



Probabilistic Modeling of Heavy Machinery Using Machine Learning

Tyler Cody

University of Virginia, Charlottesville, Virginia

Lauren McIntyre

Glenn Research Center, Cleveland, Ohio

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., “quick-release” reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.
- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at <http://www.sti.nasa.gov>
- E-mail your question to help@sti.nasa.gov
- Fax your question to the NASA STI Information Desk at 757-864-6500
- Telephone the NASA STI Information Desk at 757-864-9658
- Write to:
NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199



Probabilistic Modeling of Heavy Machinery Using Machine Learning

Tyler Cody

University of Virginia, Charlottesville, Virginia

Lauren McIntyre

Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

Acknowledgments

I would like to thank John DeGreen, Rob Norman, Debashis Sadhukhan, Lauren McIntyre, Calvin Robinson, and Herb Schilling for their mentorship and support. I would like to thank Brooke Weborg for her help in processing data. I would also like to thank all the system experts of the central air service who taught me about the system and thoughtfully answered my questions. Lastly, I want to thank NASA and Glenn Research Center for the opportunity to research interesting and exciting problems.

Level of Review: This material has been technically reviewed by technical management.

Available from

NASA STI Program
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681-2199

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is available in electronic form at <http://www.sti.nasa.gov/> and <http://ntrs.nasa.gov/>

Probabilistic Modeling of Heavy Machinery Using Machine Learning

Tyler Cody*
University of Virginia
Charlottesville, Virginia 22904

Lauren McIntyre
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Summary

NASA Glenn Research Center's Systems Management and Operations Branch seeks to leverage its extensive instrumentation and historical data with machine learning to increase system efficiencies. The ultimate goal of this effort is to probabilistically model the behavior of Glenn's central air service compressors for optimal decision making and planning. This project is a first step in that direction. We propose a multi-model approach that uses high-dimensional models to ask simple questions about complex dependent structures, and low-dimensional models to ask complex questions about simple dependent structures. While the low-dimensional models make strong assumptions, they can be visualized and they can be insightful. We show good fits for univariate models of compressor sensors and preliminary work on high-dimensional multivariate models.

Introduction

NASA Glenn Research Center's central air service consists of a large network of compressors and exhausters that facilitate the distribution of air to various testing facilities. The value of the operations and the consequences of system failure make system maintenance a priority. The size, scale, and uniqueness of the system's machinery make maintenance costly as well. The machinery and the system at large have extensive instrumentation that is used for monitoring and analytics. The central air service system's scale of operations and instrumentation necessitate some form of automation in the data analysis process, if only to direct the attention of system operators who can manually review specific occurrences or trends. This project is a first step in applying machine learning methods to probabilistically model system behavior toward these ends.

Machine learning is a broad term often considered to be a set of methods for statistical inference. Our goal is to use machine learning to identify when system behavior is trending away from baseline behavior. Our approach is most similar to unsupervised learning; however, this label can be constraining, and we prefer to consider our work to be using machine learning to probabilistically model system behavior. If we can learn accurate, empirical, probabilistic models of system behavior, we can use them to make empirically based decisions and plans.

*Summer Intern in Lewis' Educational and Research Collaborative Internship Project (LERCIP), Scientific Computing and Visualization. Graduate student at University of Virginia.

