Convolutional Variational Autoencoder-based Unsupervised Learning for Power Systems Faults

Maksudul Alam, Srikanth B. Yoginath, Isabelle Snyder, Chris Winstead, and Nils Stenvig
Oak Ridge National Laboratory.
Oak Ridge, USA
{alamm, yoginathsb, snyderib, winsteadcj, stenvignm}@ornl.gov

***Abstract*—Classification of power system event data is a growing need, particularly where non-protective relaying-based sensors are used to monitor grid performance. Given the high burden of obtaining event data with appropriate labeling, an unsupervised approach is highly valuable. This approach enables using event data without labeling, which is far easier to obtain. This paper presents an unsupervised learning method to classify and label transients observed in the distribution grid. A Convolutional Variational Autoencoder (CVAE) was developed for this purpose. We demonstrate the efficacy of our approach using the transient data generated from the simulations. The simulation data is used to train the CVAE that identifies different faults as different clusters in the latent space. The clusters are then used as the foundation model to categorize the real-world data.**

***Index Terms*—Power Grid, Unsupervised Learning, Clustering**

I. Introduction

In power system networks such as distribution grids, electric disturbances are common. The distribution circuits are commonly above ground and exposed to many natural sources of threat, such as severe weather and vegetation overgrowth. These threats manifest as electrical faults which typically result in power outages to nearby customers. The events themselves cause particular characteristics in the electric system behavior which can be observed in the impacted voltage and current waveforms. Protective relays and other devices use these characteristics to isolate system faults. However, not all disturbance types can be characterized and implemented in protection systems (nor do they need to be). The result is that many types of electric disturbances occur in power systems that are not detected nor recorded by protection devices. Other sensor devices are capable of capturing the full range of system event types, and at lower cost (by virtue of not being tied to protective relaying). Unfortunately, these other measurement devices collect data without the common labeling of protection systems. An unsupervised learning approach can use this unlabeled system event data, potentially categorizing far more event types than what standard protection systems can detect. This is an important advancement in grid monitoring and analytics to enable more sophisticated monitoring and control in an increasingly power-electronics-based grid.

Unsupervised learning using neural networks is a rapidly evolving field within artificial intelligence and machine learning. It involves training neural networks to identify patterns and structures in data without labels, unlike supervised learning, where data comes with predefined labels. This approach is particularly useful when labeled data is scarce or expensive to obtain. Neural networks used for unsupervised learning often include autoencoders. Autoencoders learn to compress data into a lower-dimensional space and then reconstruct it back to its original form, effectively learning the most important features of the data [1]. One major challenge is evaluating the model's performance, as there are no predefined labels to compare the results against. Despite the challenge, as computational power increases and algorithms become more sophisticated, neural networks' ability to uncover hidden patterns and insights in vast, unstructured datasets will only grow more impactful.

Different machine-learning techniques have been used in the past to detect faults in electric grid signals. Supervised machine learning methods have been used to detect faults in [2]. However, the supervised learning methods require the undergoing signals to be labeled, which is not readily available for real-world signals. Applying machine learning algorithms in analyzing power signal waveforms is a well-established research area [3]. For example, [4] introduced a methodology combining compressed sensing with a deep convolutional network for disturbance classification. Another study by [5] explored using an artificial neural network and decision trees for event classification, leveraging both time and frequency domain features. Additionally, unsupervised learning techniques have been employed in anomaly detection [6], [7] and event localization [8]. However, much of this research relies on either synthetically labeled data or data from Phasor Measurement Units (PMUs), with minimal focus on distribution grid analytics due to the scarcity of data from the field. In one instance, [9] employed Ward and K-means clustering methods to analyze voltage sag using micro-PMU (μPMU) recordings. [10] provides a comparative review of signal processing and AI-based methods for analyzing power events in smart grids.

One of the primary techniques in unsupervised learning is clustering, in which a neural network groups data points based on similarity. For example, a neural network might cluster customers based on their purchasing behavior, helping businesses to tailor marketing strategies. In [11], an integrated approach using an autoencoder and K-Means clustering was utilized to identify and cluster power events. Their methodology involved compressing 4096 samples down to 60 features using an autoencoder. Subsequently, these 60 features were further reduced to a three-dimensional feature space using Principal Component Analysis (PCA) and K-Means for clustering.

