Robustness of autoencoders for establishing psychometric properties based on small sample sizes: Results from a monte carlo simulation study and a sports fan curiosity study

Yen-Kuang, Lin[1], Chen-Yin, Lee[2], Chen-Yueh, Chen[3]

[1] Graduate Institute of Athletics and Coaching Science , National Taiwan Sport University, Taoyuan, Taiwan.

[2] Department of English Studies, Ming-Dao University, Changhua, Taiwan

[3] Department of Leisure and Recreation Industry Management, National Taiwan Sport University, Taoyuan, Taiwan

Corresponding Author:

Chen-Yin Lee: inin1210@gmail.com

Chen-Yuen Chen: chenchenyueh@ntsu.edu.tw

Abstract

Background. The principal component analysis (PCA) is known as a multivariate statistical model for reducing dimensions into a representation of principal components. Thus, the PCA is commonly adopted for establishing psychometric properties, i.e., the construct validity. Autoencoder is a neural network model, which has also been shown to perform well in dimensionality reduction. Although there are several ways the PCA and autoencoders could be compared for their differences, most of the recent literature focused on differences in image reconstruction, which are often sufficient for training data. In the current study, we looked at details of each autoencoder classifier and how they may provide neural network superiority that can better generalize non-normally distributed small datasets. **Method.** A Monte Carlo simulation was conducted, varying the levels of non-normality, sample sizes, and levels of communality. The performances of autoencoders and a PCA were compared using the mean square error, mean absolute value, and Euclidian distance. The feasibility of autoencoders with small sample sizes was examined. **Results.** With extreme flexibility in decoding representation using linear and non-linear mapping, this study demonstrated that the autoencoder can robustly reduce dimensions, and hence was effective in building the construct validity with a sample size as small as 100. The autoencoders could obtain a smaller mean square error and small Euclidian distance between original dataset and predictions for a small non-normal dataset. Hence, when behavioral scientists attempt to explore the construct validity of a newly designed questionnaire, an autoencoder could also be considered an alternative to a PCA.

Introduction

Selecting a proper sample size is critical when planning an empirical study. The minimum sample size is often calculated based on selected statistical procedures. For example, inferential statistics based on an independent-sample *t*-test can apply the formula, $N \geq \left(\frac{1.96}{\delta}\right)^2 * \sigma^2$, to achieve the minimum sample size to detect a difference with a satisfactory probability. However, the feasibility of using an autoencoder to establish psychometric properties is not as straightforward. Although an autoencoder has not been used as often to evaluate psychometric properties compared to its alternative, the principal component analysis (PCA), autoencoders have been used in a vast array of scientific fields for dimensional reduction.

1.1 Dimensional reduction

Dimensional reduction refers to the representation of high-dimensional information into a lower-dimensional space without losing an appreciable amount of data. Dimensional reduction can be applied to many methods of multivariate data analyses. For example, factor analysis (FA) is a tool to model interrelationships among items. In an FA, the focus is to partition the variance into either common variance or unique variance. By factor extraction, an FA can reduce the number of variables explaining the variance-covariance among items. Thus, much of the information in the original data can be retained but with fewer dimensions. As a result, this statistical tool, like so many other multivariate models, plays a superlative role in the fields of education, manufacturing industry, bioinformatics, and computer science. Two major challenges are encountered in dimension-reduction tasks. First, in order to evaluate the effectiveness of the dimensional reduction, a method of data visualization is often conducted to visualize the information in the data(Young, Valero-Mora & Friendly, 2011). However, data can only be visualized by humans in very limited lower dimensions of 2D or 3D. Thus, it will sometimes be impossible to graphically evaluate results of dimensional reduction. Second, although the dimension of a dataset can effectively be reduced by a stepwise algorithm, including a forward selection algorithm, backward elimination, stepwise procedure, etc., the impact of each variable being examined is based on its contribution in improving the model fit. However, these stepwise algorithms are only valid when all of the dimensions are independent of each other(Wang, 2008). As a result, alternatives, like an FA, partial least squares, and PCA, can also be adopted. In contrast to a stepwise algorithm where nuisance variables are excluded, the FA and PCA retain all of the original items and combine them to form latent factors or principal components.

In contrast to the PCA and similar algorithms, the autoencoder is in essence a three-layer neural network, which was developed in the 1990s and which has been studied by many researchers. The association between auto-association and singular value decomposition was first examined by Bourlard and Kamp in 1988 (Bourlard & Kamp, 1988). Kramer(1991) attempted to conduct a nonlinear PCA using auto-associative neural networks. Most studies related to autoencoders were restricted to one or two hidden layers mainly due to training difficulties (Wang et al., 2014; Wang, He & D.V. Prkhorov, 2012). In 2013, Le built a nine-layer sparse autoencoder to show that the network could be sensitive to higher-level concepts (2013). Despite recent developments with different architectures of autoencoders, creating a good reconstruction without losing significant information remains a challenging task due to high-dimensionality, small sample sizes, non-linearity, and complex data types.

1.2 Construct Validity

Construct validity is a psychological property defined as the degree to which a measure assesses the theoretical construct intended to be measured (Cronbach & Meehl, 1955). One cannot assess confounding influences of random error without estimating the construct validity when designing a questionnaire. Evaluation of what defines a psychological construct is the determinant of the test performance. A better representation of the latent psychological construct is desirable for almost all psychological tests. Thus, construct validity, in general, is considered the most fundamental aspect of psychometrics. Campbell and Fiske (1959) proposed two views of construct validity: convergent validity and discriminant validity. Convergent validity refers to the degree of agreement among measurements of the same constructs that should be related based on theory. Discriminant validity refers to the distinction of concepts that are not supposed to be related and are in fact, unrelated. Campbell and Fiske developed four steps based on inspecting the multitrait-multimethod (MTMM) matrix to operationally define convergent validity and discriminant validity. Since the concept of construct validity was introduced, an extensive effort has been made ever since to seek numerical representations of the construct validity. Starting with Douglas Jackson who employed a component analysis as an integral part of the development of psychological measures, the PCA has become a standard method for questionnaire development(Jackson, 1970). Traditionally, the PCA and FA are two of the most often employed statistical procedures in the social behavioral sciences commonly used to suggest factor profiles as latent constructs (Sherman, 1986; Yoon et al., 2019; Fontes et al., 2017).

In FAs, the confirmatory factor analysis (CFA) and exploratory factor analysis (EFA) are two methods that facilitate the transition from many observed variables to a smaller number of latent variables. Both FA models are commonly used to address the construct validity. The CFA is a tool that researchers can adopt to test the validity by comparing alternatively proposed a priori models at the latent factor level. Advantages of the CFA for being more informative than Campbell & Fiske's criteria are that it provides a statistical justification of the model fit and the degree of fit for the convergent validity and divergent validity (Bagozzi, R.P., Yi & Phillips, 1991)

In addition to the CFA, there are several statistical models based upon which the factors explored by a questionnaire are validated. For example, the PCA is a mathematical algorithm in which observations are described by several inter-correlated quantitatively dependent variables. A PCA is the default data-reduction technique in SPSS software and was adopted by many researchers, including Mohammadbeigi et al. (2015), and Parker et al. (2010). A PCA can be conducted to examine the construct validity due to the PCA's ability to integrate the full bivariate cross-correlation matrix of all item-wise measurements through dimension reduction. Its goal is to extract important information from the total number of observed variables, and represent it as a set of new orthogonal variables called principal components. These principal components, or latent variables, summarize the observed data table and display the pattern of similarity of the observations (Abdi & Williams, 2010).

Most educational researchers, behavioral scientists, and social science researchers treat uncorrelated principal components as independent identities. However, the property of principal components being independent of each other only holds when the principal components are uncorrelated, and the multivariate items are normally distributed (Kim & Kim, 2012). If the input data are not normally distributed, the variance explained by one of the traits will overlap that of another trait. PCAs have also been criticized due to limited linear mapping representation.

122   Based on the underlying definition of dimensional reduction, it is less informative to aggregate all
123   scales into a single latent variable score. A set of items or scales may share similar conceptual
124   underpinnings but not necessarily be identical (Stangor, 2014). Using a PCA, a large number of items can
125   be reduced to fewer components with possibly more variance explained than with other methods of
126   factoring (Hamzah, Othman & Hassan, 2016).

127   1.3 Autoencoder

128   Autoencoders are characterized by their function of extracting important information and
129   representing it in another space. Such a network consists of three symmetrical layers: input, hidden, and
130   output layers (Hinton & Zemel, 1994). An autoencoder attempts to approximate the original data so that
131   the output is similar to the input after feed-forward propagation. The input is projected to the hidden layer
132   that is commonly designed to be of a lower dimensionality. After information is passed through the
133   hidden layer, the output of the network should ideally resemble the original input as closely as possible.
134   As a result, the latent space contains all of the necessary information to describe the data (Ladjal, Newson
135   & Pham, 2019). In a simple autoencoder framework, each neuron is fully connected to all neurons in the
136   previous layer, where neurons in a single layer function completely independently and share no
137   connections. As illustrated by from Meng, Ding & Xue (2017), autoencoder tries to learn function S($\cdot$)
138   such that:

139    $$S_{W,W',b1,b2}(X) \approx X \qquad\qquad\qquad \text{Eq(1)}$$

140   W is weight matrix connected input layer and hidden layer while W' is weight matrix connected
141   hidden layer and output layer. b_1 and b_2 are bias vectors of hidden layer and output layer. S($\cdot$) can be
142   divided into two phases: from input layer to hidden layer is encoding phase (Eq2) and from hidden to
143   output layer is decoding phase (Eq3).

