Variational Autoencoder Model Combining Deep Learning and Probability Statistics and Its Application in Large-scale Data Analysis

Lingguo Zou[1], Meihua Zhang[2*]
[1]School of Public Education, Xiamen Ocean Vocational College Xiamen 361009, China
[2]College of General Education, Xiamen Huatian International Vocation Institute Xiamen 361102, China
E-mail: MeiHuaZhang0417@163.com
*Corresponding author

Keywords: deep learning, probability statistics, variational autoencoder, bayesian, layer-by-layer learning strategy

Received: August 16, 2024

A multi-layer generative model is proposed as a means of enhancing the accuracy of large-scale data analysis. This model addresses the problem of limited feature extraction capability and insufficient association with label information in existing topic models. The model is divided into three main modules: text encoding, autoencoder inference, and layer-by-layer learning. The model combines a hierarchical Bayesian model with a deterministic upward random downward network structure. It uses a Poisson Gamma Belief Network as a decoder to capture hierarchical implicit features in text data during text encoding, autoencoder inference, and layer-by-layer learning. Random Gradient Monte Carlo sampling is used for posterior inference to improve the model efficiency. In addition, the Fisher information matrix is used to adaptively adjust the learning rate of different levels and topic parameters, and a layer-by-layer learning strategy is introduced to construct a learning network. Based on this, text data and label information are combined for feature extraction. The results demonstrated that the test error rates of the designed model on the 20News, RCV1, and IMDB datasets were 16.52%, 18.72%, and 11.67%, respectively, all of which were the lowest. Additionally, the testing time was the shortest, at 0.020s, 0.017s, and 0.015s, respectively, indicating a high level of accuracy and efficiency. In addition, the perplexity levels on the 20News, RCV1, and Wiki datasets were 590.23, 953.12, and 982.67, respectively, significantly lower than those of other comparison models. Given this, the designed model has high data analysis and interpretation capabilities and relatively high computational efficiency, which can provide scientific tools for accurately analyzing large-scale data in batches.

Povzetek: Predstavljen je izboljšan variacijski avtoenkoder, ki združuje globoko učenje in verjetnostno statistiko za analizo obsežnih in kompleksnih podatkov.

1 Introduction

The advancement of computer technology has made feature information extraction play an important role in the development of computer information analysis systems. It can directly associate information features with subsequent system tasks through supervised learning and obtain label information [1-3]. Deep Learning (DL) and Probabilistic Statistical Models (PSM) have demonstrated strong potential in processing complex data. In text data processing, the combination of DL and PSM can improve the prediction accuracy and computational efficiency of the model, and enhance the understanding of text data. Variational Autoencoder (VAE) is a machine-generated model that maps input data to latent space through an encoder, and then converts latent variables back into raw data through a decoder [4-6]. Traditional VAE often faces certain difficulties in processing high-dimensional complex data, with deficiencies in inference efficiency and computational complexity. To address these challenges, it is essential to fully leverage the logical interpretation capabilities of PSM and enhance the efficiency of large-scale data processing [7-9].

The field of Variational Self-Coding (VSC) has witnessed a steady progression of research in recent years. Mansour R F et al. proposed an unsupervised DL-VAE model for COVID-19 detection and classification. This model utilized an adaptive Wiener filtering strategy for data preprocessing to improve the image quality of the model. This type of model had higher accuracy and better performance than traditional models in multi-classification tasks [10]. Pinheiro Cinelli et al. analyzed the potential generation factors of VAE in the data generation process and analyzed the learning characteristics of VAE under unsupervised conditions.

On this basis, two different image datasets were used for training and sample generation, and the characteristics of VAE under different optimization conditions were fully analyzed [11]. Duan et al. proposed a prediction model that combines VAE and dynamic factor models for modeling financial data noise and predicting stock returns. The feature of this model was to use the prior posterior learning method to approximate the optimal posterior factor model through future information for stock return prediction. Additionally, it employed a variance estimation technique to assess the potential spatial distribution of VSC, thereby establishing a risk model. This model has shown good practical application performance in the stock market [12].

In terms of PSM, Collins et al. analyzed the complexity of the production process of titanium alloy materials. A yield stress prediction equation containing random variables was designed to address the significant variability in composition, microstructure, and mechanical properties during the titanium alloy electron beam additive manufacturing process. They introduced the cumulative distribution function into the physical model and calibrated the experimental data using simulation techniques. This model had better predictive performance than traditional models [13]. Petersen et al. analyzed the distribution center and variability of data information and explored regression, time series, and spatial models through nonlinear methods that conform to density function constraints. They used data instances to analyze the effectiveness of probability models in practical applications [14].

In DL, Lauriola et al. analyzed the application of DL in Natural Language Processing (NLP) and focused on the influence of DL models in different tasks. This study started from various aspects such as software, hardware, and popular corpora of NLP, and then explored the limitations of NLP in the current context. They analyzed the application effect of DL models in NLP from the perspective of limitations [15]. Atz et al.

Table 1: Comparison of perplexity, error rate, and computation time.

Model	Perplexity	Error Rate	Calculate Time (Test Time)
LDA	High (800+)	High (>30%)	Long (>1s)
DocNADE	Medium (700+)	Moderate (25-30%)	Medium (0.5-1s)
DLDA	Medium (700+)	Medium (20-25%)	Long (>1s)
AVITM	Medium (650-700)	Medium (15-20%)	Faster (<0.5s)
sDAM	Low (590.23)	Low (11.67%)	Fast (0.015-0.020s)

analyzed the application of geometric DL models in chemical analysis modeling. This analysis mainly utilized neural network architecture to process chemical molecule symmetry information. It further investigated the potential applications of geometric DL in drug discovery, chemical synthesis prediction, and quantum chemistry. The geometric DL model had a wide range of application prospects [16].

In recent years, research on VAE has mainly focused on unsupervised learning applications and expanded to fields such as image processing and financial modeling. PSM has been deeply applied in complex system modeling, data distribution, and other aspects. DL is commonly used in information analysis such as NLP and medical diagnosis.

At present, VAE cannot efficiently process large-scale data in data analysis. Furthermore, the full integration of DL and PSM into VAE remains an ongoing process. Therefore, this study innovatively combines the DL, PSM, and VAE to form a Deep Autoencoding Topic Model (DAM), which improves the processing efficiency of large-scale data.