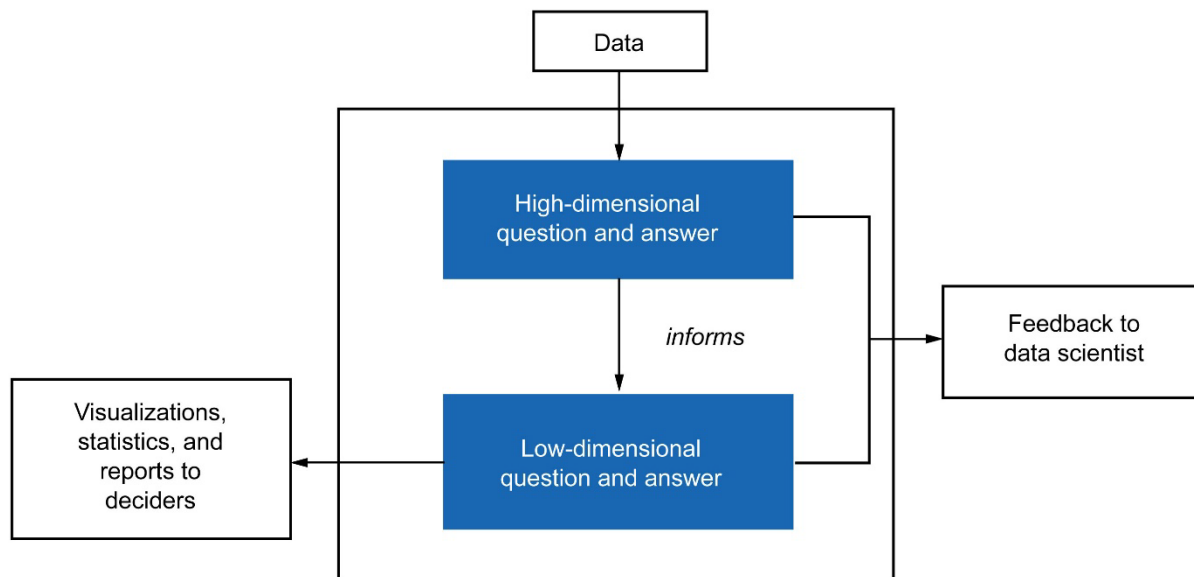


Figure 1.—Multi-model approach.

A fundamental challenge in the data-driven analysis of large-scale systems is our ability to fit high-dimensional models that provide more than high-level insight. When modeling in high dimensions, we are often forced to trade off the specificity with which we can investigate models to ask questions about our system. To address this challenge, we propose a multi-model approach that uses high-dimensional models to ask simple questions about complex dependent structures, and low-dimensional models to ask complex questions about simple dependent structures. A rough model for this approach is shown in Figure 1.

The idea represented in Figure 1 is to use high-dimensional models, which can consider system-level trends but are difficult to interpret, to direct questions and analysis of low-dimensional models, which are interpretable by deciders through visualizations, statistics, and reports. If the deciders are system experts, then the information loss that occurs by modeling in a lower dimension, as opposed to a higher dimension, can be offset by the expert knowledge that deciders bring to the analysis process. The pragmatic motivation for such an approach is that the best data-driven, high-dimensional distribution modeling techniques rely on deep learning, a notoriously difficult-to-interpret set of methods. By using low-dimensional models to explain findings and conduct analysis, this multi-model approach can interface directly with system operators, and the findings of the deep learning model are validated by interpretable, low-dimensional models.

Methods

In this project, we fit both low-dimensional and high-dimensional models to compressor data. The related methods are described here, with a discussion of the results in the following section. We did not explicitly study methods for realizing an analysis system like that shown in Figure 1; however, a discussion of how our modeling techniques can be used to realize such a system is included in the concluding section.

Low-Dimensional Modeling

For low-dimensional modeling, we fit closed-form, parametric univariate distributions to the sensor data. We treat a sequence of sensor readings over the course of a run as a set of events and model the probability of realizing an event. Closed-form univariate models are interpretable and easy to measure and manipulate. To fit a univariate model, we first construct an empirical distribution function by arranging the events in ascending order and assigning each a plotting position corresponding to the probability that the event or an event of lesser magnitude is realized. We then fit a variety of candidate parametric distributions and judge the goodness of fit using the maximum absolute difference between the candidate model and the empirical distribution.

While such an approach only considers the behavior of individual sensors, it is the foundation for constructing joint distributions using copula methods (Ref. 1) or numerically approximating joint distributions using Markov chain Monte Carlo methods (Ref. 2). Both of these approaches use lower dimensional models to fit higher dimensional models. They are well suited for modeling the distribution of small sets of sensors but are impractical methods for fitting the high-dimensional models we are interested in, which have hundreds or even thousands of sensors.

High-Dimensional Modeling

Deep learning methods are the current state of the art for data-driven distribution modeling in high dimensions (Refs. 3 and 4). Deep learning methods are parameter-heavy algorithms that use large amounts of data to learn a particular mapping. Their primary benefit is in automatically finding useful features to describe data. Instead of using expert knowledge to craft features for modeling data, deep learning algorithms iteratively search for transformations on the input data that are useful for solving the learning problem. The exact nature of this search and these transformations varies with the deep learning network architecture and learning task.