In this paper, we employ an unsupervised learning approach to cluster a set of unlabeled signals from simulations and apply the same algorithm to group real-world signals gathered from sensors. In contrast to [11], our proposed method directly compresses the data into a 2D or 3D latent space using only 1002 samples, eliminating the need for intermediate steps such as PCA and K-Means clustering. The principal contributions of this paper include:

- Developing a Convolutional Variational Autoencoder (CVAE) to cluster synthetic waveform data.
- Analyzing and improving clustering methods that utilize the reduced feature space.
- Exploring the feasibility of using the CVAE model with real-world data.

The remainder of the paper is organized as follows: Section II presents the data generation process, Section III presents pre-processing of the data, Section IV presents the unsupervised learning approach, Section V presents the results, and finally, Section VI concludes the paper.

II. Data Collection

a) Synthetic Data: The model was created in MATLAB Simulink. The individual feeders are modeled in detail, including transformers, line impedances, individual loads, and capacitor banks. To create contingency cases for the machine learning algorithms, different types of faults are applied at the end of each feeder as well as capacitors switching events. The different fault types are three-phase faults (**ABC**), three-phase to ground faults (**ABCG**), Phase to phase (**AB**, **BC**, **CA**), Phase to phase to ground (**ABG**, **BCG**, **CAG**), and Phase to ground (**AG**, **BG**, **CG**). The main components that were varied for the different simulation files are the load percentage across the whole network, the A to D converter noise as random white noise and converter bit number, and the start and end time of the fault within a cycle. The output files created include the following vectors with a sampling frequency of 20kHz, the timestamp, the individual phases' current, the individual phases' voltage, the voltage and current, the signal-to-noise ratio, and a code that is set to 1 at the start of a fault event and to 0 at the end of the event. The position and duration of the fault event vary randomly across different power signals.

b) Real World Data: The second set of input data is a collection of real-world data collected by the Electric Power Board of Chattanooga (EPB). EPB data consists of fast transient events, including 3-phase voltage and 3-phase current data. EPB collects this data at a sample rate of approximately 20,000 samples per second. The events are of variable length, but they could last as short as 0.05 milliseconds and as long as 500 milliseconds. From a practical standpoint, however, longer events have more utility for training.

The electrical data was hosted on a remote server upon collection by EPB, which could then be scraped for events over different time intervals and stored locally for further use as training or validation data as needed. These real-world samples provided a valuable reference against the synthetic data when evaluating the proposed unsupervised learning approach

III. Data Pre-processing

Prior to using unsupervised learning, we conducted an exploratory analysis of the data. The simulation dataset comprises 960 distinct simulation output files, with each file containing $16,001$ data points or samples. These data points include information such as time, voltages, and currents for three phases (Phase **A**, Phase **B**, and Phase **C**), along with a column indicating the presence of faults in the simulated data. Fig. 1 presents an example of the signal. The sequence within each file starts with a segment of normal operation data (shown as green shade in Fig. 1), transitions to a segment indicative of a fault (shown as red shade in Fig. 1), and concludes with another segment of normal operation. This sequence simulates the occurrence of a fault within a typical signal. It's important to highlight that the duration and position of the fault-related data vary across all 960 files, resulting in no uniform length for the fault segments within any given file. To align with the requirements of unsupervised learning, we extract samples of fixed lengths from each file. Given that the data exhibits a periodicity of 334 data points, we select sample lengths of 334, 668, and 1002 data points, corresponding to one, two, and three complete data periods, respectively, for our experiments. Voltage readings range from -57 kV to 55 kV, while current measurements span from -3379 A to 3367 A. We normalized the voltage and current values to a range between 0 and 1 for using for training.

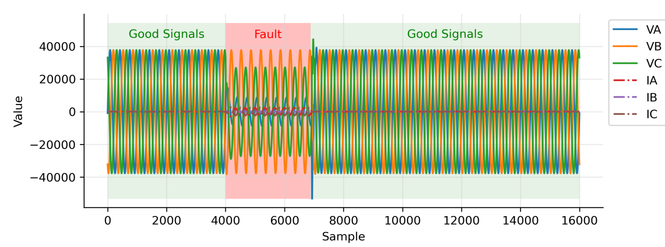


Fig. 1: An example of a signal from the dataset

Owing to the limited size of the dataset, we randomly select multiple samples from both the normal and fault-affected segments of each file for the learning process. Consequently, our dataset comprises a mix of normal and fault-affected signals, each randomly and repeatedly sampled from the files. Our approach to analyzing phase faults involves a multi-faceted analysis, focusing on autocorrelation, inter-signal correlation, and phase angle comparison. We used the following approaches to analyze the data:

Autocorrelation of Fault Signals: Fault signals display an oscillatory pattern with high autocorrelation, repeating every 334 time step. This suggests a 334 time-step window could effectively detect faults, improving model efficiency and reducing dataset noise.