144    $$h = f(W \times X + b_1) \qquad\qquad\qquad \text{Eq(2)}$$

145    $$Y = g(W' \times h + b2) \qquad\qquad\qquad \text{Eq(3)}$$

146   1.4 Autoencoder versus PCA

147   When one is interested in establishing the construct validity, it is often intuitive to apply a PCA to
148   extract latent factors. It quickly becomes apparent that the PCA shares lots of similarities with an
149   autoencoder. Both methods can serve as tools for feature generation and selection by their ability to
150   reduce dimensions. Despite autoencoder neural networks bearing a significant resemblance to PCAs,
151   there is one major difference between these two networks. In contrast to a PCA, an autoencoder applies a
152   non-linear transformation to the input, and so the autoencoder could be more flexible. That is, although
153   the PCA can effectively reduce the linear dimensionality, it still suffers when relationships among the
154   variables are not linear. Aside from the linearity restriction, a PCA may fail by its loose assumption about
155   the input data distribution. As Shlen (2014) pointed out, even though the PCA algorithm is in essence
156   completely nonparametric, because a PCA is unconcerned with the source of the data, it might not capture
157   key features of data variations. That is, a PCA makes no assumptions about the distribution of the data.
158   However, only when the data are assumed to be normal from a multivariate perspective will the joint
159   distribution of the principal components be normal from a multivariate perspective. Then, the principal
160   components will have an obvious geometrical interpretation where the first component can be determined

161 by locating the chord of maximum distance in the ellipsoid $(x - \mu)^T \Sigma^{-1}(x - \mu) = $ constant (Chatfield &
162 Collins, 1981).
163 As a result, we can directly compare various forms of autoencoders to a PCA when we attempt to build
164 the construct validity of a small sample when the data are not normally distributed. Four different forms
165 of autoencoders are considered in this study, including a simple autoencoder with a single-layered
166 autoencoder and three other candidates briefly described as follows.

167 1.5 Tie-weighted autoencoder

168   An autoencoder is a neural network with a symmetrical structure. Although the input is compressed
169 and the output is reconstructed through its latent-space representation, there is no guarantee that the
170 weights of the encoder and decoder are identical. Thus, we can impose an additional optimization
171 restriction so that the weights of the decoder layer are tied to the weights of the encoder layer. By tying
172 the weights, the number of parameters that needs to be trained and the risk of overfitting are reduced. Tie-
173 weight autoencoder is the one we set it to be W=W' from Eq(2) and Eq(3) (Meng, Ding & Xue, 2017)

174 1.6 Deep Autoencoder

175   In the autoencoder framework, there is no limitation on the number of layers for the encoder or
176 decoder. That is, the autoencoder can go deep and can be implemented with a stack of layers.
177 Theoretically, the more hidden layers there are, the more features can be learned from the hidden layers.
178 Although the layers can be stacked, the layers are often designed to remain symmetrical with respect to
179 the central layers.

180 1.7 Independent Encoded Autoencoder

181   A preferable feature of a PCA is that the weight vectors are independent of each other. If
182 orthogonality is imposed, each encoded feature explains unique information, and a smaller number of
183 encoder layers can be achieved. Thus, we also adopted an orthogonal autoencoder (COAE) for
184 comparison, which is capable of simultaneously extracting latent embedding and predicting the clustering
185 assignment (Wang et al., 2019)
186

187 1.8 Sample size

188   There are only a few studies concerning the requirement of the sample size on the dimensional
189 reduction performance of a PCA. So there is no consensus as to how large is large enough for conducting
190 a PCA. Forcino (2012) found that a too-small sample size is more likely to lead to erroneous conclusions.
191 Manjarres-Martinez et al. (2012) tested the stability performance of three ordination methods in terms of
192 their bootstrap-generated sampling variances. Bootstrap resampling techniques are used to generate larger
193 samples which may provide more-precise evaluations of the sampling error. Some researchers
194 recommend sample sizes in relation to the number of variables or correlation structure. For example,
195 Hatcher and O'Rouke (2013) suggested that the sample size should be larger than five times the number
196 of variables. Hutcheson and Sofroniou (1999) recommended that a minimum of n=150 is required for a

197   high community correlation structure. Mundfrom et al. (2005) found that $n>100$ was required for medium
198   community while MacCallum et al. (2001) achieved satisfactory results even for data with numbers of
199   items greater than the sample size. In contrast, Yeung and Ruzzo (2001) showed that a PCA is not
200   suitable for dimensionality reduction tasks when p is greater than n. The performance of a PCA is
201   worsened when a nonlinear relationship is present with limited samples.
202   A deep neural network is competitive in solving nonlinear dimensional reductions for high-
203   dimensional data. Although it may seem legitimate that a massive amount of data is required to train a
204   deep neural network, some researchers claim that deep learning can still be adopted even if n is small.
205   Seyfioğlu & Gürbüz (2017) compared a convolutional autoencoder and two convolutional neural
206   networks, VGGNet and GoogleNet, in terms of the training sample sizes. They found that when the
207   sample size exceeded 650, the convolutional autoencoder outperformed transfer learning and random
208   initialization.
209   Although there is a great number of dimensionality reduction algorithms being developed, the
210   feasibility of their use with small-sample, non-normal data is still unknown. A limited sample size and an
211   non-linear data distribution may also increase the likelihood of overfitting and decrease the accuracy. To
212   overcome the pitfalls of the sample size issue, the principal objective of the current study was to examine
213   the influence of sample size on the latent structure of PCAs and autoencoders using a Monte Carlo
214   simulation. The performances of the PCA and various transformations by autoencoders were evaluated
215   using both simulated data and a real dataset pertaining to quantifying the concept of curiosity.
216

217   **Materials & Methods**
218   The detailed design of the simulation consisted of three major states (Figure 1): data generation,
219   dimensionality-reduction algorithms, and performance evaluation. The input dataset was first divided
220   into two sub-datasets: a training set and testing set. Then various forms of autoencoders along with a
221   PCA were applied to select desirable encoded dimensions of attributes. Finally, for latent
222   dimensionality classification of the obtained reduced-dimensional data, a reconstruction error was
223   applied to evaluate the algorithms.

224   ## 2.1 Data Generation

225   A Monte Carlo simulation was used for this study. To avoid the overfitting issue, separate datasets were
226   used for the dimension-reduction algorithms. Each simulated dataset was divided into two parts: 80% of
227   samples were used as a training set, and the remaining 20% were used as a test set. The dimension-
228   reduction techniques, including various type of autoencoders and a PCA, were trained on the training
229   dataset. After the classifier was built, the testing data were used to test the effectiveness of the classifier.
230   Each simulated dataset was simulated based on its degree of non-normality, correlation among items, and
231   sample size. Correlation matrices of continuous variables, each representing a questionnaire correlation
232   structure, were generated for each condition by three manipulated variables: the degree of communality,
233   the degree of non-normality, and the sample size. The data generation python code was stored in Github,
234   and can be assessed at
235   https://github.com/robbinlin/data-generation-
236   /blob/e94206b6a16751961c3db57fbe93017dc050d746/data_generation_20211005.ipynb

2.2 Non-normality

237   A variety of mathematical algorithms have been developed over the years to simulate conditions of
238   non-normal distributions (Fan et al., 2002; Fleishman, 1978; Ramberg et al., 1979; Schmeiser & Deutsch,
239   1977). Fleishman (1978) also introduced a method for generating sample data from a population with
240   desired degrees of skewness and kurtosis. That method uses a cubic transformation to transform a
241   standard univariate normally distributed variable to obtain a nonnormal variable with specified degrees of
242   skewness and kurtosis. The transformation developed by Fleishman takes the form of

$$Y = a + bZ + cZ^2 + dZ^3; \qquad \qquad \text{Eq4}$$

245   where Y is the transformed non-normal variable, Z is a standard normal random variable, and a, b, c, and
246   d are coefficients needed for transforming the unit around the unit normal to a non-normal variable with
247   specified degrees of population skewness and kurtosis (Byrd, 2008).
248   These coefficients were tabulated in Fleishman (1978) for selected combinations of degrees of skewness
249   and kurtosis. Fleishman (1978) derived a system of nonlinear equations that given the target distribution
250   mean, variance, skewness, and kurtosis, could be solved for coefficients to produce a third-order
251   polynomial approximation to the desired distribution(Fan et al., 2002).

2.3 Correlation structure

253   In order to generate non-normal correlated observation, the interaction between inter-variable
254   correlation and degree of non-normality needs to be considered since difference combination of inter-
255   variable correlations and non-normality conditions would cause sample data to deviate from the specified
256   correlation pattern. The abovementioned Fleishman's method can be extended to multivariate non-normal
257   data with a specified correlation(Wicklin, 2013). For example, two non-normal variables, Y1 and Y2, can
258   be generated with specified skewness and kurtosis from equation 4. i.e.,

$$Y_1 = a_1 + b_1 Z + c_1 Z^2 + d_1 Z^3 \qquad \qquad \text{Eq5}$$

$$Y_2 = a_2 + b_2 Z + c_2 Z^2 + d_2 Z^3 \qquad \qquad \text{Eq6}$$

261   Coefficients a_1, b_1, c_1, d_1, a_2, b_2, c_2, and d_2 can be derived from Fleishman's table once the degree of
262   skewness and kurtosis are known. After these coefficients (a_i, b_i, c_i, d_i) are found, the intermediate
263   correlations could be derived by specifying R_{x1x2}, the population correlation between two non-normal
264   variables Y_1 and Y_2. Vale and Maurelli (1983) demonstrated that through the following relationship,

$$R_{x_1 x_2} = \rho(b_1 b_2 + 3b_1 d_2 + 3d_1 b_2 + 9d_1 d_2) + \rho^2(2c_1 c_2) + \rho^3(6d_1 d_2), \qquad \text{Eq7}$$

266   the intermediate correlation, ρ, can be derived. These coefficients in the Fleishman power transformation
267   above are required to derive intermediate correlations, ρ. After all of the intermediate correlation
268   coefficients are assembled into an intermediate correlation matrix, this intermediate correlation matrix is
269   then used to extract factor patterns to transform uncorrelated items into correlated items (Vale & Maurelli,
270   1983). Kaiser and Dickman (1962) presented a matrix decomposition procedure that imposes a specified
271   correlation matrix on a set of uncorrelated random normal variables specified with population correlations
272   *R* as represented by the imposed correlation matrix. The basic matrix decomposition procedure takes the
273   following form (Kaiser & Dickman, 1962):

$$\hat{R}_{k*N} = F_{k*k} \times \hat{X}_{k*N}; \qquad \qquad \text{Eq8}$$

275   where k is the number of variables, N is the number of sample, $\hat{R}$ is the resulting data matrix, F is the
276   principal component factor pattern coefficients obtained by principal component factorization to the