The scale of Glenn’s central air service system may necessitate representation-learning techniques such as deep learning. Manual feature selection may be insufficient to deal with the number of sensors and the complexity of the interactions between subsystems. Fortunately, deep learning methods are well suited for the central air service system because of the volume of data the system produces. Thus, deep learning was selected as the starting point for high-dimensional modeling because of its relevance in the technical literature and its relevance to the fundamental nature of modeling the dynamics of large-scale systems like the central air service.

The two main approaches to distribution modeling using deep learning are variational autoencoders (VAEs) (Ref. 5) and generative adversarial networks (GANs) (Ref. 6). The two approaches solve similar problems, but they are fundamentally different. GANs consist of a generator that produces synthetic data and approximates the joint distribution of interest, and a discriminator that tries to identify whether a given sample is real or synthetic. The generator tries to “trick” the discriminator by generating realistic data, and the two iteratively update their parameters to improve their performance as they play this game. In doing so, GANs implicitly learn a joint distribution by learning a generator that can be sampled to produce realistic synthetic samples, and a discriminator that can be queried to approximate the probability that a given sample comes from the joint distribution of interest. VAEs are used to learn an explicit, closed-form latent distribution in a lower dimension than the input dimension. VAEs consist of an encoder that maps the input data into a latent distribution and a decoder that maps the latent distribution back to the original input space. The VAE is trained to learn a well-fit latent distribution that minimizes the difference between the input and the reconstructed input—that is, the input after it has been passed through the encoder and decoder.

We are interested in studying how system behavior varies in the central air service. GANs are inferior to VAEs in accomplishing this task because (1) comparing generators still requires high-dimensional modeling of the output of the generators and (2) although the discriminator can provide a probability of whether distributional changes have occurred in system behavior, it does not provide direct insight into how the distribution has changed. VAEs, in contrast, learn a mapping to an explicit, low-dimensional probability distribution that can be studied in detail. Differences between behavior in the latent space can be decoded into the original input space. Furthermore, learning an explicit distribution aligns directly with our high-level goal of learning an empirical probabilistic model of the system for decision making and planning.

We learn the canonical VAE, which learns a latent multivariate Gaussian distribution with independent marginals. A VAE provides a mapping to a conditional distribution function: the latent random variable conditioned on the input. Thus, for a single sample of system behavior, we obtain many conditional distributions; we obtain one distribution for each instance. To reduce this set of distributions to a cardinality more amenable for our purposes, we cluster over the means of the conditional multivariate Gaussians and average the distributions of each cluster to obtain a single distribution function for each cluster. Because we are learning a multivariate Gaussian distribution with independent marginals, it can be shown that the average of the distributions is a multivariate Gaussian distribution with independent marginals as well. In other words, we retain a closed-form parametric model of system behavior after clustering.

Results

We tested our proposed methods on a single casing of a single compressor. We used 12 sensors, which provided a mix of temperature, vibration, and voltage readings. The data was minimally processed, in part because of the time constraints.

Low-Dimensional Modeling

We successfully modeled the sensors with closed-form univariate distributions. Most models of sensors obtained a tight fit. The benefit of doing this is that we can compare runs by using measures between the distributions of their respective sensor readings. Furthermore, if we wanted to compare runs using summary statistics such as quantiles instead of the full distribution, we can obtain these statistics from a well-fit distribution. We do not have to assume normality to obtain statistical measures of the distribution. This is particularly useful considering that of all the sensors we studied, none were Gaussian and all were skewed. Examples of a fitted temperature distribution and a vibration distribution plotted against raw data are shown in Figure 2 and Figure 3, respectively.