Correlation Within Fault Types: Single-phase fault signals show a strong correlation, especially in voltage, indicating the chosen window size works well across similar fault signals.

Phase Angles Analysis: Using Hilbert transformations for phase angle generation reveals phase synchrony across various fault signals, despite amplitude differences. This highlights the importance of periodic patterns in model refinement.

It is widely recognized that clustering high-dimensional data presents considerable challenges. Therefore, utilizing dimensionality reduction and executing clustering in feature space rather than data space enhances clustering effectiveness [12]. We used a deep convolutional variational autoencoder on these waveforms to capture the essential characteristics of the data in a reduced-dimensional feature space.

IV. Unsupervised Learning using Convolutional Variational Autoencoder

The use of a Convolutional Variational Autoencoder (CVAE) for unsupervised learning of electrical signals, such as those found in power distribution grids is useful for several reasons [13]. CVAEs are adept at extracting features from high-dimensional data and compressing these features into a lower-dimensional latent space [14]. Electrical signals, especially transient signals indicative of faults or other anomalies, can exhibit significant variability. CVAEs can model this variability by learning the distribution of the data in the latent space, allowing for a better understanding of the underlying patterns and variations in the signals. By learning a model of what ”normal” signals look like, CVAEs can be used to identify anomalies or faults when electrical signals deviate significantly from the learned distribution.

The CVAE framework extends the traditional Variational Autoencoder by incorporating convolutional layers, enabling the model to capture spatial hierarchies in data, which is especially beneficial for handling image and sequence data. The foundation of the CVAE can be encapsulated by two main equations that govern its operation: the encoder and decoder networks.

Encoder: The encoder part of a CVAE maps the input data x to a latent space z through a probabilistic mapping. The encoder network outputs parameters to a posterior distribution $q_\phi(z|x)$, typically assumed to be Gaussian:

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x)I)$$

where $\mu_\phi(x)$ and $\sigma_\phi(x)$ are the mean and standard deviation vectors computed by the CNN layers of the encoder.

Decoder: Conversely, the decoder network aims to reconstruct the input data from the latent space. The decoder defines the likelihood of the data given the latent variables, $p_\theta(x|z)$, which is also typically modeled as a Gaussian:

$$p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \sigma^2_\theta(z)I)$$

Here, $\mu_\theta(z)$ and $\sigma_\theta(z)$ are derived from the latent variables using the CNN layers of the decoder.

The training of a CVAE involves optimizing the Evidence Lower Bound (ELBO) on the marginal likelihood of the observed data, which can be broken down into two terms: the reconstruction loss, which encourages the decoded samples to be close to the original inputs, and the KL divergence, which regularizes the learned latent space:

$$\mathcal{L}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x) \| p(z))$$

By integrating convolutional layers, CVAEs not only improve the feature extraction capabilities of standard VAEs but also enhance the model’s applicability to a wider range of real-world data, such as images and sensor signals. This paper explores the potential of CVAEs to effectively cluster and analyze the waveforms.

A. CVAE Model for Power System Faults

Before developing the CVAE model, we initially created a supervised model to classify the data as either normal or fault. The architecture of this model included three sets of convolutional layers, each followed by a batch normalization layer and a pooling layer. We employed ReLU as the activation function. The final output layer consisted of two outputs indicating ”Normal” or ”Fault” signals. During the preprocessing step, we utilized three different sets of input sample lengths: 334, 668, and 1002, to train our model. This base model successfully classified faulty signals from normal signals with an accuracy exceeding 95%, which bolstered our confidence to further develop the variational autoencoder model.