277   desired population matrix R, and X are uncorrelated k random variables, with N observations. After the
278   intermediate correlations are derived from Vale and Maurelli 's formula using iterative Newton-Raphson
279   method, we then apply the assembled intermediate correlation matrix to Kaiser and Dickman's method as
280   R in order to generate N sample.
281   In the current study, three correlation structures for 15 items were simulated to approximate three
282   communalities of high, in which communalities were assigned values of 0.8; wide, in which they could
283   have value from 0.6 to 0.9; and low, in which they could have value from 0.3 and 0.5. The communality
284   estimate is the estimated proportion of variance in each variable that is accounted for (Mamdoohi et al,
285   2016). These estimates reflect the proportion of variation in that variable explained by the latent factors
286   (Youn & Pearce, 2013). In other words, a high communality means that if we perform multiple regression
287   of curiosity against the three common factors, we obtain a satisfactory proportion of the variation in
288   curiosity explained by the factor model. These estimates reflect the variance of a variable in common with
289   all others together. The communality is denoted by h^2 and is the summation of the squared correlations of
290   the variable with the factors (Barton, K., Cattell, R. B., & Curran, J. (1973)). The formula for deriving the
291   communalities is $h_j^2 = a_{j1}^2 + a_{j2}^2 + ...a_{jm}^2$ where a denotes the loadings for j variables. The communality
292   levels correspond inversely to levels of importance of unique factors. High communalities imply several
293   variables load highly on the same factor and the model error is low. (MacCallum, Widaman, Preacher &
294   Hong, 2001). These three correlation structures were designed to mimic a three-factor solution and are
295   presented as follows.

296   **Correlation matrix for high communality**

$$
\begin{bmatrix}
1 & 0.7 & 0.7 & 0.7 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.7 & 1 & 0.7 & 0.7 & 07 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.7 & 0.7 & 1 & 0.7 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.7 & 0.7 & 0.7 & 1 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.7 & 0.7 & 0.7 & 0.7 & 1 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0.7 & 0.7 & 0.7 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 1 & 0.7 & 0.7 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 07 & 0.7 & 1 & 0.7 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 07 & 0.7 & 0.7 & 1 & 0.7 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 0.7 & 0.7 & 0.7 & 1 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0.7 & 0.7 & 0.7 & 0.7 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 1 & 0.7 & 0.7 & 0.7 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 0.7 & 1 & 0.7 & 0.7 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 0.7 & 0.7 & 1 & 0.7 \\
0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.7 & 0.7 & 0.7 & 0.7 & 1 \\
\end{bmatrix}
$$

297

298   **Correlation matrix for intermediate communality**

$$
\begin{bmatrix}
1 & 0.9 & 0.8 & 0.7 & 0.6 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.9 & 1 & 0.7 & 0.6 & 0.5 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.8 & 0.7 & 1 & 0.5 & 0.4 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.7 & 0.6 & 0.5 & 1 & 0.3 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.6 & 0.5 & 0.4 & 0.3 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.9 & 0.8 & 0.7 & 0.6 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.9 & 1 & 0.7 & 0.6 & 0.5 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.8 & 0.7 & 1 & 0.5 & 0.4 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.7 & 0.6 & 0.5 & 1 & 0.3 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.6 & 0.5 & 0.4 & 0.3 & 1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1 & 0.9 & 0.8 & 0.7 & 0.6 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.9 & 1 & 0.7 & 0.6 & 0.5 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.8 & 0.7 & 1 & 0.5 & 0.4 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.7 & 0.6 & 0.5 & 1 & 0.3 \\
0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.6 & 0.5 & 0.4 & 0.3 & 1 \\
\end{bmatrix}
$$

299

300   **Correlation matrix for weak communality**

$$
\begin{bmatrix}
1 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.5 & 1 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.5 & 0.5 & 1 & 0.5 & 0.4 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.5 & 0.5 & 0.5 & 1 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.5 & 0.5 & 0.4 & 0.3 & 1 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 1 & 0.5 & 0.5 & 0.5 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 1 & 0.5 & 0.4 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 0.5 & 1 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 0.4 & 0.3 & 1 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1 & 0.5 & 0.5 & 0.5 & 0.5 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 1 & 0.5 & 0.5 & 0.5 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 1 & 0.5 & 0.4 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 0.5 & 1 & 0.3 \\
0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.5 & 0.5 & 0.4 & 0.3 & 1 \\
\end{bmatrix}
$$

301

302   Factor patterns were estimated with one of these three correlation matrices. These pattern matrices

303   were then adopted to generate 15 correlated normal variables with specified population correlation

304   coefficients, variable means, standard deviations (SDs), skewness, and kurtosis.

2.4 Sample Size

The sample size was chosen based on the recommendations of Mundfrom et al. (2005). Sample size increments were according to the following algorithm:

- When $n<200$, it was increased by 10;
- When $n<500$, it was increased by 50;

2.5 Dimensionality-reduction algorithms

- Simple autoencoder : A simple autoencoder is an autoencoder with two main components: The Encoder and the decoder in addition to the latent-space representation layer also known as the bottle neck layer. Linear activation function and SGD optimizer were adopted. The encoding and decoding algorithms will be chosen to be parametric functions and to be differentiable with respect to the loss function (Chollet, 2016). By minimizing the reconstruction loss, the parameters of the encoder and decoder can be optimized. The basic autoencoder, which refers to simple autoencoder in section 1.4 is a autoencoder with a single fully-connected neural layer as encoder and as decoder (Figure 2a).
- Tie-weighted autoencoder : A tie-weighted autoencoder is a single-layer autoencoder with three neurons in addition to a decoder and an encoder layer. A restriction is imposed so that the weights of the encoder and decoder are identical.
- Deep autoencoder : In order to demonstrate that the autoencoder algorithms do not have to limit to a single layer as encoder or decoder, we could instead use a stack of layers, so called "deep autoencoder" in section 1.6. A deep autoencoder is a sevenlayer autoencoder. The first layer is a dense layer with 11 neurons follow by two layers each with six neurons. Before the decoder, a bottleneck layer with three neurons is included. The architecture is depicted in Figure 2b.
- Independent encoded autoencoder : With this custom layer, we impose penalty on the sum of off-diagonal elements of the encoded features covariance to create uncorrelated features as well as applying orthogonality on both encoder and decoder Weights.

2.6 Real Dataset

In this subsection, we implemented the PCA and autoencoders with a real dataset—a Sports Fan Curiosity dataset--to see how the autoencoders differed from the PCA and provide visualization results. More specifically, we compared the ability to build psychometric properties between the autoencoders and the PCA on a Sports Fan Curiosity questionnaire. Curiosity is considered a fundamental intrinsic motivational subdomain for initiating human exploratory behaviors in many field of study, such as psychology, education, and sports. The Sports Fan Curiosity questionnaire is commonly used in the field of sports management, which is a subset of questionnaires for measuring behavioral intentions. This subcategory was designed to measure and quantify the construct of curiosity. Although leisure management has received greater attention, little is known about how the curiosity construct can be realized with a structured questionnaire. We deployed the autoencoders for this questionnaire and evaluated their ability to identify the latent construct. Items of the Sports Fan Curiosity questionnaire are listed in Table 1. The data set along with the python code was stored in Google Colab and Google drive, and can be assessed at
https://colab.research.google.com/drive/1pC8A10sRVUHDttkLF2ATb51CcpnIzD6u?usp=sharing

346   ## 2.7 Performance Metrics

347   ### 2.7.1 Reconstruction Error:

348   The performance of the establishment of construct validity was evaluated using the mean square
349   error (MSE), mean absolute error (MAE), and the average of the Euclidean distance.
350   Denote inputs to the network by $\underline{X}$ and the outputs of the network by $\underline{y}$. Then the network can be
351   described by mapping from inputs to outputs: $\underline{y}=f(x)$. Reconstruction of the input is then a mapping from
352   outputs to inputs $\hat{X} = g(y)$. It is then reasonable to measure the reconstruction error as $\underline{X} - \hat{X}$ with a given
353   error function, $\epsilon(x)$. In this study, the reconstruction error is defined as the average of the squares of the
354   errors for subject n with dimension m, i.e.,

355

$$\text{MSE} = \frac{1}{n} \cdot \frac{1}{m} \sum_{i=1}^{n} \sum_{j=1}^{m} (X_{i,j} - \hat{x}_{i,j})^2$$

356

357   ### 2.7.2 Mean Absolute Error (MAE):

358   The MAE is defined as the deviation between the paired predicted value $(\hat{X}_i)$ and original value (X).
359   That is, it is the average of the absolute errors for subject n with dimension m:

360

$$\text{MAE} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} |\widehat{x_{i,j}} - x_{i,j}|}{nm}$$

361

362   ### 2.7.3 Normalized-Euclidean Distances (NEDs)

363   The average of the Euclidean distance between each pair of samples in X and $\hat{X}_i$ was also
364   calculated to accommodate possible missing values for subject n with dimension m.