Compared with existing models such as Linear Discriminant Analysis (LDA), DocNADE, Direct Linear Discriminant Analysis (DLDA), Autoencoding Variational Inference for Topic Models (AVITM), etc., the proposed model performs relatively better in perplexity, error rate, and computation time. Among them, the LDA model, as a classic topic model, has a high overall error rate and confusion. The DLDA and DocNADE models incorporate techniques such as autoregression and DL but still have shortcomings in terms of confusion and computational efficiency. The AVITM model, like the model studied and designed in this study, also includes self-coding techniques. However, the level of confusion is still insufficient. Given this, the proposed model has a perplexity of 590.23, an error rate of 11.67%, and a testing time of only 0.015-0.020 seconds. Overall, there has been an improvement in reasoning ability, accuracy, and computational efficiency. Due to the combination of the Poisson Gamma Belief Network (PGBN), the model can perform fast inference and capture deeper-level feature information.

The research mainly has three parts. Part 1 designs a DL Self-Coding Theme Model (SCTM) based on tag information. Part 2 validates the performance of the model. Part 3 draws research conclusions.

2 Methods and materials

2.1 DL self-coding theme model design

To address such problems, it is necessary to fully utilize the logical interpretation capability of PSM and improve efficiency in large-scale data processing. However, this cannot be achieved without first addressing the issue of

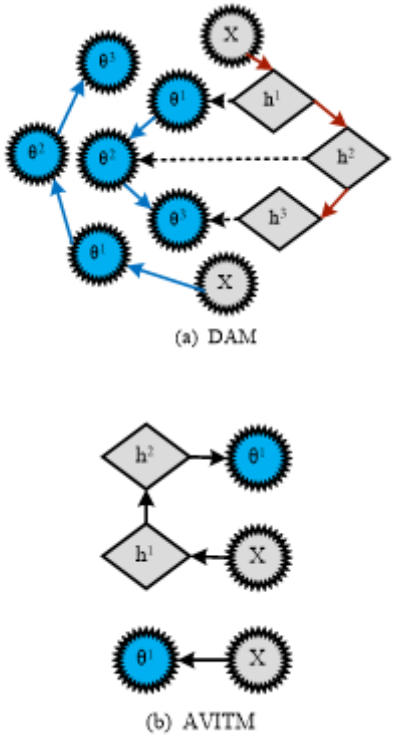


Figure 1: DAM and AVITM model structures.

forming logically smooth data analysis and processing [17-19]. Therefore, this study proposes the DAM. This model combines a hierarchical Bayesian model with a "deterministic upward random downward" network structure to provide more accurate topic representations for data. This model can be divided into three main modules: text encoding, autoencoder inference, and layer-by-layer learning. In the encoding and decoding process of the model, a PGBN model is used as the decoder to capture hierarchical implicit features in text data. The AVITM algorithm mainly uses separate and shallow LDA for decoding [20-22]. This study is different, using a deep generative model for decoding. The DAM and AVITM models are shown in Figure 1.

In PGBN, the generative model utilizes multiple hidden layers to recursively model high-dimensional sparse data.

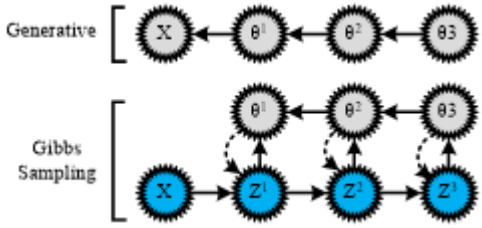


Figure 2: Structure of DLD model.

The network applies Dirichlet prior constraints on the topic weights at each layer, thereby making the topic weights normalized. All theme weight elements must be greater than 0, and the sum of all elements must be 1, which can ensure the simplification of the model inference process. The generative model with hidden layers can be represented as formulas (1) to (3).

$$\theta_n^{(L)} \to Gam\left(r, 1/c_n^{(L+1)}\right), \quad r \to Gam\left(\gamma_0 / K_L, 1/c_0\right) \tag{1}$$

In formula (1), $\theta_n^{(L)}$ is the weight of the hidden layer, L is the number of hidden layers, and r is the shape parameter of the gamma distribution, then formula (2) exists.

$$\theta_m^{(l)} \to Gam\left(\Phi^{(l+1)}\theta_n^{(l+1)}, 1/c_n^{(l+1)}\right), \quad l = 1, \ldots, L-1 \tag{2}$$

If $\Phi^{(l+1)}$ is the factor loading matrix and $1/c_n^{(l+1)}$ is the scale parameter in formula (2), then formula (3) exists.

$$x_n \to Pois\left(\Phi^{(1)}\theta_n^{(1)}\right) \tag{3}$$

The Depth Linear Discrimination (DLD) can be expressed as formula (4).

$$x_{kn}^{(1)} = \sum_{k'=1}^{K_1} x_{kk'n}^{(1)}, k = 1, \ldots, K_0 \tag{4}$$

DLD is a representative supervised subspace analysis method, as shown in Figure 2.

To process large-scale data, this study adopts Stochastic Gradient Monte Carlo (SGMC) as a posterior inference algorithm, which adaptively improves inference efficiency in a layered manner. SGMC is a technique that utilizes random gradient information for sampling, enabling efficient parameter estimation on large-scale data. The sampling process can be specifically expressed as formula (5).

$$z_{t+1} = z_t + \varepsilon_t \left\{ -\left[D(z_t) + Q(z_t) \right] \nabla H(z_t) + \Gamma(z_t) \right\} + \mathrm{N}\left(0, \varepsilon_t \left[2D(z_t) - \varepsilon_t B_t \right]\right) \tag{5}$$

In formula (5), ε_t is the step size, B_t is the variance of the random gradient estimation noise, and $2D(z_t)$ is the positive definite condition. The calculation for $H(z_t)$ and $\Gamma(z_t)$ is formula (6).

$$\begin{cases} H(z) = -\ln p(z) - \rho \sum_{x \in \hat{X}} \ln p(x|z) \\ \Gamma_i(z_t) = \sum_j \dfrac{\partial}{\partial_{zjt}} \left[D_{ij}(z_t) + Q_{ij}(z_t) \right] \end{cases} \tag{6}$$

In formula (6), ρ is the ratio between the size of the dataset and the size of the subset. Under this framework, formula (7) needs to be satisfied.

$$D(z) = G(z)^{-1}, Q(z) = 0, B_t = 0 \tag{7}$$

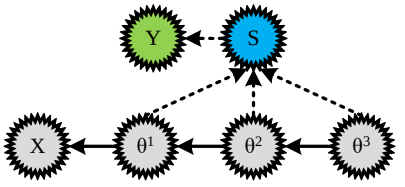
In formula (7), $G(z)$ is the Fisher Information Matrix (FIM), calculated as in formula (8).

$$G(z) = \mathrm{E}_{\Pi|z} \left[-\frac{\partial^2}{\partial z^2} \ln p(\Pi|z) \right] \tag{8}$$