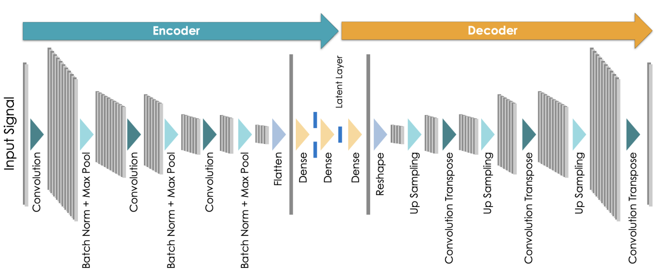


Fig. 2: Convolutional Variational Autoencoder model

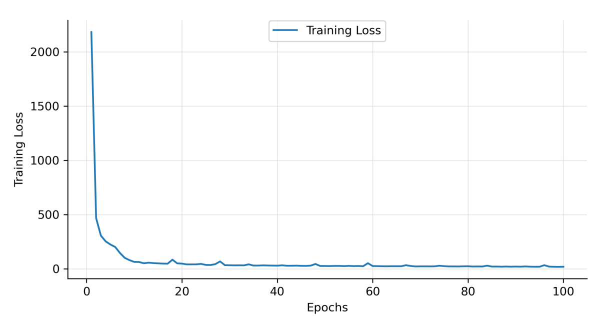
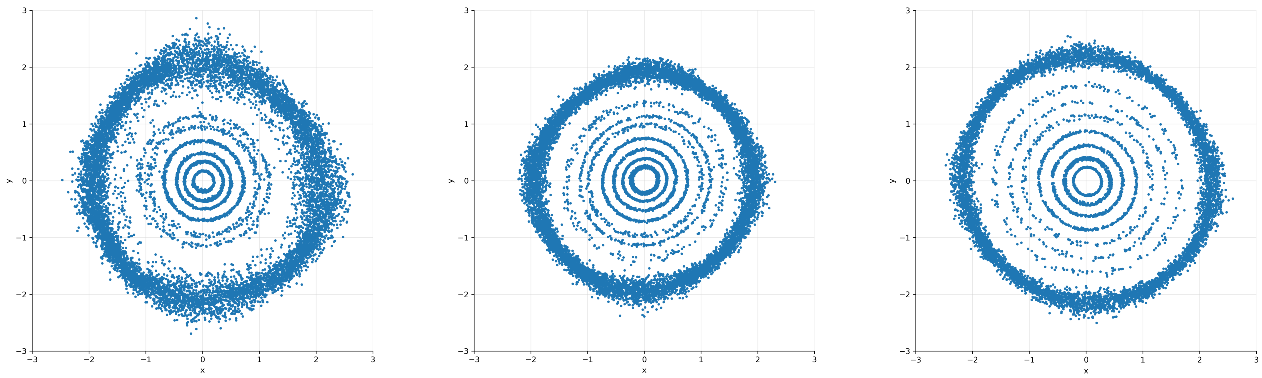


Fig. 3: Convergence of training data

This supervised model served as the encoder part of our initial CVAE model, with the decoder part designed as a mirrored version of the encoder layers. However, we omitted the batch normalization layer from our decoder as it did not enhance the results.



(a) Sample Length = 334 Samples (b) Sample Length = 668 Samples (c) Sample Length = 1002 Samples

Fig. 4: Formation of clusters using the similar CVAE model with different sample sizes

Using our initial CVAE model, we successfully populated clusters that contained four distinguishable groups within a two-dimensional latent space. We experimented with various kernel sizes and channel configurations for the convolutional layers. By adding several additional layers, we refined our model further. This enhanced version of the CVAE model was capable of identifying up to eight distinct clusters in the two-dimensional latent space. Our final CVAE model includes a 12-layer encoder, consisting of a combination of 2D Convolutional layers, Batch Normalization layers, and MaxPooling layers, leading to a latent space of either two or three dimensions. This is complemented by an 8-layer deep decoder, which utilizes multiple repetitions of UpSampling and Transposed Convolutional layers. An illustration of the model configuration is provided in Fig. 2.

We designed the foundational Convolutional Variational Autoencoder (CVAE) framework using TensorFlow 2.16, adhering to a specific model architecture for training. It's important to note that we adapted this CVAE model slightly to accommodate three input dataset sizes with sample lengths of 334, 668, and 1002, ensuring that our approach is robust across different data scales. We trained the model for 100 epochs, which resulted in convergence as illustrated in Fig. 3.

V. Results

To deepen our understanding of the problem at hand, we chose to implement a latent layer with a dimensionality of 2. After undergoing 100 training iterations, we analyzed the latent space distribution corresponding to each input. This distribution is illustrated in Fig. 4. We used sample size of 334, 668, and 1002 to study the formation of clusters. Our model converges quickly within 50 iterations as seen from Fig. 3. Note that our model with the smallest sample size is able to form multiple distinct clusters but the model with 1002 provides the clearest view of cluster formation. We also experimented with models with higher sample sizes but the sample size of 1002 provides a clear clustering of input data and also quite succinct to be useful for faults with smaller data duration.

