.cie.2014.09.031

Albahri, A. S., Alnoor, A., Zaidan, A. A., Albahri, O. S., Hameed, H., Zaidan, B. B., Peh, S. S., Zain, A. B., Siraj, S. B., Masnan, A. H. B., & Yass, A. A. (2022). Hybrid artificial neural network and structural equation modelling techniques: A survey. *Complex & Intelligent Systems*, *8*(2), 1781–1801. https://doi.org/10.1007/s40747-021-00503-w

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science [Series Title: Unsupervised and Semi-Supervised Learning]. In M. W. Berry, A. Mohamed, & B. W. Yap (Eds.), *Supervised and Unsupervised Learning for Data Science* (pp. 3–21). Springer International Publishing. https://doi.org/10.1007/978-3-030-22475-2_1

Brusco, M. J., Shireman, E., & Steinley, D. (2017). A comparison of latent class, K-means, and K-median methods for clustering dichotomous data. *Psychological Methods*, *22*(3), 563–580. https://doi.org/10.1037/met0000095

Caruana, R., & Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning - ICML '06*, 161–168. https://doi.org/10.1145/1143844.1143865

Cheung, M. W.-L., & Jak, S. (2016). Analyzing Big Data in Psychology: A Split/Analyze/Meta-Analyze Approach. *Frontiers in Psychology*, *7*. https://doi.org/10.3389/fpsyg.2016.00738

Curry, H. B. (1944). The method of steepest descent for non-linear minimization problems. *Quarterly of Applied Mathematics*, *2*(3), 258–261. https://doi.org/10.1090/qam/10667

de Leeuw, J., Young, F. W., & Takane, Y. (1976). Additive structure in qualitative data: An alternating least squares method with optimal scaling features. *Psychometrika*, *41*(4), 471–503. https://doi.org/10.1007/BF02296971

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77. https://doi.org/10.1145/3359786

Ebrahimi, P., Salamzadeh, A., Soleimani, M., Khansari, S. M., Zarea, H., & Fekete-Farkas, M. (2022). Startups and Consumer Purchase Behavior:

Application of Support Vector Machine Algorithm. *Big Data and Cognitive Computing*, *6*(2), 34. https://doi.org/10.3390/bdcc6020034

Elgendy, N., & Elragal, A. (2014). Big Data Analytics: A Literature Review Paper [Series Title: Lecture Notes in Computer Science]. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, A. Kobsa, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, D. Terzopoulos, D. Tygar, G. Weikum, & P. Perner (Eds.), *Advances in Data Mining. Applications and Theoretical Aspects* (pp. 214–227). Springer International Publishing. https://doi.org/10.1007/978-3-319-08976-8_16

Enders, C. K., & Tofighi, D. (2007). Centering predictor variables in cross-sectional multilevel models: A new look at an old issue. *Psychological Methods*, *12*(2), 121–138. https://doi.org/10.1037/1082-989X.12.2.121

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 226–231.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452–459. https://doi.org/10.1038/nature14541

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT Press.

Grimmer, J., Roberts, M. E., & Stewart, B. M. (2021). Machine Learning for Social Science: An Agnostic Approach. *Annual Review of Political Science*, *24*(1), 395–419. https://doi.org/10.1146/annurev-polisci-053119-015921

Hindman, M. (2015). Building Better Models: Prediction, Replication, and Machine Learning in the Social Sciences. *The ANNALS of the American Academy of Political and Social Science*, *659*(1), 48–62. https://doi.org/10.1177/0002716215570279

Jacobucci, R., Grimm, K. J., & McArdle, J. J. (2016). Regularized Structural Equation Modeling. *Structural Equation Modeling: A Multidisciplinary Journal*, *23*(4), 555–566. https://doi.org/10.1080/10705511.2016.1154793

Jang, J., & Jiang, H. (2019). DBSCAN++: Towards fast and scalable density clustering [arXiv:1810.13105 [cs, stat]]. Retrieved March 13, 2023, from http://arxiv.org/abs/1810.13105

Jöreskog, K. G. (1970). ESTIMATION AND TESTING OF SIMPLEX MODELS. *British Journal of Mathematical and Statistical Psychology*, *23*(2), 121–145. https://doi.org/10.1111/j.2044-8317.1970.tb00439.x

Le, N., Rathour, V. S., Yamazaki, K., Luu, K., & Savvides, M. (2022). Deep reinforcement learning in computer vision: A comprehensive survey. *Artificial Intelligence Review*, *55*(4), 2733–2819. https://doi.org/10.1007/s10462-021-10061-9

MacCallum, R. C., & Austin, J. T. (2000). Applications of Structural Equation Modeling in Psychological Research. *Annual Review of Psychology*, *51*(1), 201–226. https://doi.org/10.1146/annurev.psych.51.1.201

MacQueen, J. (1967). Classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (pp. 281–297).