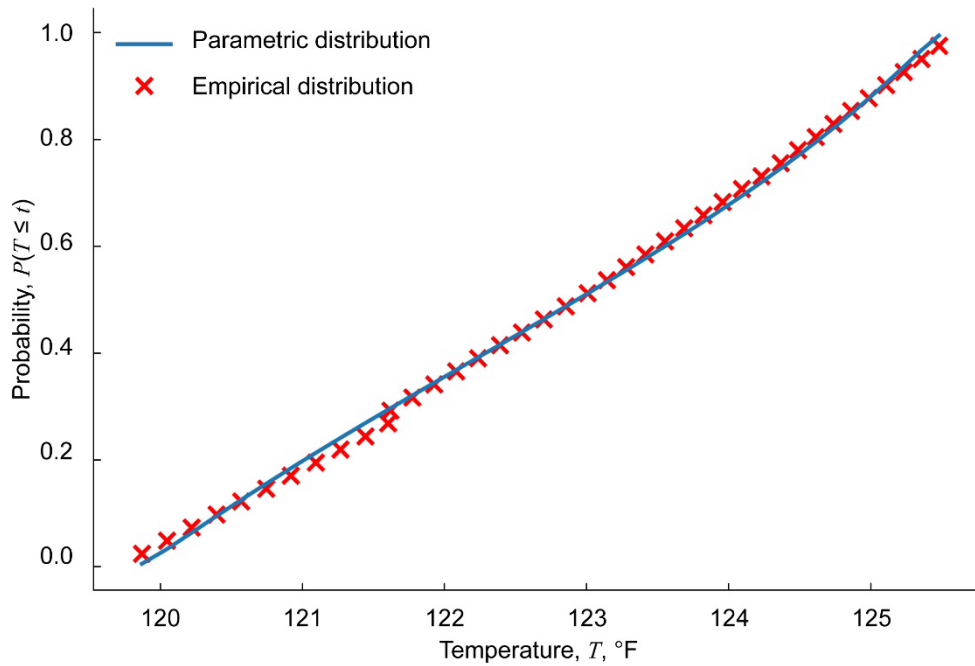


Figure 2.—Univariate temperature example. Specific temperature, t .

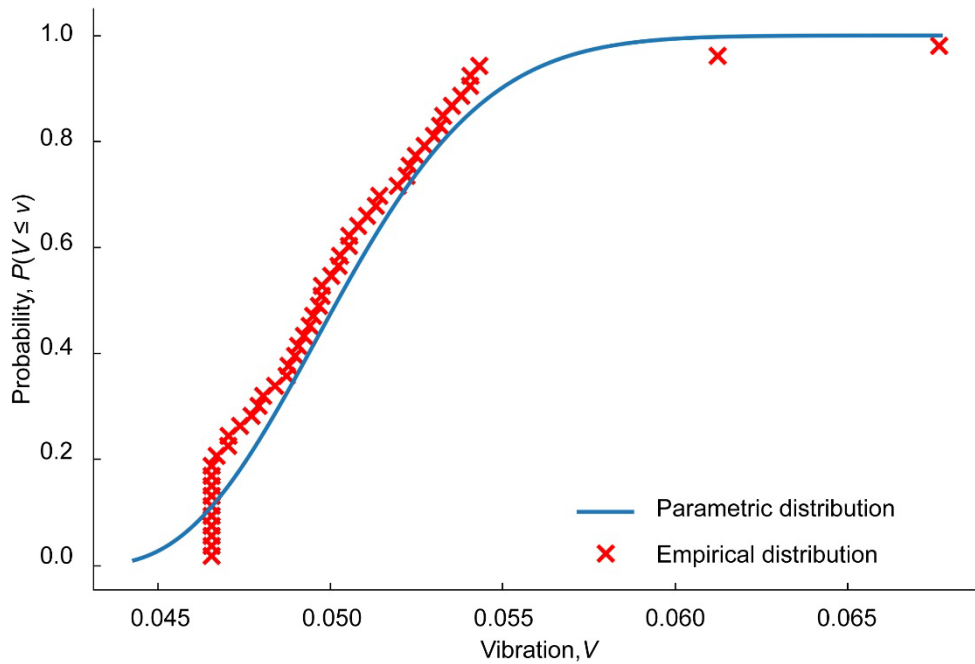


Figure 3.—Univariate vibration model example. Specific vibration, v .

High-Dimensional Modeling

We trained a VAE to learn a latent bivariate Gaussian distribution. The model was trained on roughly 10 runs, which together contained hundreds of thousands of samples. The runs were taken from a mix of summer and winter months. Validation is plotted in Figure 4. The model learned little with respect to the loss function; however, as we will show, it still provided insight. The performance of the model was highly limited by the data quality. Proper feature selection, data selection, and in-depth analysis of the structure of the samples would likely improve model performance. Thus, this model serves as a rough upper bound on performance.