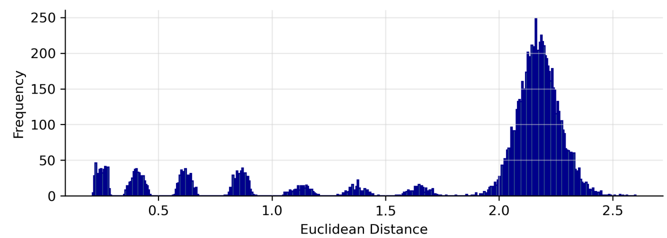


Fig. 5: Histogram of Euclidean distances from the center

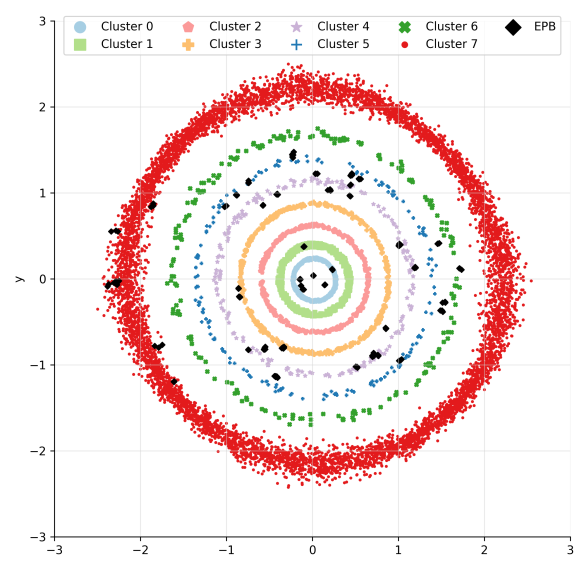


Fig. 6: Formation of clusters based on latent variables

A. Analyzing and Labeling of Clusters using 2D Latent Space

While the clusters are visible in the figures, accurately segregating the data points into distinct groups is challenging. Specifically, Fig. 4(c) reveals the presence of eight distinct circular clusters. To categorize them, we explored several clustering algorithms suitable for circular data, including Spectral Clustering and DBSCAN, yet these methods struggled to cohesively unify the clusters. This led us to investigate alternative, more effective clustering techniques for our dataset. Observing that the cluster formations appeared circular around a central point, we employed Euclidean distance measures to group the data points. The Euclidean distance between the origin point $(0, 0)$ and the two dimensional latent space (x_1, y_1) is given by $d = \sqrt{x_1^2 + y_1^2}$. After calculating the distance, we can plot the distribution of the distances to get an idea of the clustering. The distribution of these distances, depicted in Fig. 5, clearly indicates eight separate clusters with peaks in different distances.

We then devised an algorithm to identify these distinct regions, subsequently organizing the data points into clusters accordingly. The refined clustering, illustrated in Fig. 6, successfully delineates the eight clusters, each represented by a different color.

B. Properties of the Generated Clusters

In the simulated data set, various fault types are labeled, including **AG**, **BG**, **CG** (single-phase faults), **AB**, **BC**, **CA**, **ABG**, **BCG**, **CAG** (double-phase faults), and **ABC**, **ABCG** (three-phase faults). In this section, we examine the composition of each cluster with respect to these fault types to measure the performance of the unsupervised learning method. The details of the faults statistics are shown in Table I.

Clusters 0 through 6 exclusively contain fault data, while all normal signals are grouped into Cluster 7, which comprises 97% of the points in that cluster.

Cluster 0 is characterized by three-phase faults (**ABC** and **ABCG**). Clusters 1 and 2 are composed of two-phase faults such as **AB**, **ABG**, **BCG**, and **BC**. Notably, Cluster 1 predominantly includes AB phase faults, whereas Cluster 2 primarily consists of **BC** phase faults. Cluster 3 is dedicated to two-phase faults involving **CA** and **CAG**. Clusters 4, 5, and 6 contain single-phase faults. Cluster 4 includes **AG** and **CG** faults, with a predominance of **AG** faults. Cluster 5 features **BG** phase faults. Similarly, Cluster 6 contains **AG** and **CG** faults; however, it primarily consists of **CG** phase faults, distinguishing it from Cluster 4, which focuses mainly on **AG** faults.