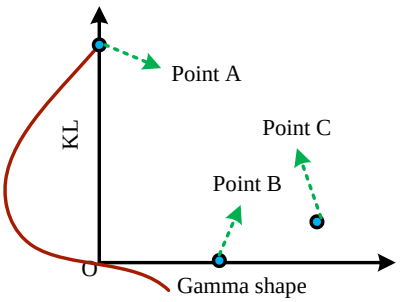
In formula (8), Π represents the observed and local variables. FIM can provide adaptive learning rates in gradient computation, achieving faster convergence in the inference process by adaptively adjusting the learning rates of different levels and topic parameters. It calculates $D(z)$ by providing a step size to improve computational efficiency. Since the topic weights are vectors on the probability normalization plane, it is necessary to convert the original variables into a form suitable for the probability normalization plane through parameterization. Furthermore, by adjusting the Weibull distribution parameters, the gradient instability during the sampling process is eliminated, and the efficiency of large-scale data processing is improved in the form of reducing computational complexity. After parameterization conversion, the sampling process evolves into formula (9).

$$(\varphi_k)_{t+1} = \left[(\varphi_k)_t + \frac{\varepsilon_t}{M_k} \left[(\rho \tilde{x}_{:k} + \eta) - (\rho \tilde{x}_{:k} + \eta V)(\varphi_k)_t \right] + \mathrm{N}\left(0, \frac{\varepsilon_t}{M_k} [diag](\varphi_k)_t - (\varphi_k)_t (\varphi_k)_t^T \right) \right]_\Delta \tag{9}$$

In formula (9), $[\ ]_\Delta$ represents the constraint. To update parameters more effectively during the inference process, this study introduces a method of adaptive step size for each layer of the topic. This method utilizes FIM to adaptively adjust the learning rate of different levels and thematic parameters. In each iteration, it uses sampling methods to approximate local variables, updates them layer by layer, and optimizes the estimation results of global and local variables. To further achieve fast inference, the VSC method is adopted to design a self-variational coding network. This model can project observation data into a hidden space, avoiding the iterative process during the testing phase and significantly improving inference speed. To effectively handle sparse text representations, this study introduces Weibull distribution as the probability distribution of latent variables. The parameter changes of gamma distribution scale are shown in Figure 3.



(a) sDPATTM

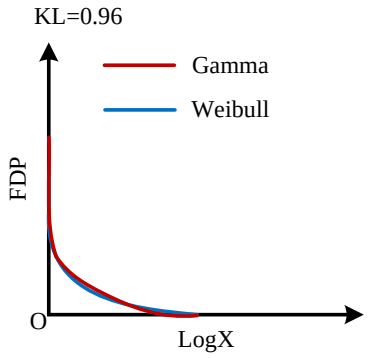


(b) KL change

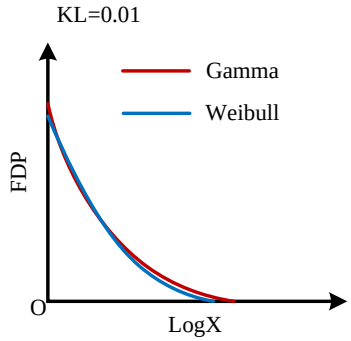
Figure 3: Parameter changes of gamma distribution scale.

When the shape parameters of Weibull are not large enough or not close enough to 0, the estimated Weibull distribution can be more approximate to a given large gamma distribution. The distance between the Weibull distribution and the target gamma distribution is shown in Figure 4.

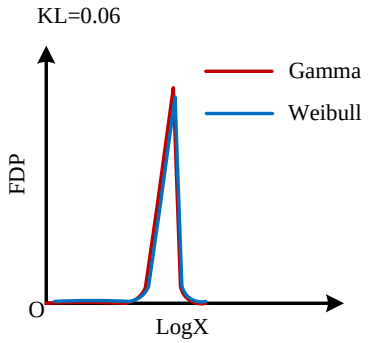
The initial step is to ascertain the shape and scale parameters of the Weibull distribution. These parameters are then employed to generate latent variable representations, thereby establishing a mapping from observed data to the parameters of the Weibull distribution. By combining bottom-up deterministic transmission with top-down stochastic transmission, the accuracy of inference is improved. Through this Up and Down Variational Coding Network (UDVCN), the information transmission between the upper and lower layers can be effectively utilized during the inference process. Unlike traditional models, UDVCN not only considers



(a) Gamma (0.05,1.00)



(b) Gamma (0.5,1.0)



(c) Gamma (5,1)

Figure 4: The distance between Weibull distribution and target gamma distribution.

the bottom-up information flow but also takes into account the prior information of the generated model, achieving more accurate posterior estimation in the inference process.

The expression of UDVCN is shown in formula (10).

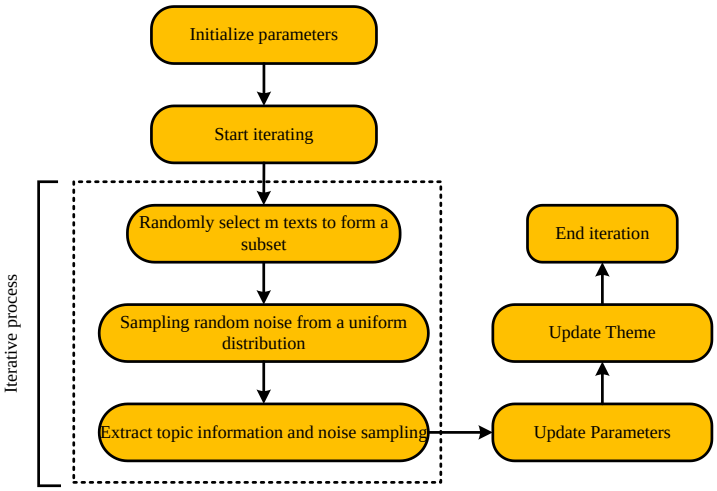


Figure 5: Self-coded variational inference algorithm.

$$q\left(\theta_n^{(l)} \middle| \Phi^{(l+1)}, h_n^{(l)}, \theta_n^{(l+1)}\right) = Weibull\left(k_n^{(l)} + \Phi^{(l+1)}\theta_n^{(l+1)}, \lambda_n^{(l)}\right) \quad (10)$$

This mechanism of information transmission allows the model to more efficiently capture low-level features and higher-level abstract features in the data, making the inference process more stable and efficient. To jointly solve the theme parameters of the decoding network and the neural network parameters of the encoding network, this study introduces a mixed stochastic gradient and VAE inference algorithm to improve the posterior inference speed. The algorithm flow is shown in Figure 5.