365

$$\text{NED} = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \left(\frac{\hat{x}_{i,j}}{|\hat{x}_{i,j}|} - \frac{x_{i,j}}{|x_{i,j}|} \right)}$$

366   In summary, a $3 \times 3 \times 27$ factorial design was implemented according to the manipulated variables
367   of community level, non-normality level, and sample size, resulting in a total of 243 population
368   conditions (Table 2). Each of the scenario was simulated for 10 times.
369   .
370
371
372
373
374
375

Results

376   **Results**

377   The efficacy of the autoencoders was identified using a Monte Carlo simulation that manipulated
378   five population parameters. The correlation structure was determined with three levels of communality
379   (high, wide, and low), three levels of normality (normal, slightly un-normal, and un-normal), and nine
380   levels of hidden layers of neurons. One should be aware that the current study focused on the feasibility
381   of analyzing small samples. As a result, sample sizes beyond 1000 were not considered.
382   Overall, the autoencoders had smaller reconstruction errors compared to the PCA counterpart.
383   Specifically, the tied-weight autoencoder, deep autoencoder, and independent-feature autoencoder
384   outperformed the PCA in all three communality conditions. The Tied autoencoder was the most stable
385   algorithm among all candidates, as its SD for reconstruction was the smallest (Table 3). In general, the
386   autoencoder family produced smaller MAEs and NEDs compared to the PCA, except for the simple
387   encoder. The simple autoencoder generated larger MSEs and MAEs for all three communality conditions.
388   When the communality was low, performances of the deep autoencoder, tied-weight autoencoder, and
389   independent autoencoder were less affected, in contrast to the PCA and simple encoder.
390   With respect to the input data distributions, it was observed that the performances of the
391   autoencoder family and PCA were not affected by non-normality conditions for this small sample size
392   simulation. However, it was interesting to observe that for data that was extremely un-normal, the tied-
393   weight encoder, deep autoencoder, and independent-feature encoder outperformed the PCA in terms of
394   the MSE and MAE. Of all autoencoder variations, the tied-weight encoder generated the smallest MSE
395   (Table 4). We also evaluated reconstruction errors under different combinations of communality
396   conditions and non-normality conditions. When input data were normally distributed, the resulting
397   reconstruction errors were similar among the three communalities. However, if the correlation structure
398   was low, the MSE increased for non-normal data, even if it only slightly deviated from a normal
399   distribution (Figure 3).
400   Regarding the sample size, the MSE for the autoencoder family decreased as the sample size
401   increased. MSEs for the deep autoencoder, tied-weight encoder, and independent encoder were smaller
402   than that for the PCA for a small sample size. The MSE appeared to decrease to a local minimum when
403   the sample size reached 200 (Figure 4). The simple encoder had the largest MSE, while the tied-weight
404   encoder had the smallest MSE for all of the different sample sizes considered. On the other hand, the PCA
405   had the largest Euclidean distance between the original data and its predictions, while the NEDs were
406   smaller for the tied-weight encoder and deep autoencoder. We further analyzed reconstruction errors for
407   all of the autoencoder family and PCA under different combinations of normality conditions and
408   communality. MSEs for the tied-weight encoder continued to decrease until the sample size reached 200
409   regardless of the deviation from a normal distribution and communality conditions(Figure 4 & Figure 5).
410   A similar trend was also found for the deep autoencoder. The MSE was negatively associated with the
411   sample size, and the MSE stabilized for values of *n* greater than 400. In contrast, the reconstruction error
412   for the PCA was much contaminated by the weak commonality condition. If the correlation structure was
413   low communality, the MSE was always higher than those of the wide and high communality. A similar
414   pattern was also observed for the MAE, as the PCA had a larger MAE when the correlation structure was
415   low communality. In contrast, the deep autoencoder seemed to be insensitive to community conditions
416   and deviations from normality in terms of the MAE compared to that obtained from PCA (Figure 6).
417   There are several algorithms available for determining the number of retaining factors for a PCA,
418   e.g., a scree plot and parallel analysis. However, there are no guidelines to choose the size of the
419   bottleneck layer in the autoencoder. From the simulation results, the autoencoder seemed to perform

420 better compared to the PCA especially when k was small (Figure 7). Even when the number of
421 components was correctly specified by the PCA (which was k=3 in our simulation), all of the autoencoder
422 variations besides the simple encoder outperformed the PCA in terms of MSE.
423
424   1.1 With a Real Dataset

425 We directly compared reconstruction errors of both the deep autoencoder and PCA on the Sports Fan
426 Curiosity questionnaire. Based on previous simulation result, a random sample of 100 subjects was
427 chosen from the original data of 400 subjects to examine the performance with small data. When the PCA
428 was applied to the curiosity data, three components were extracted with an R^2 of 0.53 and an MSE of 0.46
429 while R^2 reached to 60.1% and the MSE decreased to 0.36 for the autoencoder. We adopted t-distributed
430 stochastic neighboring entities (t-SNEs) to visualize the reduced dimension. It was observed that the three
431 latent components could separate the clusters using the PCA, but there was clearly some information that
432 overlapped the extracted components. In contrast to the PCA, the autoencoder could better separate the
433 three underlying constructs, as we saw that there was a significant improvement over the PCA (Figure 8).
434 Regarding the sport fan curiosity questionnaire, if the questionnaire is construct valid, all items together
435 well represent the underlying construct. Based on the weights estimated from the autoencoder's
436 bottleneck layers, each item could be classified into one of the three latent dimensions by its weights in
437 absolute values. For example, the item "I enjoy collecting and calculating statistics of my favorite
438 basketball team" has the highest correlation with construct 1. Similarly, we see that item "Watching
439 basketball games with my friends is joyful" has the highest correlation with construct 2. As such,
440 autoencoders can elucidate how different items and constructs relate to one another and help develop new
441 theories. For example, in Table 5b, the items "I enjoy collecting and calculating statistics of my favorite
442 basketball team," "I enjoy reading articles about basketball players, teams, events, and games," "I am
443 eager to learn more about basketball" and "I enjoy any movement that occurs during a basketball game"
444 appear to have large coefficient on one latent construct, which we assign as "Learning Motivation" factor.
445 "Watching basketball games with my friends is joyful," "I enjoy probing deeply into basketball," and "I
446 often imagine how my favorite basketball team is playing to defeat their opponent" depict the "Social
447 Interaction "factor. The three extracted constructs were knowledge, social interaction, and facility. From
448 the weighted estimate from the bottleneck layer in the autoencoder, each item was classified into one of
449 three latent constructs based on the largest weights among the three clusters. As a result, the first
450 component consisted of items 1, 5, 10, and 11. Items 2, 3, and 6 were classified into cluster 2, while items
451 4, 7, 8, and 9 belonged to cluster 3. Based on the items in each cluster, three latent constructs were
452 identified, i.e., learning motivation, social interaction, and facility (Table 5b).
453   The results from the current study were conceptually equivalent to the three constructs of sport
454 fan curiosity scale developed by Park, Ha and Mahony (2014), i.e., specific information, general
455 information and sport facility information. The slight differences between findings from the study and the
456 work of Park et al may result from different research contexts. More specifically, the real dataset used in
457 the study was collected from a specific sports context (i.e., basketball) whereas Park et al developed the
458 sport fan curiosity scale in a general sport context. However, the findings from the study with small
459 sample size (n=100) yielded psychometrically similar factor structure to the work of Park et al with a
460 much larger sample size (n=407). Consequently, the effectiveness and efficiency of the proposed
461 methodology in the study enrich the relevant literature theoretically and practically.

Discussion

463   The PCA and autoencoders are two common ways of reducing the dimensionality of a high-dimensional
464   feature space. The PCA is a linear-transformation algorithm which projects features onto an orthogonal
465   basis and thus, generates uncorrelated features. An autoencoder is an unsupervised learning technique that
466   can be adopted to tackle the task of representative learning. That is, a bottleneck layer in the network is
467   imposed so that knowledge is compressed through the bottleneck layer, and the output is a representation
468   of the original input. A correlation structure that exists in the data can be learned and consequently
469   leveraged when forcing the input through the network's bottleneck. This characteristic of an autoencoder
470   allows the encoder to compress information into a low-dimensional space to model complicated non-
471   linear associations.
472   One of the major objectives of the current study was to provide a feasibility analysis for adopting a
473   neural network model for establishing construct validity as a psychometric property. Our simulation
474   results indicated that if a low communality correlation structure existed, the reconstruction error for the
475   autoencoder would increase for the non-normal data scenario when the sample size was small. This result
476   showed that although the computational resources for the neural network were generally more expensive,
477   besides the simple encoder, the reconstruction error for the autoencoder was uniformly smaller across
478   three different communality conditions as well as three different non-normality conditions.

479
480

Conclusions

482   A common practical question in conducting factor analyses and PCAs is how many subjects are
483   sufficient to obtain a reliable estimate. There are many rules of thumb proposed suggesting a certain
484   absolute sample size, such as a minimum of 250 or 500. Some suggested a size-to-time ratio to be as high
485   as 10 times as many subjects as variables. There is no universally applicable answer, but the answer
486   depends upon the clarity of the structure being examined, i.e., the communality of the variables. If there is
487   well-structured communality among the variables, the size of the sample required goes down. As the
488   number of items for a factor increase, the sample size needed decreases. That is, if the number of
489   variables is high relative to the number of factors (e.g.,15:3), and the communality is high, then sample
490   sizes as small as 60~100 are adequate. In FA models, more subjects are always better, but what is more
491   important is to have good markers for each factor (high communality) as well as many markers (a high
492   item-to-factor ratio) than it is to increase the number of subjects. Unfortunately, although it will never be
493   wrong advice to have many markers of high communality, if using an autoencoder to analyze the
494   structure of items rather than tests, the communality requirement will tend to be low. In cases where
495   communality is low and data are not linearly distributed, increasing the number of subjects is advised.
496   Based on this simulation study, the autoencoder tended to perform relatively better when the neurons of
497   the bottleneck, k, were small, which means that the same reconstruction accuracy of the original scale
498   could be achieved with fewer components and hence a smaller dataset. This is important when dealing
499   with many items or variables in a questionnaire. For all sample sizes being considered, the tied-weight
500   encoder had relatively small reconstruction errors. Based on this Monte Carlo simulation, we
501   demonstrated that it is feasible for an autoencoder to be used in psychometric research to establish the
502   construct validity with a small sample size.
503   .
504

505 **Acknowledgements**

506   N/A.

507

References

508 **References**

509 Abdi, H. and L.J. Williams, Principal component analysis. *Wiley interdisciplinary*
510 *reviews:computational statistics*, 2010. **2**(4): p. 433-459.

511

512 Barton, K., Cattell, R. B., & Curran, J. Psychological states: Their definition   through P-
513 technique and differential R (dR) technique factor analysis. *Journal of Behavioural Science*,
514 (1973)

515

516 Bagozzi, R.P., Y. Yi, and L.W. Phillips, Assessing construct validity in organizational research.
517 *Administrative science quarterly*, 1991: p. 421-458.

518

519 Bourlard, H. and Y. Kamp, Auto-association by multilayer perceptrons and singular value
520 decomposition. *Biological cybernetics*, 1988. **59**(4-5): p. 291-294.