The latent distributions of the data are plotted and clustered using their bivariate means in Figure 5, using K-means clustering, and in Figure 6, using agglomerative clustering (Ref. 7). The number of clusters were chosen by observing the improvement in intracluster density as the number of clusters increased, as measured by the median Euclidean distance from the cluster center. Under visual inspection, the agglomerative clusters appear to better fit the nuanced structure of the encoded data.

The distributions are averaged together according to their agglomerative cluster, and the averaged distributions are plotted in Figure 7, where the center point is the bivariate mean and the ring around the center point is a single standard deviation from the bivariate mean. The five distributions in Figure 7 represent a model of the expected behavior of the system as defined by a sample of a year's operations. If this model is taken to represent baseline behavior, any encoded run for which data does not occupy regions of the latent space near the clusters can be said to deviate from baseline behavior.

We know from observation and experience that system behavior in summer months differs from system behavior in winter months. We tested to see if this was reflected by our model. Figure 8 and Figure 9 show two summer runs and two winter runs, respectively, plotted over the averaged distributions, where each season is differentiated by color. We can see that the summer runs are similar to each other and that the winter runs are similar to each other.

The differences between the summer and winter runs can be seen more clearly after the distributions are clustered and averaged, as in Figure 7. The averaged summer and winter distributions are plotted in Figure 10 and Figure 11, respectively, using the same convention as Figure 7. We can see that the summer months tend to have behavior occupying higher values for the y-axis, whereas the winter months tend to occupy lower values for the y-axis. These differences show that the model reflects a basic system dynamic we know to exist. Despite the poor fit, the VAE proves able to capture aspects of the system's behavior. This is promising, as we expect future iterations to benefit from feature selection and higher data quality that will improve the fit.

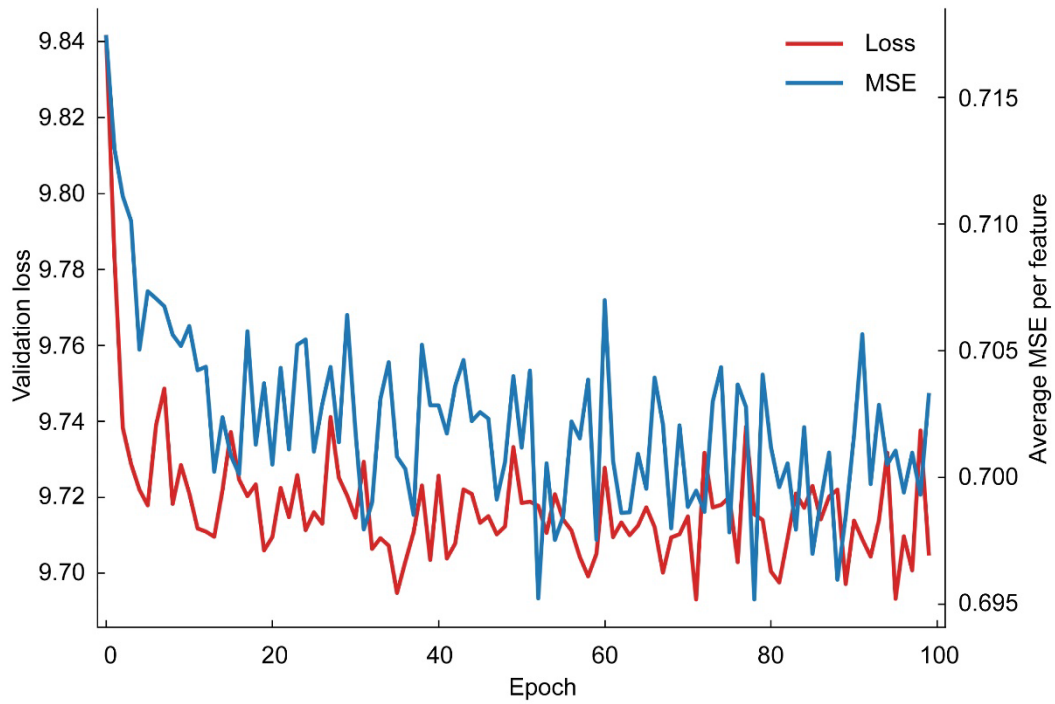


Figure 4.—Variational autoencoder validation. Mean squared error, MSE.