During the model training process, a Layer-by-layer Learning Strategy (LLS) is used to construct the learning network. LLS is an effective method for optimizing the structure of DL models. This strategy allows for the gradual inference of the optimal width for each layer of the network starting from an initial set width, thereby achieving more accurate model training. This strategy does not require specifying the width of each layer. Given the width of the first layer, the model will spontaneously infer the width of each layer. After the training of the previous layer is completed, the algorithm will stage the gamma non-negative binomial process and model the hidden integer data of the new layer. This adaptive strategy can effectively reduce useless network parameters, improve the training efficiency and prediction accuracy of the model. Through LLS, the optimal structure of the network can be effectively inferred given the initial layer width. The process of LLS is shown in Figure 6.

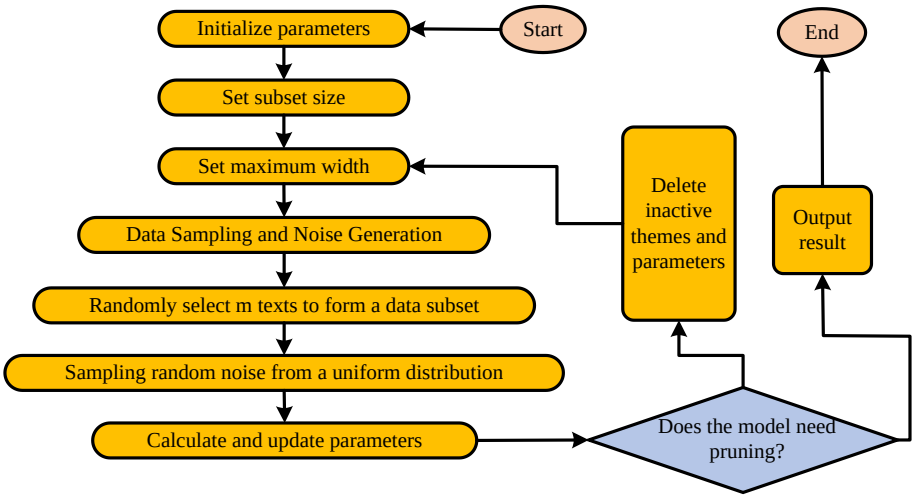


Figure 6: Layered learning algorithm.

Overall, this study uses the Weibull distribution to approximate the gamma distribution, and then determines the parameters of the Weibull distribution by minimizing the distance between the Weibull distribution and the gamma distribution. Then, through UDVCN, the posterior distribution of the latent variable is quickly estimated. To further improve computational efficiency, global variables are randomly updated in each iteration, while local variables are sampled through inference networks to achieve efficient parameter updates.

2.2 SCTM based on tag information

The designed DAM can effectively infer thematic data, project the original text towards multiple layers of random hidden space, and generate label information closely related to the text. Traditional models are unable to closely correlate feature extraction with label information. Therefore, this study combines modeling text with label information to improve classification and recognition performance. This study further improves DAM to form a supervised deep attention topic model. This model can perform text classification while generating text and combine text data with label information. The operating principle of traditional autoencoder models is to first extract text features unsupervised, and then train a classifier using data features [23-24]. However, its limitation is that feature information extraction is difficult to correlate with label information, resulting in low classification performance. To address this issue, this study combines text modeling with label information to enhance classification and recognition capabilities. In text datasets with labeled information, label generation is based on category distribution. All labels are closely related to category probabilities. In the DAM designed for this research, the hidden representations of the various layers are randomly connected, and the themes of the different layers display

varying levels of features. The first layer of features is directly related to data generation. Therefore, to strongly correlate the label information with the learning ability of each layer, this study concatenates the features of each layer to form the final text features, as shown in formula (11).

$$s_n = \left[\theta_n^{(1)}, \ldots, \theta_n^{(L)} \right] \quad (11)$$

In formula (11), $\theta_n^{(L)}$ is the feature of each layer. After determining the connection, label information can directly affect different layers. After concatenating the features, they are projected onto the label probability space through nonlinear methods. This is because linear methods can only simply project features onto label probabilities. Nonlinear methods can project the features of each layer onto a shared hidden space through multi-layer perceptron, and then concatenate them to obtain a joint modeling of text data and label information. The calculation after projection and concatenation is given by formula (12).

$$s_n = \left[g_1^{(l)}\left(\theta_n^{(L)}\right), \ldots, g_1^{(L)}\left(\theta_n^{(L)}\right) \right] \quad (12)$$

In formula (12), $g_1^{(L)}$ is the projection process of the multi-layer perceptron. On this basis, a confidence lower bound for joint modeling can be obtained, and the expected values can be calculated through variational inference to optimize the model parameters. The setting of prior and posterior distributions ensures the robustness of the model, and non-negativity is ensured through logarithmic parameterization, while the backpropagation algorithm is used to update the model parameters. The final model formed is as shown in formulas (13) to (15).

$$g_1^{(l)}\left(\theta_n^{(l)}\right) = \ln\left[1 + \exp\left(W_{m1}^{(l)}\theta_n^{(l)} + b_{m1}^{(l)}\right)\right] \quad (13)$$

Formula (13) is the first layer parameter of the multi-layer perceptron, and the second layer parameter is given by formula (14).

$$h_n = \ln\left[1 + \exp\left(W_{m2}^{(l)}\theta_n^{(l)} + b_{m2}^{(l)}\right)\right] \quad (14)$$

Based on formula (14), the third layer parameters can be obtained, namely formula (15).

$$g_2\left(s_n\right) = \ln\left[1 + \exp\left(W_{m3}h_n + b_{m3}\right)\right] \quad (15)$$

Establishing a strongly correlated DAM can effectively combine text generation and label information, improve text classification performance, and enhance feature discrimination ability.

3 Results

In this study, the Deep Boltzmann Machine model (OR-softmax), LDA, Autoregressive Distribution Estimation model based on forward deep neural network (DocNADE), Deep Poisson Factor Analysis (DPFA), AVITM, DLDA model using Gibbs sampling inference (DLDA-Gibbs), DLDA model using TLASGR-MCMC inference (DLDA-TLASGR), and research model (sDAM) are introduced for horizontal model comparison. Table 1 shows the comparison data of model perplexity and testing time.