C. Using Real-World Fault Data with the 2D Latent Space

Next we used real-world data to see how it aligns with the clusters. We used the EPB data for the purpose and used the trained model to produce the latent space from the EPB dataset. The latent space variables are shown in Fig. 6 in black color. As seen from the figure, the EPB data aligns well with the generate clusters.

D. Extending the CVAE Model using 3D Latent Space

To demonstrate the performance of our model with a higher-dimensional latent space, we refined our initial Convolutional Variational Autoencoder (CVAE) to use a three-dimensional latent space. The model was trained for 100 epochs until the training error converged. The distribution of these latent spaces is illustrated in Fig. 7.

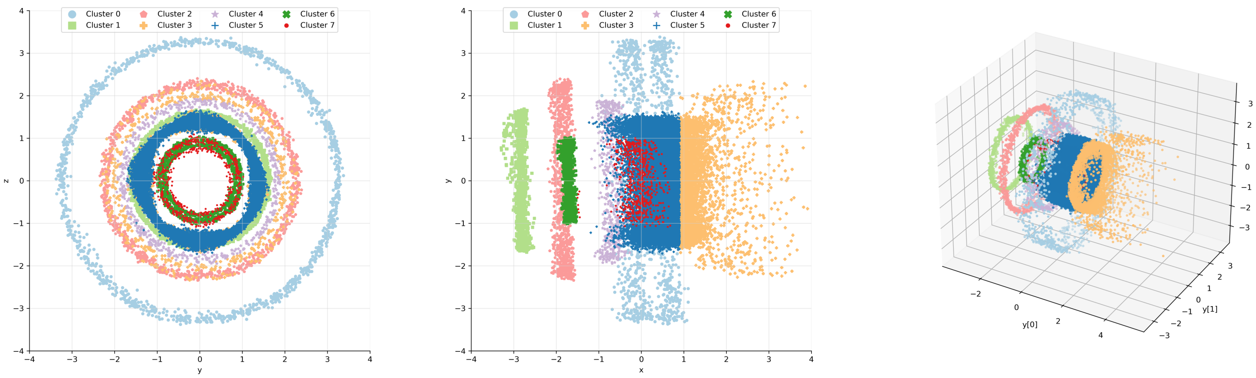
In exploring the structure within the three-dimensional latent space of our Convolutional Variational Autoencoder (CVAE), defined as (x, y, z), we observed the formation of distinct configurations. Fig. 7(a) displays this latent space projected onto a 2D grid along the y and z axes, revealing clusters similar to those identified in the two-dimensional latent space, albeit less distinct. Conversely, Fig. 7(b) illustrates the x, y projection, where the latent space appears as lines parallel to the y axis. Fig. 7(c) depicts the clustering in the three-dimensional space, resembling groups of circular clusters.

From these observations, it is clear that clustering is more complex in the three-dimensional space compared to two dimensions. However, a closer examination allows us to define clusters using two different Euclidean distances. In the (y, z) plane, the first distance from the origin $(0, 0)$ is calculated as $d_1 = \sqrt{y^2 + z^2}$. In the (x, y) plane, the second distance is simply the x-coordinate, defined as $d_2 = x$. This approach enables us to classify points into groups based on ranges of $\langle d_1, d_2 \rangle$, and we have applied clustering accordingly, with the clusters colored distinctively.

Interestingly, the three-dimensional model identifies the same number of clusters (eight) as the two-dimensional model. However, the two-dimensional model proves simpler and more straightforward for defining clustering. Consequently, for this application, the two-dimensional unsupervised model is more appropriate, demonstrating that higher dimensionality does not necessarily lead to a more practical or clearer representation of the model.

TABLE I: Composition of Clusters by Fault Types

	Normal	1-Phase Faults			2-Phase Faults						3-Phase Faults	
		AG	**BG**	**CG**	**AB**	**ABG**	**BC**	**BCG**	**CA**	**CAG**	**ABC**	**ABCG**
Cluster 0											46%	54%
Cluster 1					35%	30%	16%	18%				
Cluster 2					17%	18%	33%	33%				
Cluster 3									45%	55%		
Cluster 4		66%		34%								
Cluster 5			100%									
Cluster 6		37%		63%								
Cluster 7	97%	0.3%	0.4%	0.4%	0.3%	0.3%	0.2%	0.2%	0.4%	0.3%	0.2%	0.3%