521

522 Byrd, J. K. (2008). *A Monte Carlo investigation of three different estimation methods in*
523 *multilevel structural equation modeling under conditions of data nonnormality and varied*
524 *sample sizes*. Texas A&M University.

525

526 Campbell, D.T. and D.W. Fiske, Convergent and discriminant validation by the multitrait
527 multimethod matrix. *Psychological bulletin*, 1959. 56(2): p. 81.

528

529 Cronbach, L.J. and Meehl, P.E, Construct validity in psychological tests. *Psychological bulletin*,
530 1955. **52**(4): p. 281.

531

532 Chatfield, C. and A. Collins, *Introduction to multivariate analysis*. Vol. 1. 1981: CRC Press.

533

534 Chollet, F. (2016). Building autoencoders in keras. *The Keras Blog*, 14.

535

536 Fan, X., Felsovalyi, A., Sivo, S. A., & Keenan, S. C. *SAS for Monte Carlo studies. Proc. SAS*
537 *Inst.*, 2002, Cary, 87-89.

538

539 Fontes, F., Gonçalves, M., Maia, S., Pereira, S., Severo, M., & Lunet, N. *Reliability and validity*
540 of the Pittsburgh Sleep Quality Index in breast cancer patients. *Supportive Care in Cancer*, 2017.
541 **25**(10): p. 3059-3066.

542

543 Fleishman, A.I., A method for simulating non-normal distributions. *Psychometrika*, 1978. **43**(4):
544 p. 521-532.

545

546 Forcino, Frank L. "Multivariate assessment of the required sample size for community
547 paleoecological research." *Palaeogeography, Palaeoclimatology, Palaeoecology* 315 (2012):
548 134-141.

549

550   Hatcher, L., & O'Rourke, N. A *step-by-step approach to using SAS for factor analysis and*
551   *structural equation modeling.* Sas Institute. (2013).

552

553   Hamzah, M.I., A.K. Othman, and F. Hassan. Transcending Beyond the Organizational Context:
554   Validation of the Individual Market Orientation Construct. in *ECIC2016-Proceedings of the 8th*
555   *European Conference on Intellectual Capital: ECIC2016*. 2016. Academic Conferences and
556   publishing limited.

557

558   Hinton, G.E. and R.S. Zemel. Autoencoders, minimum description length and Helmholtz free
559   energy. in *Advances in neural information processing systems*. 1994.

560

561   Hutcheson, G. D., and Nick Sofroniou. *The multivariate social scientist: Introductory statistics*
562   *using generalized linear models*. Sage, 1999.

563

564   Jackson, D.N., *A sequential system for personality scale development*, in *Current topics in*
565   *clinical and community psychology*. 1970, Elsevier. p. 61-96.

566

567   Kaiser, H.F. and K. Dickman, Sample and population score matrices and sample correlation
568   matrices from an arbitrary population correlation matrix. *Psychometrika,* 1962. **27**(2): p. 179-
569   182.

570

571   Kramer, M.A., Nonlinear principal component analysis using autoassociative neural networks.
572   *AIChE journal,* 1991. **37**(2): p. 233-243.

573

574   Kim, D. and Kim, S.-K., Comparing patterns of component loadings: Principal Component
575   Analysis (PCA) versus Independent Component Analysis (ICA) in analyzing multivariate non-
576   normal data. *Behavior research methods*, 2012. **44**(4): p. 1239-1243.

577

578   Ladjal, S., A. Newson, and C.-H. Pham, A pca-like autoencoder. *arXiv preprint*
579   *arXiv*:1904.01277, 2019.

580

581   Le, Q.V. Building high-level features using large scale unsupervised learning. in *2013 IEEE*
582   *international conference on acoustics, speech and signal processing*. 2013. IEEE.

583

584   MacCallum, R. C., Widaman, K. F., Preacher, K. J., & Hong, S. Sample size in factor analysis:
585   The role of model error. *Multivariate behavioral research*, 2001, 36(4), 611-637.

586

587   Mamdoohi, A., Axhausen, K. W., Mahpour, A., Rashidi, T. H., & Saffarzadeh, M. (2016). Are
588   there latent effects in shopping destination choice?: Survey methods and response behavior. In
589   16th Swiss Transport Research Conference (STRC 2016). *Swiss Transport Research Conference*
590   *(STRC)*.

591

592   Manjarrés-Martínez, L. M., Gutierrez-Estrada, J. C., Hernando, J. A., & Soriguer, M. C. "The
593   performance of three ordination methods applied to demersal fish data sets: stability and
594   interpretability." *Fisheries Management and Ecology* 19.3 (2012): 200-213.

595

596   Meng, L., Ding, S., & Xue, Y. Research on denoising sparse autoencoder. *International Journal*
597   *of Machine Learning and Cybernetics,* 2017, 8(5), 1719-1729.

598

599   Mundfrom, Daniel J., Dale G. Shaw, and Tian Lu Ke. "Minimum sample size recommendations
600   for conducting factor analyses." *International Journal of Testing* 5.2 (2005): 159-168.

601

602   Mohammadbeigi, A., Mohammadsalehi, N., & Aligol, M. Validity and reliability of the
603   instruments and types of measurments in health applied researches. *Journal of rafsanjan*
604   *university of medical sciences*, 2015, 13(12), 1153-1170.

605

606   Park, S. H., Ha, J. P, and Mahony, D. Development and validation of a measure of sport fans'
607   specific curiosity. *Journal of Sport Management*, 2014. 28: 621-632

608

609   Parker, S. K., Bindl, U. K., & Strauss, K. Making things happen: A model of proactive
610   motivation. *Journal of management*, 2010, 36(4), 827-856.

611

612   Ramberg, J. S., Dudewicz, E. J., Tadikamalla, P. R., & Mykytka, E. F. A probability distribution
613   and its uses in fitting data. *Technometrics*, 1979, 21(2), 201-214.

614

615   Schmeiser, B.W. and S.J. Deutsch, A versatile four parameter family of probability distributions
616   suitable for simulation. *AIIE Transactions,* 1977. **9**(2): p. 176-182.

617

618   Seyfioğlu, M.S. and S.Z. Gürbüz, Deep neural network initialization methods for micro-Doppler
619   classification with low training sample support. *IEEE Geoscience and Remote Sensing Letters*,
620   2017. **14**(12): p. 2462-2466.

621

622   Sherman, R.E., Evidence of the construct validity and potential application of a multivariate
623   approach to alcoholism diagnosis. *The American journal of drug and alcohol abuse,* 1986. **12**(1-
624   2): p. 131-146.

625

626   Shlens, J. A tutorial on principal component analysis. *arXiv preprint arXiv*, 2014, 1404.1100.

627

628   Stangor, C., Research methods for the behavioral sciences. 2014: Nelson Education.

629

630 Vale, C. D., & Maurelli, V. A. Simulating multivariate nonnormal distributions. *Psychometrika*,
631 1983, 48(3), 465-471.
632
633 Wang, Y., Statistical Techniques for Network Security: Modern Statistically-Based Intrusion
634 Detection and Protection*: Modern Statistically-Based Intrusion Detection and Protection*. 2008:
635 Igi Global.
636
637 Wang, W., Huang, Y., Wang, Y., & Wang, L. Generalized autoencoder: A neural network
638 framework for dimensionality reduction. In: *Proceedings of the IEEE conference on computer
639 vision and pattern recognition workshops*. 2014. p. 490-497.
640
641 Wang, J., H. He, and D.V. Prokhorov, A folded neural network autoencoder for dimensionality
642 reduction. *Procedia Computer Science*, 2012. **13**: p. 120-127.
643
644 Wang, W., Yang, D., Chen, F., Pang, Y., Huang, S., & Ge, Y., Clustering With Orthogonal
645 AutoEncoder. *IEEE Access*, 2019. **7**: p. 62421-62432.
646
647 Wicklin, R., *Simulating data with SAS*. 2013: SAS Institute.
648
649 Yeung, K.Y. and W.L. Ruzzo, Principal component analysis for clustering gene expression data.
650 *Bioinformatics, 2001*. **17**(9): p. 763-774.
651
652 Young, F.W., P.M. Valero-Mora, and M. Friendly, *Visual statistics: seeing data with dynamic
653 interactive graphics*. Vol. 914. 2011: John Wiley & Sons.
654
655 Yoon, J.W., Chung, C. Y., Lee, D. W., Shin, J. H., Cho, J. W., Kim, C. K., ... & Kim, J. O,
656 Reliability and Validity of the Mental Health Questionnaire for Adult. *Journal of Korean
657 Neuropsychiatric Association*, 2019. 58(1): p. 64-73.
658
659 Yong, A. G., & Pearce, S. A beginner's guide to factor analysis: Focusing on exploratory factor
660 analysis. *Tutorials in quantitative methods for psychology*, 2013, *9*(2), 79-94.

Items on the Curiosity questionnaire

1   Table 1. Items on the Curiosity questionnaire

Item	Description
1	I enjoy collecting and calculating statistics for my favorite basketball team.
2	I often imagine how my favorite basketball team is playing to defeat their opponent.
3	I enjoy exploring my favorite basketball stadiums or facilities.
4	Watching basketball games with my friends is joyful.
5	I enjoy reading articles about basketball players, teams, events, and games.
6	I am interested in learning how much it costs to build a brand-new basketball stadium.
7	When I miss a game, I often look for information on television, the internet, or newspaper to catch the game results.
8	I am interested in learning how large a basketball court is.
9	I enjoy probing deeply into basketball.
10	I am eager to learn more about basketball.
11	I enjoy any movement that occurs during a basketball game.