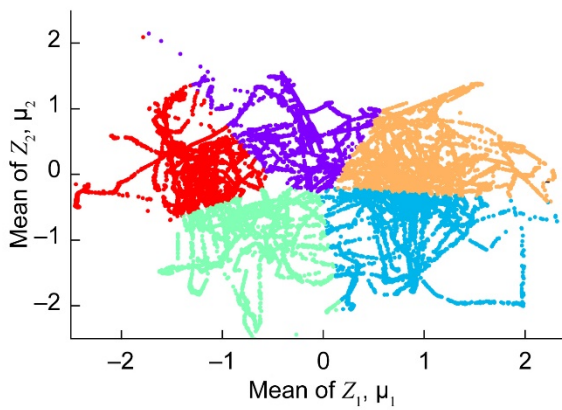


Figure 5.—Full-year K-means clusters. Z is latent space.

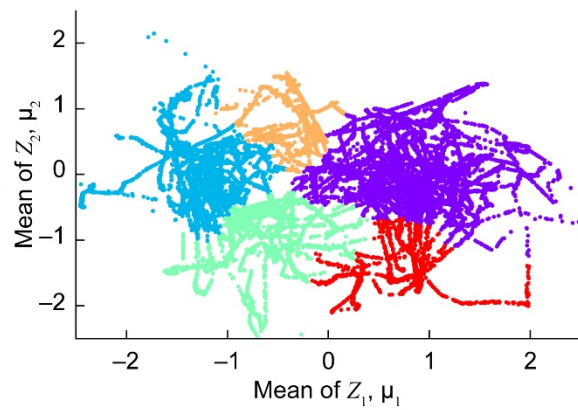


Figure 6.—Full-year agglomerative clusters. Z is latent space.

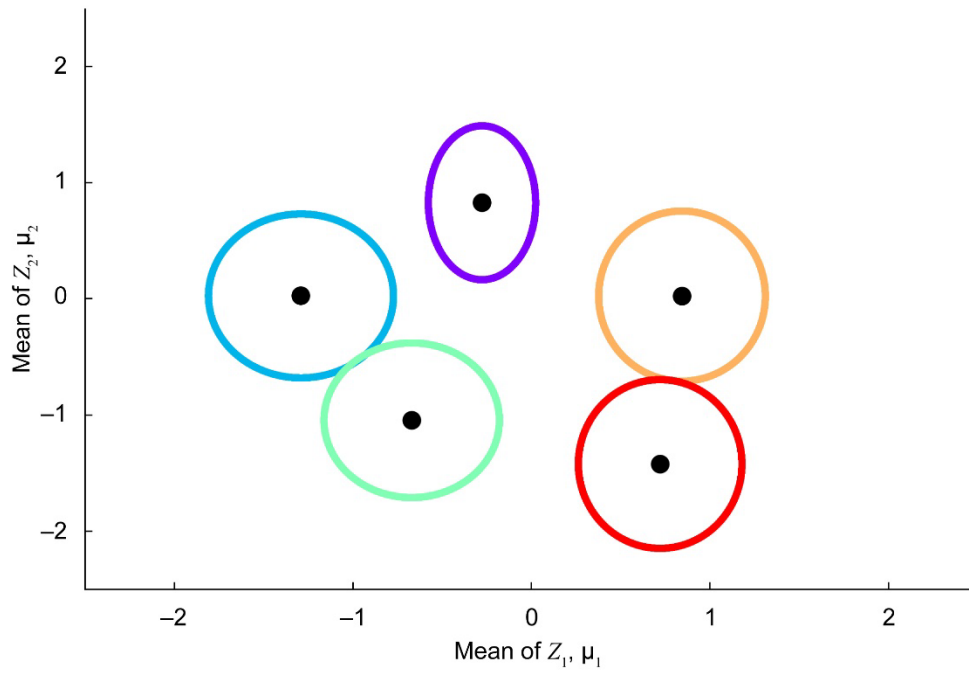


Figure 7.—Full-year grouped distributions. Z is latent space.

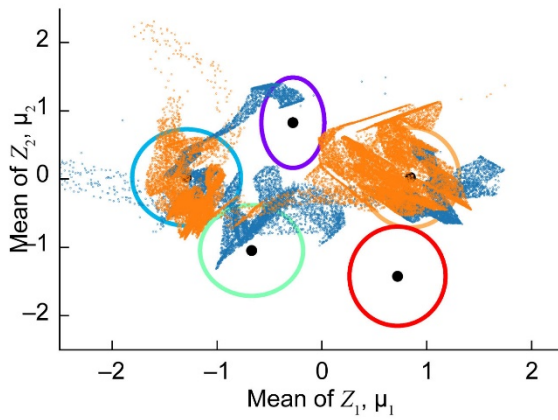


Figure 8.—Summer data; full-year grouped distributions. Z is latent space.