Perplexity in the table refers to the uncertainty of the model with respect to the data. The lower the Perplexity, the better the model's data processing ability. In the 20News, RCV1, and Wiki datasets, the Perplexity of sDAM under the 128-64-32 structure is 590.23, 953.12, and 982.67, which is the lowest compared to other models. This indicates that sDAM with the deepest network structure has the strongest data interpretation ability. From the perspective of testing time, sDAM under the 128-64-32 structure has the lowest testing time in all three datasets, at 0.21 s, 0.72 s, and 0.85 s, respectively. This indicates that the model has the shortest computation time and the highest efficiency. Therefore, the research model can demonstrate the highest data interpretation ability under appropriate network structure, and can achieve high information interpretation completion while efficiently responding to urgent tasks.

Figure 7 shows the adaptive step size variation of the proposed DAM on three datasets: 20News, RCVI, and Wiki, where the step size of a single layer is the average of all topic steps within the layer. The step size of high-level data is relatively large, and the larger the dataset, the smaller the step size. The Wiki theme has the largest volume and the smallest step size. The smaller the step size, the more stable the model, indicating that the research model is actually more stable when

Table 2: Comparison of model confusion and testing time.

Model	Structure	Perplexity			Test time (s)		
		20News	RCV1	Wiki	20News	RCV1	Wiki
OR-softmax	128-64-32	610.75	1018.64	1040.23	3.45	8.91	10.12
DPFA	128-64-32	658.12	1054.23	1071.65	20.45	34.89	35.75
LDA	128	887.32	1042.93	1070.45	4.25	11.52	12.45
DocNADE	128	606.83	973.19	1011.47	0.46	0.95	1.11
AVITM	128	678.25	1075.62	1099.83	0.27	0.73	1.51
DLDA-Gibbs	128	618.92	979.45	1010.23	4.92	13.02	13.76
DLDA-Gibbs	128-64	614.53	977.84	1006.83	9.12	18.95	20.13
DLDA-Gibbs	128-64-32	610.47	975.23	1004.67	10.78	23.95	24.32
DLDA-TLASGR	128	620.25	975.45	1008.23	4.92	13.03	13.75
DLDA-TLASGR	128-64	605.13	969.57	995.34	9.14	18.92	20.12
DLDA-TLASGR	128-64-32	599.78	964.43	994.12	10.78	23.92	24.31
sDAM	128	608.35	962.19	986.41	0.65	1.28	0.85
sDAM	128-64	595.67	957.84	984.53	0.42	0.98	1.09
sDAM	128-64-32	590.23	953.12	982.67	0.21	0.72	0.85

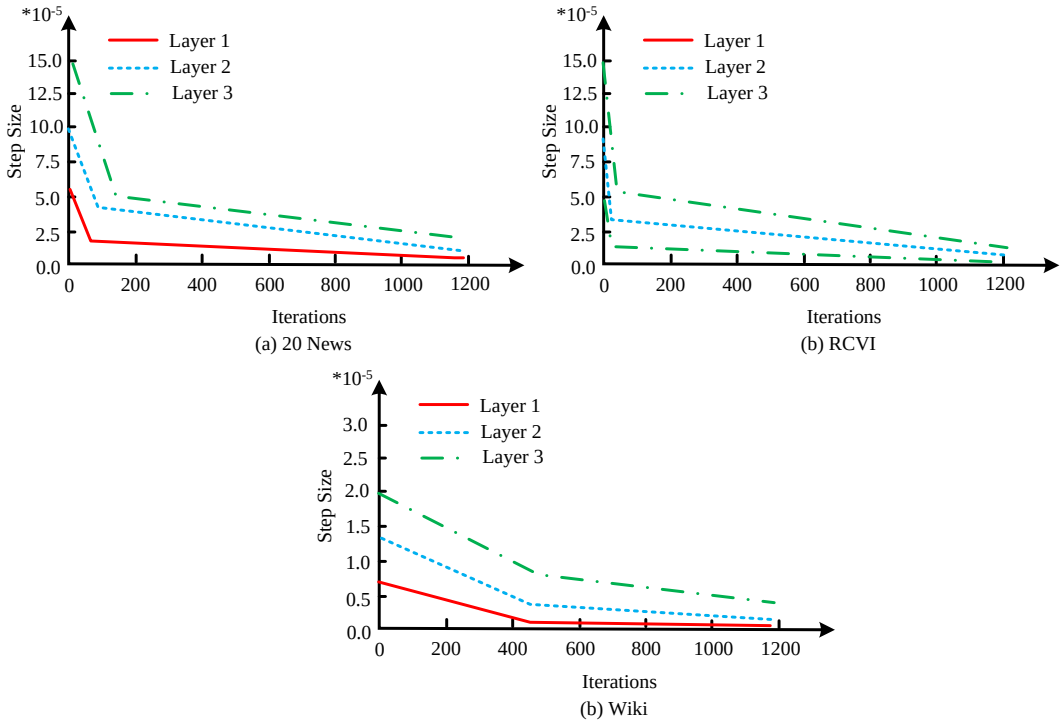
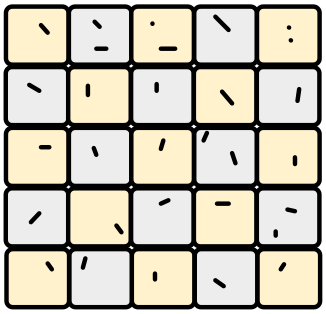
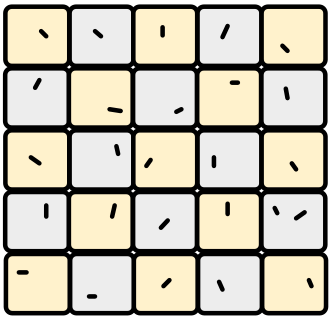


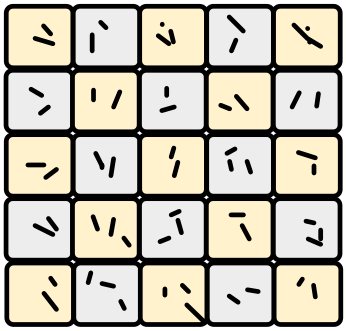
Figure 7: Adaptive step size variation.