Parameters for generating non-normal data

1   Table 2. Parameters for generating non-normal data

Scenario no.	Mean	Standard deviation	Skewness	Kurtosis
1	0	1	0	0
2	0	1	1	3
3	0	1	2	20

Performance metrics for three communality conditions

1   Table 3. Performance metrics for three communality conditions

Metric	Algorithm	High Communality		Wide Communality		Low Communality	
		Mean	SD	Mean	SD	Mean	SD
MSE	Simple encoder	0.343	0.175	0.382	0.203	0.458	0.146
	Tied encoder	0.034	0.009	0.034	0.010	0.034	0.009
	PCA	0.230	0.160	0.254	0.196	0.337	0.148
	Deep autoencoder	0.032	0.019	0.036	0.022	0.044	0.017
	Independent autoencoder	0.036	0.021	0.040	0.023	0.048	0.018
MAE	Simple encoder	0.444	0.112	0.460	0.128	0.523	0.086
	Tied encoder	0.126	0.018	0.126	0.018	0.127	0.017
	PCA	0.357	0.116	0.356	0.144	0.443	0.101
	Deep autoencoder	0.107	0.030	0.110	0.034	0.128	0.024
	Independent autoencoder	0.113	0.031	0.117	0.035	0.133	0.025
NED	Simple encoder	2.288	0.166	2.277	0.179	2.226	0.144
	Tied encoder	1.053	0.076	1.055	0.074	1.060	0.073
	PCA	4.730	0.303	4.745	0.341	4.634	0.291
	Deep autoencoder	1.146	0.086	1.144	0.094	1.114	0.079
	Independent autoencoder	1.138	0.099	1.132	0.104	1.111	0.085

2   MSE, mean squared error; PCA principal component analysis; MAE, mean absolute error; NED, non-

3   Euclidian distance; SD, standard deviation.

Performance metrics for three normality conditions

1   Table 4. Performance metrics for three normality conditions

Metric	Algorithm	Normal		Slightly un-normal		Un-normal	
		Mean	SD	Mean	SD	Mean	SD
MSE	Simple encoder	0.397	0.182	0.394	0.181	0.392	0.185
	Tied encoder	0.034	0.009	0.034	0.009	0.034	0.010
	PCA	0.277	0.174	0.275	0.175	0.269	0.177
	Deep autoencoder	0.038	0.020	0.038	0.020	0.037	0.019
	Independent autoencoder	0.042	0.021	0.041	0.021	0.041	0.021
MAE	Simple encoder	0.484	0.117	0.473	0.114	0.471	0.115
	Tied encoder	0.128	0.018	0.126	0.018	0.125	0.017
	PCA	0.393	0.130	0.382	0.128	0.380	0.127
	Deep autoencoder	0.117	0.032	0.114	0.031	0.114	0.030
	Independent autoencoder	0.123	0.033	0.121	0.032	0.120	0.031
NED	Simple encoder	2.305	0.158	2.263	0.164	2.223	0.166
	Tied encoder	1.075	0.073	1.053	0.074	1.041	0.072
	PCA	4.787	0.299	4.699	0.314	4.622	0.314
	Deep autoencoder	1.155	0.086	1.134	0.087	1.116	0.087
	Independent autoencoder	1.147	0.095	1.126	0.096	1.108	0.096

2   MSE, mean squared error; MAE, mean absolute error; NED, non-Euclidian distance; PCA, principal

3   component analysis; SD, standard deviation.

Absolute values of bottleneck weights on the Curiosity dataset

1   Table 5. Absolute values of bottleneck weights on the Curiosity dataset

2   a.   Bottleneck weights estimates from the population

Learning Motivation	Social Interaction	Facility	
0.329	0.029	0.237	I enjoy collecting and calculating statistics of my favorite basketball team.
0.474	0.178	0.293	I enjoy reading articles about basketball players, teams, events, and games.
0.335	0.158	0.261	I am eager to learn more about basketball.
0.68	0.003	0.227	I enjoy any movement that occurs during a basketball game.
0.385	0.433	0.415	Watching basketball games with my friends is joyful.
0.351	0.575	0.114	I enjoy probing deeply into basketball.
0.314	0.662	0.039	I often imagine how my favorite basketball team is playing to defeat their opponent.
0.125	0.519	0.469	I enjoy exploring my favorite basketball stadiums or facilities.
0.011	0.385	0.023	I am interested in learning how much it costs to build a brand new basketball stadium.
0.104	0.19	0.581	I am interested in learning how large a basketball court is.
0.099	0.127	0.231	When I miss a game, I often look for information on television, the internet, or newspaper to catch the game results.

5   b. Bottleneck weights estimates from sample of n=100

Learning Motivation	Social Interaction	Facility	
0.431	0.090	0.138	I enjoy collecting and calculating statistics of my favorite basketball team.
0.532	0.067	0.299	I enjoy reading articles about basketball players, teams, events, and games.
0.322	0.239	0.312	I am eager to learn more about basketball.
0.541	0.051	0.170	I enjoy any movement that occurs during a basketball game.
0.308	0.767	0.124	Watching basketball games with my friends is joyful.
0.089	0.440	0.100	I enjoy probing deeply into basketball.
0.355	0.685	0.077	I often imagine how my favorite basketball team is playing to defeat their opponent.
0.431	0.175	0.431	I enjoy exploring my favorite basketball stadiums or facilities.
0.117	0.203	0.615	I am interested in learning how much it costs to build a brand new basketball stadium.
0.126	0.344	0.383	I am interested in learning how large a basketball court is.
0.031	0.078	0.224	When I miss a game, I often look for information on television, the internet, or newspaper to catch the game results.

Figure 1

Flow chart for the Monte Carlo (MCMC) simulations.
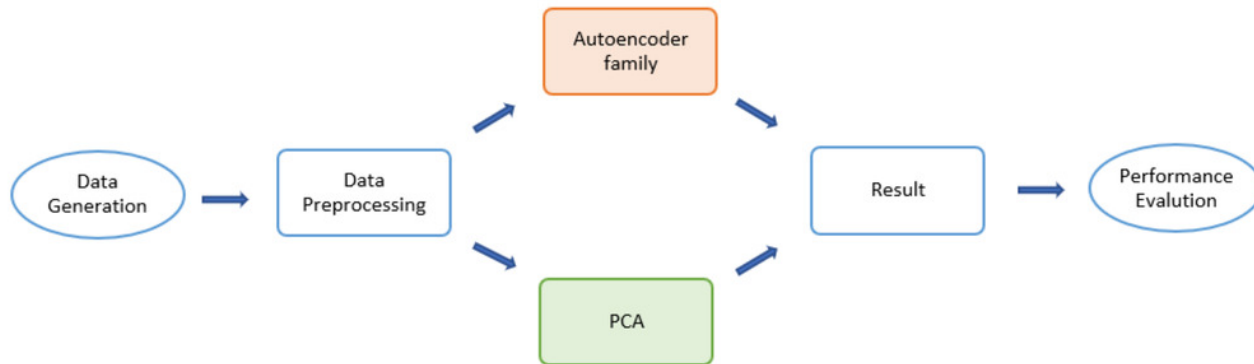
Figure 2