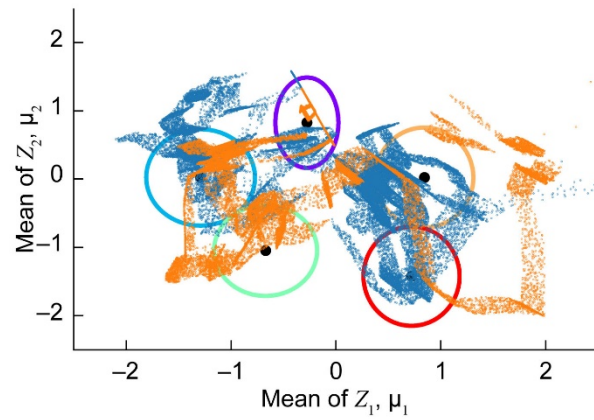


Figure 9.—Winter data; full-year grouped distributions. Z is latent space.

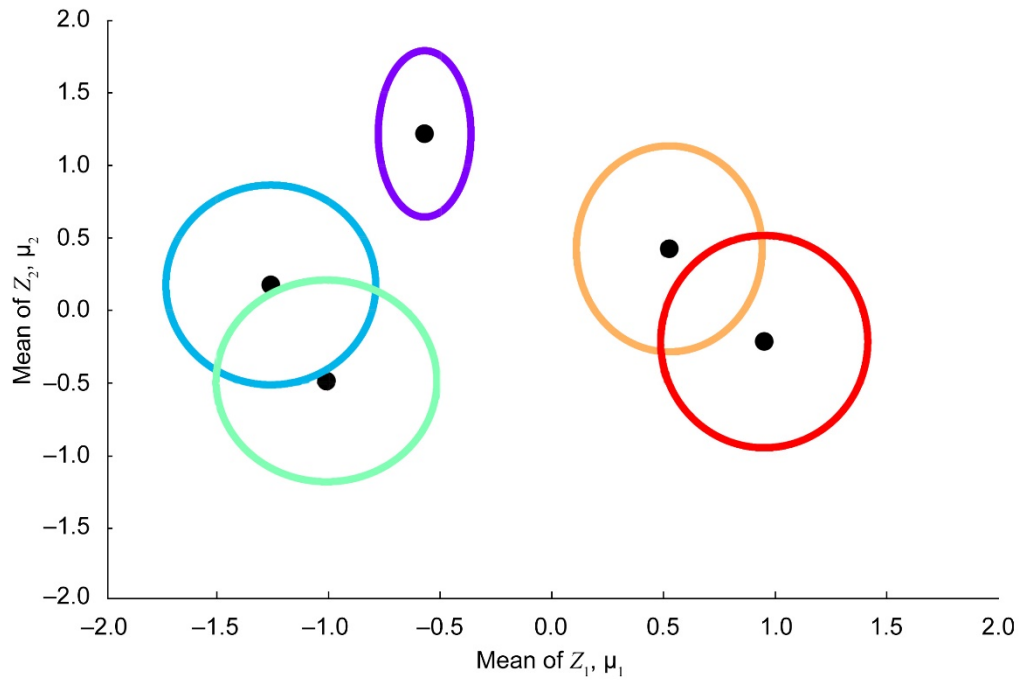


Figure 10.—Summer grouped distributions. Z is latent space.