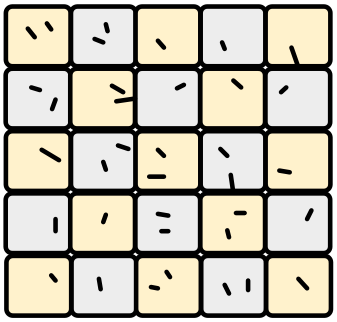
(a) Top down structure: first layer



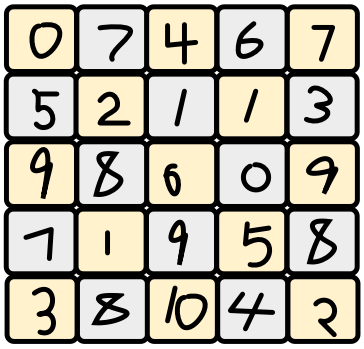
(d) Bottom up structure: first layer



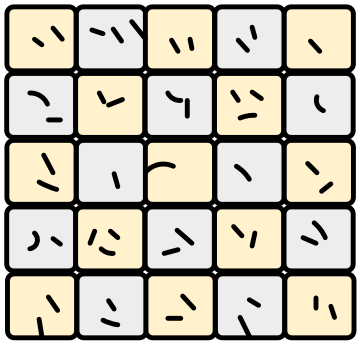
(b) Top down structure: second layer



(e) Bottom up structure: second layer



(c) Top down structure: Third layer

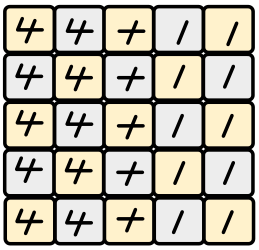


(f) Bottom up structure: third layer

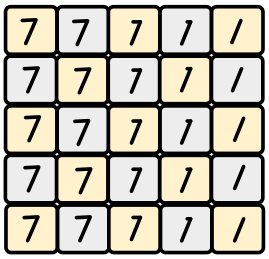
Figure 8: Application effect of handwritten dataset.

facing large-scale data.

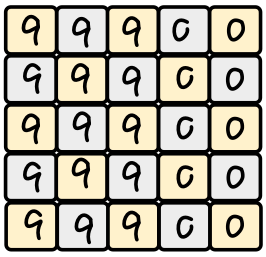
Figure 8 shows the application effect of DAM on handwritten dataset. Figures (a) to (c) show the top-down stochastic network inference structure (Structure 1). Figures (d) to (f) show the bottom-up deterministic network inference structure (Structure 2). According to the three-layer structure of the designed multi-layer perceptron, it is divided into three layers. Regarding Structure 1, Figure 8 (a) serves as the first layer and only learns local points, representing the most basic details in handwritten digit images, mainly capturing low-level feature information. As the second layer, Fig.8 (b) has learned some contour shapes, and the information features are gradually becoming more apparent, but still unclear. As the third layer, Figure 8(c) can already distinguish obvious numerical forms. This information feature representation is a further combination of the first two layers of features, demonstrating its ability to capture advanced information features. Overall, Structure 1 can successfully capture information features. For Structure 2, Figure 8 (d) serves as the first layer and only learns local points, capturing low-level feature information. The information capture situation is similar to Figure 8 (a). As the second layer, although the captured information features in Figure 8 (e) have increased, the information features are still not clear. As the third layer, although some information features have been captured in Figure 8 (f), the information



(a) Column 1



(b) Column 2



(c) Column 3

Figure 9: Interpolation effect.

features are still not clear and cannot distinguish the full picture of the information. The overall information form is far less than in Figure (c). Overall, Structure 2 has stronger feature capture ability.

Figure 9 shows the interpolation effect of hidden space on the MNIST dataset. The image gradually transitions from the leftmost row to the rightmost row. Under model interpolation, the numerical changes are smooth, and the samples formed in the middle of the interpolation process are interpretable, allowing for a full observation of the changing state. This indicates that the model has better inference performance.

Figures 10 (a) and (b) show the changes in the model as the network structure changes. The two graphs show a consistent trend of change, that is, the deeper the network architecture of the model, the smaller the change in test error rate. Figure 10(c) shows the variation of model error rate with sample size in the 20News dataset. As the sample data increases, the error rate of the model generally shows a downward trend. In the comparison results, sDAM has the lowest error rate, showing a decreasing trend in the range of 25% to 0%, and when the sample size reaches 12,000, the error rate is almost 0%.

The table compares the model testing error rate and testing time on the 20News, RCV1, and IMDB datasets. On the 20News dataset, the sDAM-N model has the lowest test error rate at only 16.52%, while the FNN-BOW model has the lowest test error rate at 32.1%. Among all the comparison models, the sDAM-N model has the highest accuracy. On the RCV1 dataset, the sDAM-N model still has the lowest test error rate at only 16.52%, while the LDA model has the highest. On the IMDB dataset, the sDAM-N model still has the lowest error rate with an error rate of 11.67%, while the AVITM model has the highest error rate. On all three datasets, the sDATM-N model has the lowest error rate. On the 20 News, RCV1, and IMDB datasets, the sDAM-N model has the shortest testing time, only 0.020 s, 0.017 s, and 0.015 s, while the DLDA model has the longest testing time, at 1.18 s. Overall, the sDAM-N model has the lowest error rate and shortest computation time on all datasets, indicating its highest computational efficiency and accuracy. In the process of changing the size of the

dataset, the overall calculation accuracy and calculation time of the model are relatively stable, indicating that the model has stability and accuracy.

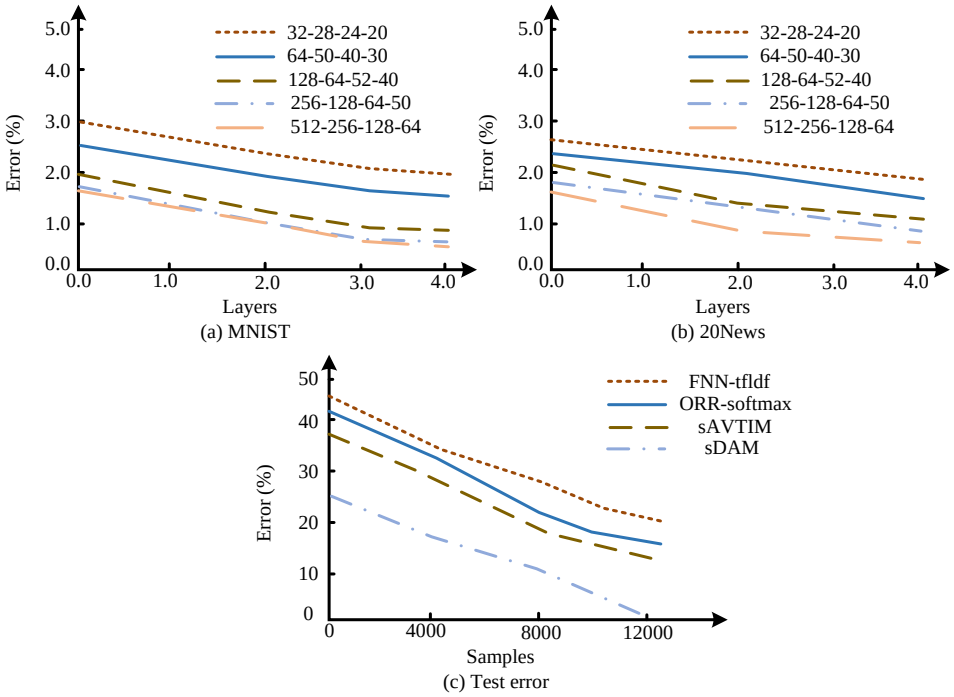


Figure 10: Error rate changes with sample size.