Madhulatha, T. S. (2012). An Overview on Clustering Methods [Publisher: arXiv Version Number: 1]. https://doi.org/10.48550/ARXIV.1205.1117

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, *5*(4), 115–133. https://doi.org/10.1007/BF02478259

McCutcheon, A. L. (1987). *Latent class analysis*. Sage.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, *116*(44), 22071–22080. https://doi.org/10.1073/pnas.1900654116

Murtagh, F., & Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. *WIREs Data Mining and Knowledge Discovery*, *2*(1), 86–97. https://doi.org/10.1002/widm.53

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

Pedregal, P. (2003). *Introduction to optimization*. Springer.

Reier Forradellas, R. F., Náñez Alonso, S. L., Jorge-Vazquez, J., & Rodriguez, M. L. (2020). Applied Machine Learning in Social Sciences: Neural Networks and Crime Prediction. *Social Sciences*, *10*(1), 4. https://doi.org/10.3390/socsci10010004

Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O. P., Tiwari, A., Er, M. J., Ding, W., & Lin, C.-T. (2017). A review of clustering techniques and developments. *Neurocomputing*, *267*, 664–681. https://doi.org/10.1016/j.neucom.2017.06.053

Seifert, T. L. (1997). Academic goals and emotions: Results of a structural equation model and a cluster analysis. *British Journal of Educational Psychology*, *67*(3), 323–338. https://doi.org/10.1111/j.2044-8279.1997.tb01247.x

Serang, S., Jacobucci, R., Stegmann, G., Brandmaier, A. M., Culianos, D., & Grimm, K. J. (2021). Mplus Trees: Structural Equation Model Trees Using Mplus. *Structural Equation Modeling: A Multidisciplinary Journal*, *28*(1), 127–137. https://doi.org/10.1080/10705511.2020.1726179

Sharma, S., Sharma, S., & Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology*, *04*(12), 310–316. https://doi.org/10.33564/IJEAST.2020.v04i12.054

Szandała, T. (2021). Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks [Series Title: Studies in Computational Intelligence]. In A. K. Bhoi, P. K. Mallick, C.-M. Liu, & V. E. Balas (Eds.), *Bio-inspired Neurocomputing* (pp. 203–224). Springer Singapore. https://doi.org/10.1007/978-981-15-5495-7_11

Takane, Y., & Hwang, H. (2005). An extended redundancy analysis and its applications to two practical examples. *Computational Statistics & Data Analysis*, *49*(3), 785–808. https://doi.org/10.1016/j.csda.2004.06.004

Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-l. A., Elkhatib, Y., Hussain, A., & Al-Fuqaha, A. (2019). Unsupervised Machine Learning for Networking: Techniques, Applications and Research Challenges. *IEEE Access*, *7*, 65579–65615. https://doi.org/10.1109/ACCESS.2019.2916648

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need [arXiv:1706.03762 [cs]]. Retrieved March 11, 2023, from http://arxiv.org/abs/1706.03762

Vijayakumar, R., Choi, J. Y., & Jung, E. H. (2022). A Unified Neural Network Framework for Extended Redundancy Analysis. *Psychometrika*. https://doi.org/10.1007/s11336-022-09853-x

Whittingham, H., & Ashenden, S. K. (2021). Hit discovery. In *The Era of Artificial Intelligence, Machine Learning, and Data Science in the Pharmaceutical Industry* (pp. 81–102). Elsevier. https://doi.org/10.1016/B978-0-12-820045-2.00006-4

Wright, S. (1934). The Method of Path Coefficients. *The Annals of Mathematical Statistics*, *5*(3), 161–215. https://doi.org/10.1214/aoms/1177732676

Yarkoni, T., & Westfall, J. (2017). Choosing Prediction Over Explanation in Psychology: Lessons From Machine Learning. *Perspectives on Psychological Science*, *12*(6), 1100–1122. https://doi.org/10.1177/1745691617693393

Yim, O., & Ramdeen, K. T. (2015). Hierarchical Cluster Analysis: Comparison of Three Linkage Measures and Application to Psychological Data. *The Quantitative Methods for Psychology*, *11*(1), 8–21. https://doi.org/10.20982/tqmp.11.1.p008

Zobair, K. M., Sanzogni, L., Houghton, L., & Islam, M. Z. (2021). Forecasting care seekers satisfaction with telemedicine using machine learning and structural equation modeling (C. Delcea, Ed.). *PLOS ONE*, *16*(9), e0257300. https://doi.org/10.1371/journal.pone.0257300