(a) Clustering in first two dimensions (b) Clustering in another two dimensions (c) Clustering in three dimensions

Fig. 7: Formation of clusters based on 3D latent space

VI. Conclusion and Future Work

The use of a Convolutional Variational Autoencoder (CVAE) for unsupervised learning to analyze electrical signals from power distribution grids has yielded promising results. Our CVAE model effectively formed distinct clusters in both two-dimensional and three-dimensional latent spaces. However, the effectiveness of the two-dimensional model highlights that higher dimensionality does not necessarily provide more useful insights. The ability of these models to identify various fault types in both simulated and real-world data establishes a strong foundation for future smart grid analytics applications. Future efforts will focus on adapting these models to different data types and enhancing clustering techniques and model architectures to improve their analytical performance.

Acknowledgment

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy (DOE). The work is sponsored by DOE Office of Electricity (OE). The U.S. government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (https://www.energy.gov/doe-public-access-plan).

References

[1] G. San Martín, V. Meruane, E. L. Droguett, and M. Moura, “A Deep Variational Auto-Encoder based Dimensionality Reduction for Fault Diagnosis in Ball Bearings,” in *Safety and Reliability–Safe Societies in a Changing World*. CRC Press, 2018. 1

[2] K. Galbraith, O. Alaca, A. R. Ekti, A. Wilson, I. Snyder, and N. M. Stenvig, “On the Investigation of Phase Fault Classification in Power Grid Signals: A Case Study for Support Vector Machines, Decision Tree and Random Forest,” in *North American Power Symposium*. IEEE, 2023. 1

[3] R. A. de Oliveira and M. H. Bollen, “Deep Learning for Power Quality,” *Electric Power Systems Research*, vol. 214, 2023. 1

[4] J. Wang, Z. Xu, and Y. Che, “Power Quality Disturbance Classification based on Compressed Sensing and Deep Convolution Neural Networks,” *IEEE access*, vol. 7, 2019. 1

[5] A. Aligholian, M. Farajollahi, and H. Mohsenian-Rad, “Unsupervised learning for online abnormality detection in smart meter data,” in *IEEE Power & Energy Society General Meeting*. IEEE, 2019. 1

[6] S. Pandey, A. K. Srivastava, and B. G. Amidan, “A Real Time Event Detection, Classification and Localization using Synchrophasor Data,” *IEEE Transactions on Power Systems*, vol. 35, no. 6, 2020. 1

[7] A. Aligholian, A. Shahsavari, E. M. Stewart, E. Cortez, and H. Mohsenian-Rad, “Unsupervised Event Detection, Clustering, and Use Case Exposition in Micro-PMU Measurements,” *IEEE Transactions on Smart Grid*, vol. 12, no. 4, 2021. 1

[8] H. Li, Y. Weng, E. Farantatos, and M. Patel, “An Unsupervised Learning Framework for Event Detection, Type Identification and Localization using PMUs without any Historical Labels,” in *IEEE Power & Energy Society General Meeting*. IEEE, 2019. 1

[9] T. J. Swenson, E. Vrettos, J. Müller, and C. Gehbauer, “Open μpmu Event Dataset: Detection and Characterization at LBNL Campus,” in *IEEE Power & Energy Society General Meeting*. IEEE, 2019. 1

[10] R. K. Beniwal, M. K. Saini, A. Nayyar, B. Qureshi, and A. Aggarwal, “A Critical Analysis of Methodologies for Detection and Classification of Power Quality Events in Smart Grid,” *IEEE Access*, vol. 9, 2021. 1

[11] M. M. Islam, M. O. Faruque, J. Butterfield, G. Singh, and T. A. Cooke, “Unsupervised Clustering of Disturbances in Power Systems via Deep Convolutional Autoencoders,” in *IEEE Power & Energy Society General Meeting*. IEEE, 2023. 1, 2

[12] M. Ali, A. Alqahtani, M. W. Jones, and X. Xie, “Clustering and Classification for Time Series Data in Visual Analytics: A Survey,” *IEEE Access*, vol. 7, 2019. 3

[13] D. P. Kingma and M. Welling, “Auto-encoding Variational Bayes,” *arXiv preprint arXiv:1312.6114*, 2013. 3

[14] S. Sun, B. Zhang, L. Xie, and Y. Zhang, “An Unsupervised Deep Domain Adaptation Approach for Robust Speech Recognition,” *Neurocomputing*, vol. 257, 2017. 3