Table 3: Horizontal comparison.

Model	Error rate			Test time		
	20News (%)	RCV1 (%)	IMDB (%)	20News(s)	RCV1(s)	IMDB(s)
LDA	24.850	25.120	22.370	0.730	0.310	0.560
DLDA	21.540	19.760	17.630	1.350	0.850	1.180
DocNADE	24.010	21.500	18.320	0.042	0.027	0.037
OR-softmax	23.050	21.020	19.250	0.780	0.260	0.620
AVITM	25.190	24.890	22.610	0.027	0.020	0.022
DAM	23.720	22.440	20.890	0.029	0.022	0.024
FNN-BOW	32.100	27.560	20.450	0.021	0.018	0.020
FNN-tfidf	25.010	18.720	17.520	0.022	0.015	0.017
AVITM	19.640	17.980	16.730	0.024	0.019	0.021
Med LDA	19.230	17.100	14.720	0.308	0.122	0.278
wv-LSTM	18.520	15.620	14.320	-	-	-
sDAM-N	16.520	14.230	11.670	0.020	0.017	0.015

Table 4: Accuracy comparison analysis.

Model	Data set	Perplexity	Error rate (%)	Accuracy (%)	Recall (%)	F1 score (%)
AVITM	20News	953.12	16.52	83.45	79.23	81.29
LDA	20News	750	32.1	75.2	70.5	72.78
sDAM-N	20News	590.23	12.52	88	82.5	85.1
LDA	IMDB	1100	40	70	65	67.5

sDAM-N	IMDB	982.67	11.67	90	88	89
AVITM	IMDB	1150	35	60	58	59

Table 5: Cross validation experiment.

Model	Data set	First fold error rate (%)	Second fold error rate (%)	Fourth fold error rate	Fourth fold error rate	Fifth fold error rate (%)	Average error rate (%)
sDAM-N	20News	16	15.8	16.2	16.1	16.3	16.1
sDAM-N	RCV1	16.5	16.7	16.8	16.4	16.6	16.62
sDAM-N	IMDB	11.5	11.8	11.6	11.4	11.7	11.62

Table 6: Ablation experiment.

Model	Error rate (%)	Perplexity	Accuracy (%)	Recall (%)	F1 score (%)
sDAM-N	16.52	590.23	83.45	79.23	81.29
SDAM-N (Remove Hierarchical Learning)	20.4	620	76.5	72	74.2
SDAM-N (with FFIM removed)	18.8	610	80	75	77.4
SDAM-N (remove both)	25	700	70	65	67.5

The error rate of the sDAM-N model on the IMDB dataset is only 11.67%, while the error rates of LDA and AVITM are 40% and 35%, respectively. Its error rate on the 20News dataset is only 12.52%, while the error rates of LDA and AVITM are 32.1% and 16.52%, respectively. In terms of accuracy, the sDAM-N model has an error rate of up to 90% on the IMDB dataset, while the error rates of LDA and AVITM are 70% and 60%, respectively. The error rate on the 20News dataset is as high as 88%, while the error rates of LDA and AVITM are 75.2% and 83.45%, respectively. From the F1 score, the sDAM-N model has an F1 score of 85.1 on the IMDB dataset and 89 on the 20News dataset, both higher than the LDA and AVITM models. Overall, in terms of model accuracy, the sDAM-N model designed for research has the highest precision, superior overall performance in processing text data, and better application effects in scenarios that require balancing accuracy and recall.

The results of five-fold cross validation show that the sDAM-N model exhibits relatively stable error rates under different data segmentation, indicating that the model's performance in handling different datasets is reliable. Among them, on the 20News dataset, the average error rate of the sDAM-N model is 16.10%, and the data fluctuation between folds does not exceed 0.30%, indicating that the model has strong generalization ability. On the RCV1 dataset, the average error rate of the sDAM-N model is 16.62%, which is similar to the data on the 20News dataset, indicating the stable performance of the model on diverse datasets. On the IMDB dataset, the average error rate of the model is 11.62%. Furthermore, the difference between folds is not significant, indicating that the model has an excellent capacity for processing high-dimensional sparse text data and exhibits high accuracy. Overall, the sDAM-N model is stable when facing different types of data.

The importance of FIM and hierarchical learning structure in the sDAM-N model can be seen through ablation research. Specifically, the removal of the hierarchical learning structure results in an increase in the error rate of the sDAM-N model from 16.52% to 20.40%. This indicates that the hierarchical learning structure facilitates the extraction of data features in a layered manner, whereby each layer of the network can capture the diverse hierarchical features of the data more

Table 7: Progressive testing of dataset size.

Dataset size (number of samples)	sDAM-N		LDA		AVITM	
	Computing time (s)	Memory usage (MB)	Computing time (s)	Memory usage (MB)	Computing time (s)	Memory usage (MB)
1000	0.02	50	0.03	60	0.05	70
5000	0.05	100	0.1	150	0.15	200
10000	0.12	200	0.25	300	0.3	400

20000	0.25	400	0.6	600	0.6	800
50000	0.6	800	1.5	1200	1.2	1600

comprehensively, thus ensuring optimal model performance. After removing the FIM, the error rate of the model increases from 16.52% to 18.8%. The FIM can effectively adjust the update speed of parameters at different levels by providing adaptive learning. This module helps to improve the convergence performance and robustness of the model. The lack of this module can lead to problems such as insufficient convergence speed of the model. After removing both the FIM and the hierarchical learning structure, the error rate of the sDAM-N model increases to 25%. Therefore, the improved structure is necessary for the sDAM-N model, as it helps improve the model's parameter optimization and feature capture capabilities.

Scalability analysis shows that the sDAM-N model performs well in handling large-scale data and can effectively cope with the increasing amount of data. From 1,000 samples to 50,000 samples, the computation time increases from 0.020s to 0.600s, with a relatively moderate growth rate. Compared with LDA and AVITM models, it has the advantage of computation time, indicating that it still performs stably when processing large-scale data. In terms of memory usage, from 1,000 samples to 50,000 samples, the computation time of the sDAM-N model has increased from 50MB to 800, which still has significant advantages compared to the LDA and AVITM models. This indicates that the sDAM-N model is still efficient in resource utilization in large-scale data environments. Overall, the sDAM-N model still has good efficiency and stability when facing large-scale datasets, and has promising application prospects.