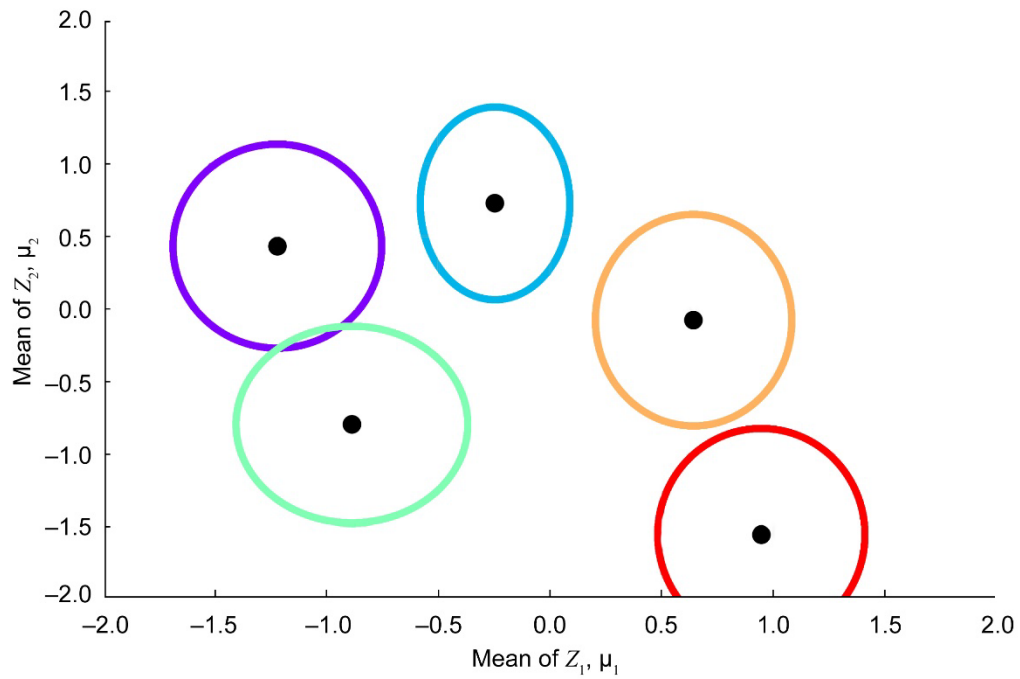


Figure 11.—Winter grouped distributions. Z is latent space.

Concluding Remarks

We fit both univariate and multivariate models to the data. We found that closed-form univariate distributions can capture the behavior of individual sensors and that the variational autoencoders can capture some aspects of system behavior. This work is preliminary and will benefit greatly from future work. In particular, feature selection, data quality, and the development of validation test cases will greatly improve the ability to experiment with model selection and hyperparameter tuning.

We have not shown how these models can be used to realize a multi-model system that uses high-dimensional models to ask simple questions about complex dependent structures, and low-dimensional models to ask complex questions about simple dependent structures. We studied the latent distributions, but we did not demonstrate how we can study differences between latent distributions in terms of actual sensor data by using the decoder to transform a region of the latent space back into the input space. For example, what do the summer and winter cluster differences mean in terms of sensor readings? We can identify a region of interest, decode it, and then study the differences in the sensor readings between the summer and winter months. Future work demonstrating this would show how black-box, deep learning models can be usefully applied to the central air service system in an explainable way that allows system operators to integrate their knowledge with the model output.

This work used a small dataset relative to the available data and considered a particular subsystem of the central air service. Access to more data and more subsystems would increase the performance of models, test the proposed frameworks' ability to scale, and introduce new aspects such as transfer learning and meta learning.

References

1. Jaworski, Piotr, et al., eds.: Copula Theory and Its Applications. Vol. 198, Springer-Verlag, Heidelberg, 2010.
2. Gilks, W.R.; Richardson, S.; and Spiegelhalter, David: Markov Chain Monte Carlo in Practice. First ed., Chapman and Hall/CRC, Boca Raton, FL, 1995.
3. Goodfellow, Ian; Bengio, Yoshua; and Courville, Aaron: Deep Learning. MIT Press, Cambridge, MA, 2016.
4. Abadi, Martín, et al.: Tensorflow: A System for Large-Scale Machine Learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, USENIX Association, Berkeley, CA, 2016.
5. Kingma, Diederik P.; and Welling, Max: Auto-Encoding Variational Bayes. 2013. arXiv preprint arXiv:1312.6114 Accessed Nov. 17, 2020.
6. Goodfellow, Ian J., et al.: Generative Adversarial Nets. Proceedings of the 27th International Conference on Neural Information Processing Systems, vol. 2, MIT Press, Cambridge, MA, 2014, pp. 2672–2680.
7. Pedregosa, Fabian, et al.: Scikit-Learn: Machine Learning in Python. J. Mach. Learn. Res., vol. 12, 2011, pp. 2825–2830.