Autoencoder architecture
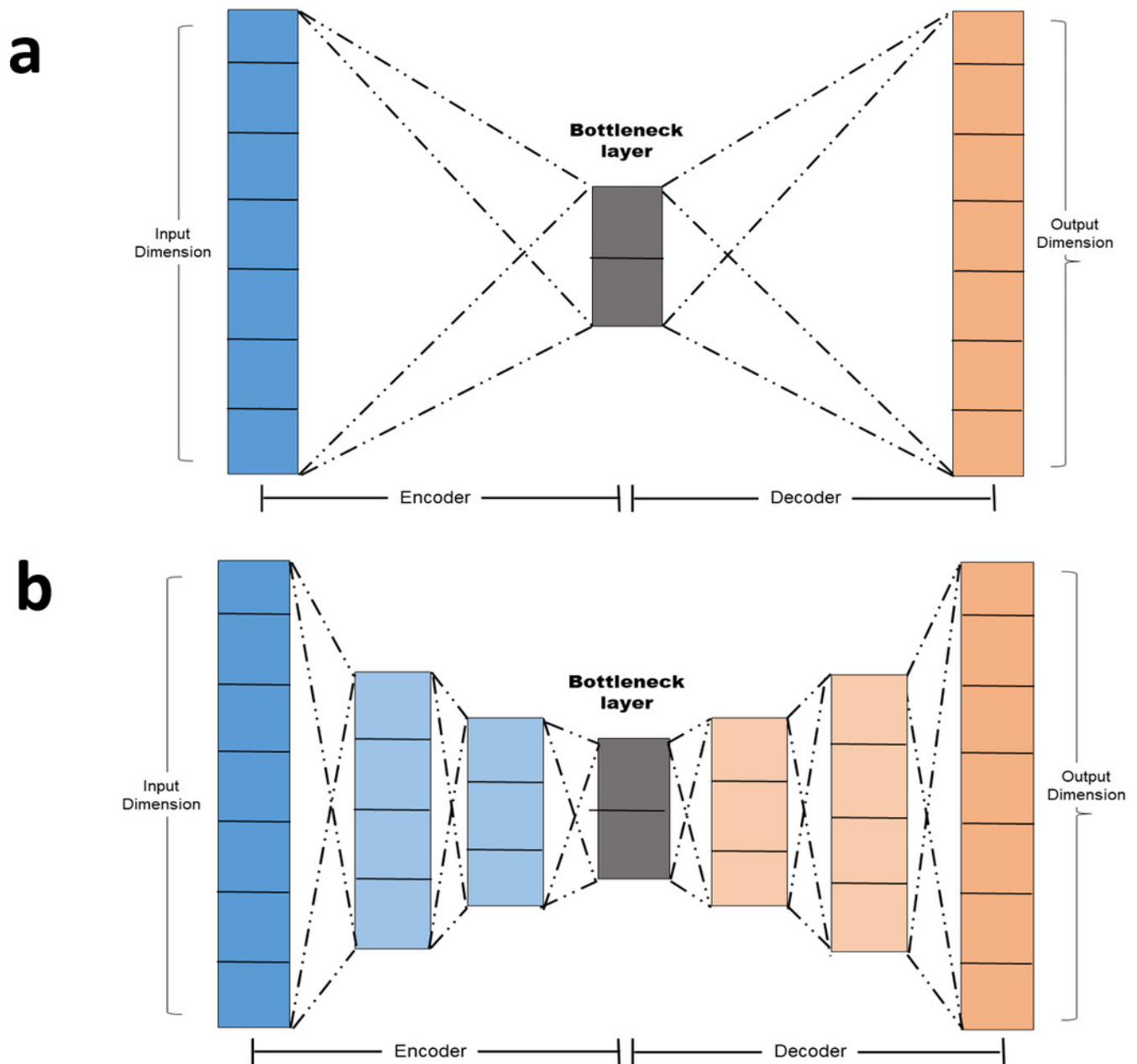
Figure 3

Interaction of communality and non-normality condition on the mean squared error (MSE) for the simple autoencoder.
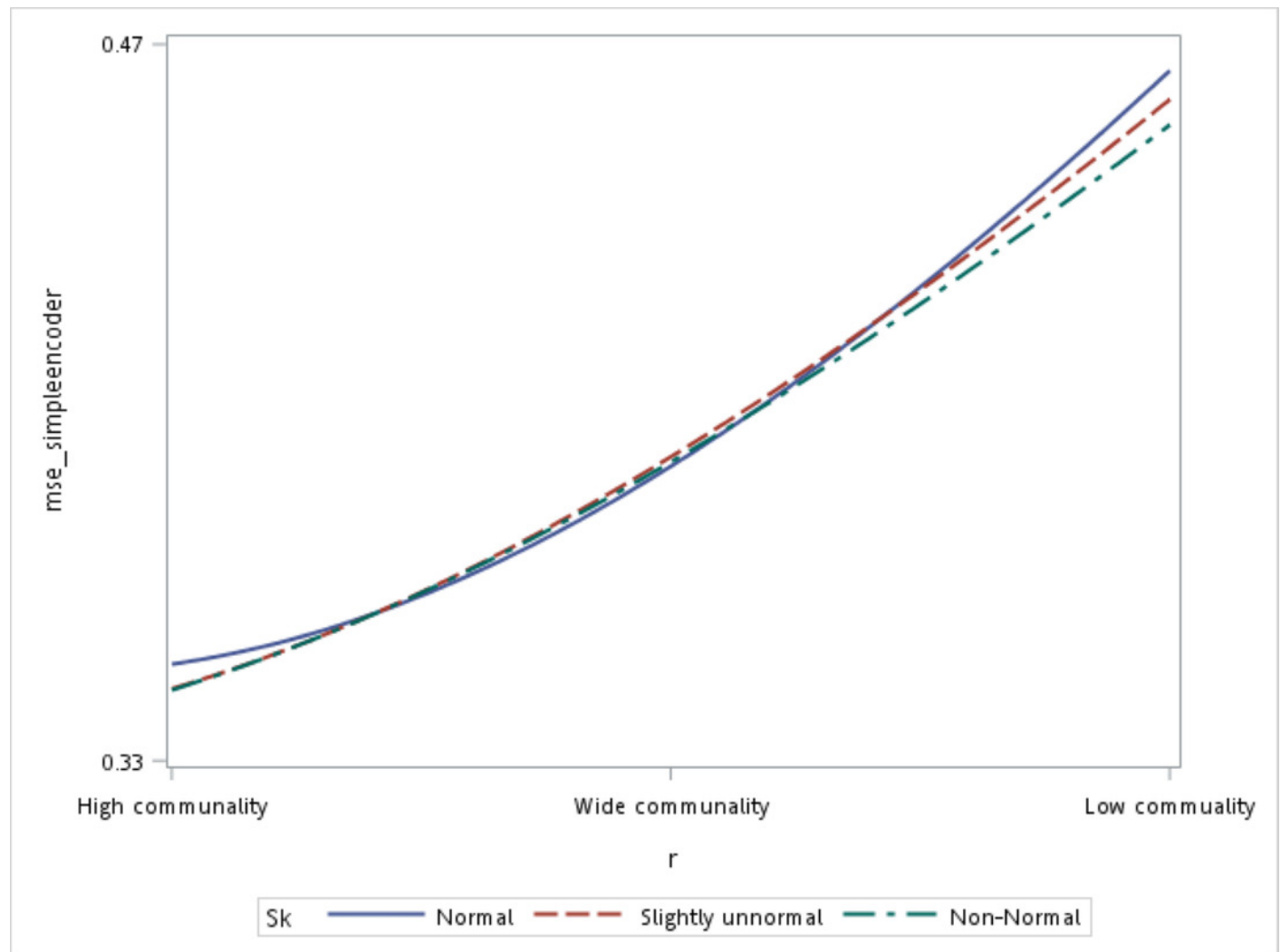
Figure 4

Reconstruction error for the autoencoder family under different sample sizes.
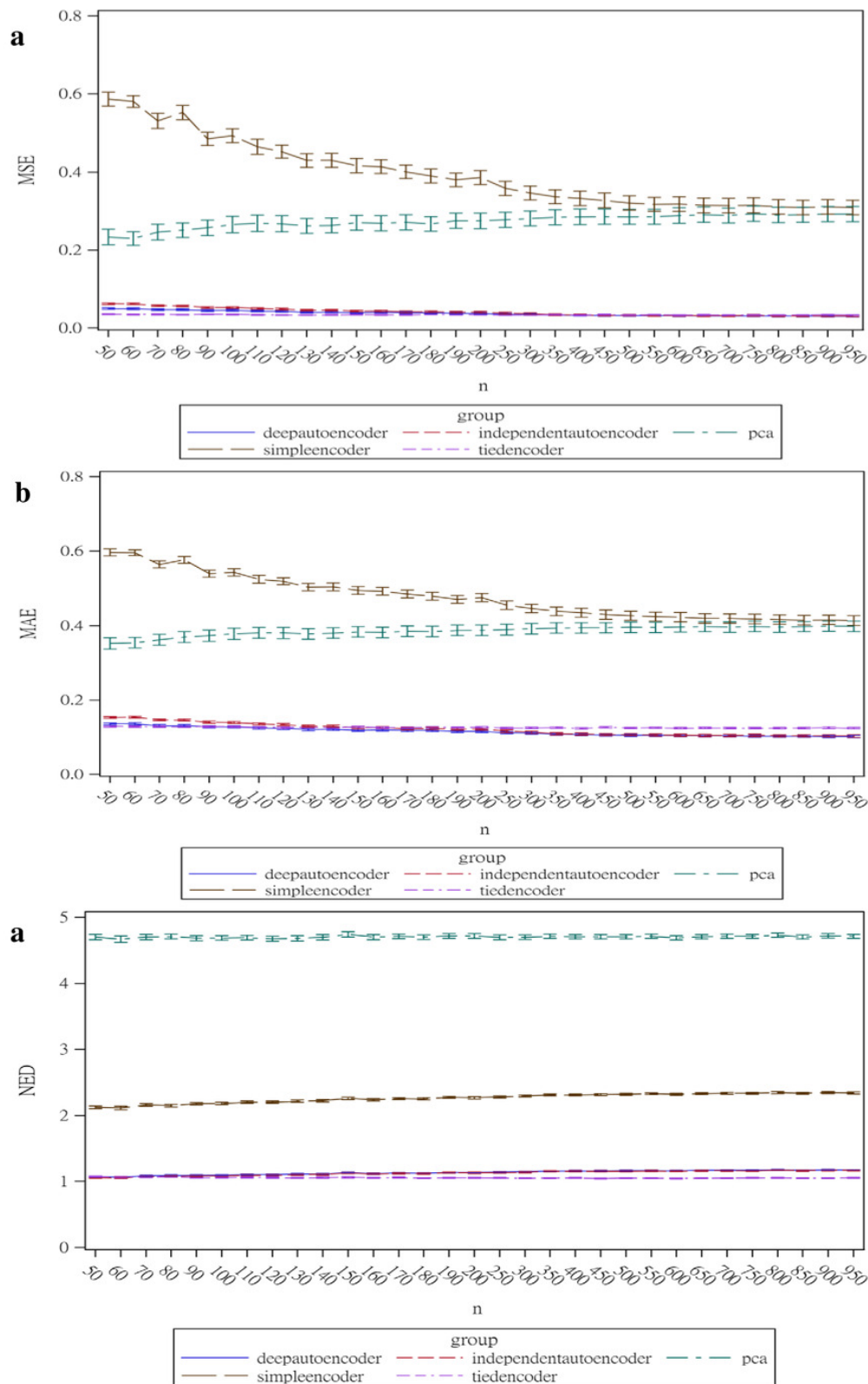
Figure 5

Mean square errors (MSEs) for different combinations of communality and normality.
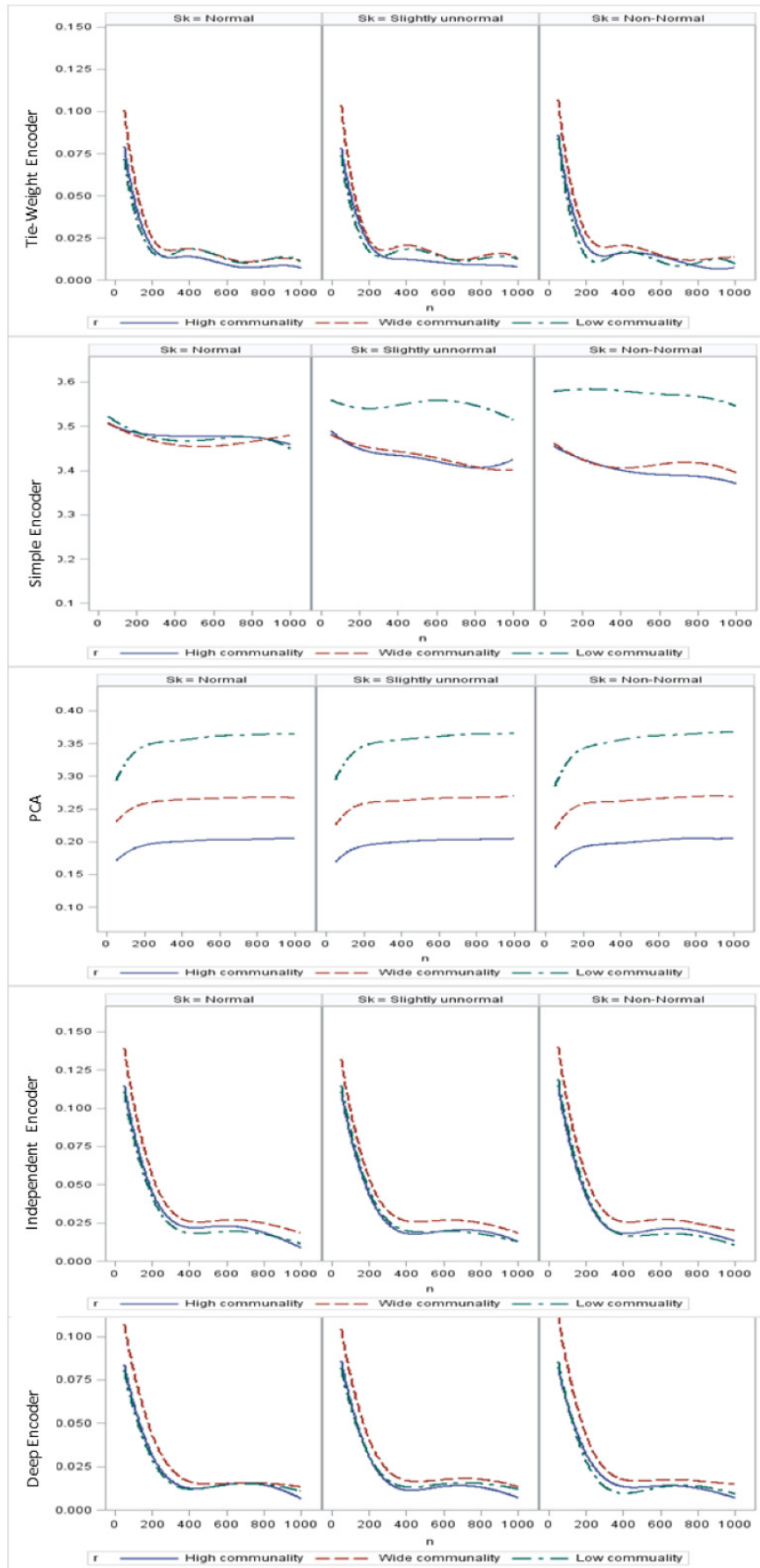
Figure 6

Mean absolute errors (MAEs) for the principal component analysis (PCA) and deep autoencoder
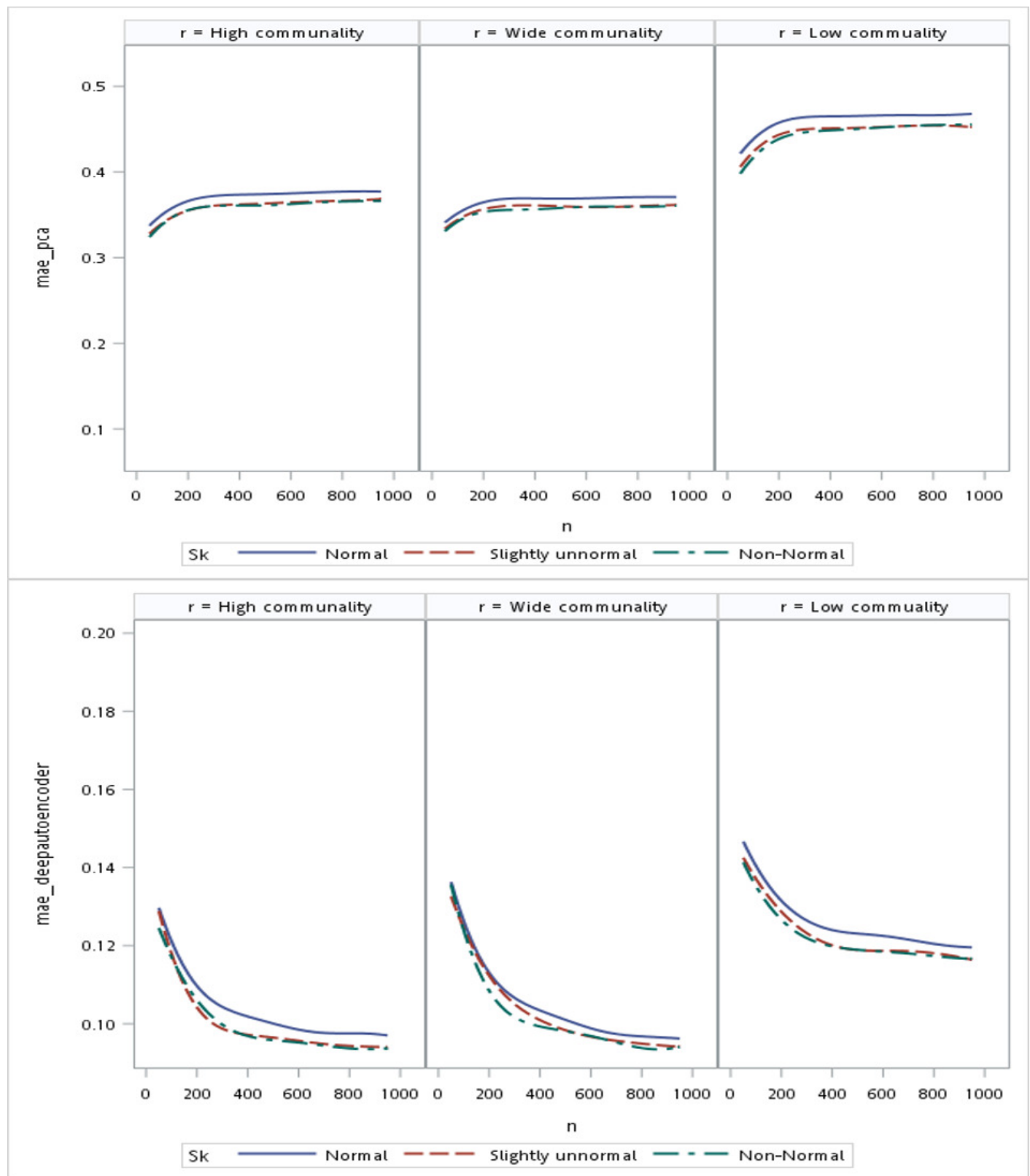
Figure 7

Mean squared errors (MSEs) of the autoencoder and principal component analysis (PCA) under various extracted components
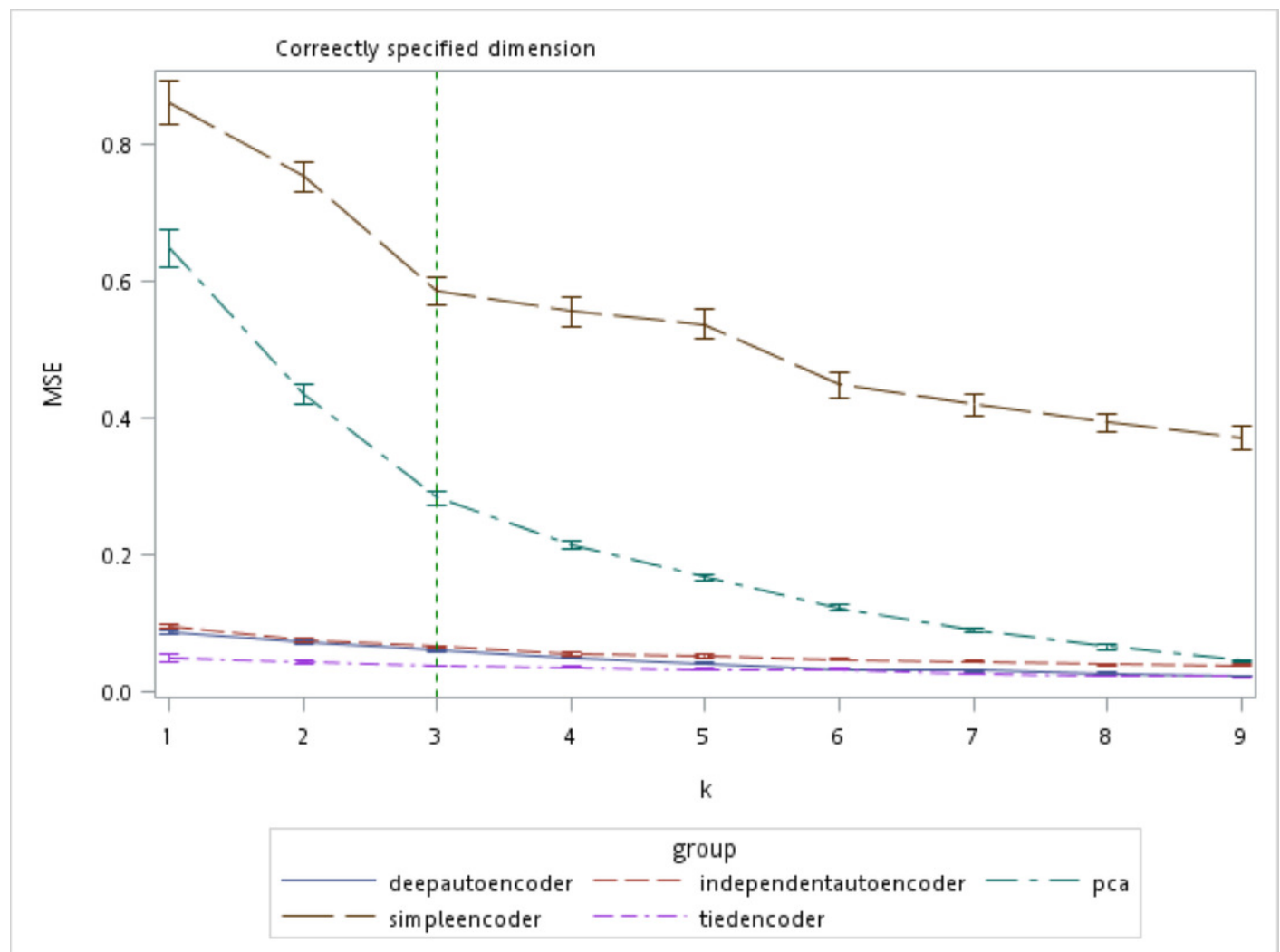
Figure 8

Figure 8 tSNE visualizations for PCA and Autoencoder