4 Discussion

The designed sDAM model has shown significant superiority on multiple datasets. The error rates of sDAM on the 20News, RCV1, and IMDB datasets are 16.520%, 14.230%, and 11.670%, respectively, significantly lower than LDA's 24.850%, 25.120%, and 22.370%, and DocNADE's 24.010%, 21.500%, and 18.320%, respectively. This is attributed to the feature learning mechanism and multi-level structure adopted by the sDAM model, which can effectively extract potential relationships from text data. In terms of computation time, sDAM has test times of 0.020 seconds, 0.017 seconds, and 0.015 seconds on the 20News, RCV1, and IMDB datasets, respectively. In comparison, LDA has test times of 0.730 seconds, 0.310 seconds, and 0.560 seconds, while DLDA has longer test times. This is because sDAM effectively utilizes the FIM to dynamically adjust the learning rate and accelerate the model's adaptation speed during training. In terms of perplexity, the value of sDAM is 590.23, significantly lower than LDA's 887.32 and DocNADE's 606.83, indicating the information extraction and topic construction capabilities of the sDAM model.

Overall, the testing period, perplexity, and error rate of the sDAM model have advantages, indicating that the overall efficiency of the sDAM model is higher. This is based on the research using PGBN and FIM. The PGBN serves as the decoder for the model, which can help the model capture the hierarchical implicit features of text data. It utilizes recursive modeling to perform high-dimensional sparse processing on the data, enhancing modeling accuracy, and ensuring the normalization of topic weights. This reduces interference between topic features, simplifies the inference process, and improves inference efficiency. The FIM can adaptively adjust the learning rate in gradient calculation, allowing different levels and topic parameters to be updated at the optimal step size. By dynamically adjusting the learning step size of each layer, the convergence speed and accuracy of the model can be improved.

In practical applications, sDAM models may encounter difficulties when confronted with data sparsity issues, particularly when dealing with highly specialized textual data such as that found in the medical or legal domains. Due to the scarcity of data, the model may not be able to capture sufficient information, which can have a certain impact on the model's generalization ability. Nevertheless, the sDAM model retains considerable potential for application across a range of fields, including sentiment analysis and spam detection. In particular, it can be utilized in the context of social customer feedback, media analysis, and market research. The sDAM model can effectively extract and analyze potential themes in data information, helping businesses gain insights into consumer needs.

5 Conclusion

In light of the challenges posed by the exponential growth in text data, this paper presented a novel approach to data processing, namely the design of an sDAM that integrates a DL algorithm with a PSM model. This integration enabled the extraction of features from text data and label information. The results showed that the research model exhibited the lowest Perplexity under the 128-64-32 network structure, with values of 590.23, 953.12, and 982.67, which were significantly lower than other models. The testing times were 0.21s, 0.72s, and 0.85s, all of which were the lowest. The results of the application to the handwritten dataset indicated that the research model

produced a good interpolation effect on the dataset's hidden space, manifesting as smooth graphical changes. This suggested that the model's inference effect was relatively smooth and interpretable. From a horizontal comparison, the testing error rates of the research model on the 20News, RCV1, and IMDB datasets were 16.52%, 14.23%, and 11.67%, respectively, which were the lowest compared to other models. The testing time of the model on different datasets was the shortest, at 0.020 s, 0.017 s, and 0.015 s. Overall, the research model had the highest computational efficiency and accuracy, and was more stable and performed better than other models under changes in the dataset. Although the research model has effectively addressed the challenges of large-scale data processing, its limitation lies in the lack of feasibility validation when dealing with specific data categories. Therefore, further analysis of special data is the future direction.

References

[1] X. Li, C. Li, M. M. Rahaman, H. Sun, H. Li, J. Wu, Y. Yao, and M. Grzegorzek, "A comprehensive review of computer-aided whole-slide image analysis: From datasets to feature extraction, segmentation, classification and detection approaches," Artificial Intelligence Review, vol. 55, pp. 4809–4878, 2022. https://doi.org/10.1007/s10462-021-10121-0

[2] A. A. Barbhuiya, R. K. Karsh, and R. Jain, "CNN based feature extraction and classification for sign language," Multimedia Tools and Applications, vol. 80, pp. 3051–3069, 2021. https://doi.org/10.1007/s11042-020-09829-y

[3] S. K. S. Al-doorı, Y. S. Taspınar, and M. Koklu, "Distracted driving detection with machine learning methods by cnn based feature extraction," International Journal of Applied Mathematics Electronics and Computers, vol. 9, no. 4, pp. 116–121, 2021. https://doi.org/10.18100/ijamec.1035749

[4] R. Taiwo, T. Zayed, and M. E. A. B. Seghier, "Integrated intelligent models for predicting water pipe failure probability," Alexandria Engineering Journal, vol. 86, pp. 243–257, 2024. https://doi.org/10.1016/j.aej.2023.11.047

[5] K. Sharifani, and M. Amini, "Machine learning and deep learning: A review of methods and applications," World Information Technology and Engineering Journal, vol. 10, pp. 3897–3904, 2023.

[6] G. Menghani, "Efficient deep learning: A survey on making deep learning models smaller, faster, and better," ACM Computing Surveys, vol. 55, no. 12, pp. 1–37, 2023. https://doi.org/10.1145/3578938

[7] K. Yonekura, and K. Suzuki, "Data-driven design exploration method using conditional variational autoencoder for airfoil design," Structural and Multidisciplinary Optimization, vol. 64, pp. 613–624, 2021. https://doi.org/10.1007/s00158-021-02851-0

[8] J. Bai, W. Wang, and C. P. Gomes, "Contrastively disentangled sequential variational autoencoder," Advances in Neural Information Processing Systems, pp. 10105–10118, 2021. https://doi.org/10.48550/arXiv.2110.12091

[9] V. T. R. P. Kumar, M. Arulselvi, and K. B. S. Sastry, "Comparative assessment of colon cancer classification using diverse deep learning approaches," Journal of Data Science and Intelligent Systems, vol. 1, no. 3, pp. 128–135, 2023. https://doi.org/10.47852/bonviewJDSIS32021193

[10] R. F. Mansour, J. Escorcia-Gutierrez, M. Gamarra, D. Gupta, O. Castillo, and S. Kumar, "Unsupervised deep learning based variational autoencoder model for COVID-19 diagnosis and classification," Pattern Recognition Letters, vol. 151, pp. 267–274, 2021. https://doi.org/10.1016/j.patrec.2021.08.018

[11] L. Pinheiro Cinelli, M. Araújo Marins, E. A. Barros da Silva, and S. L. Netto, "Variational autoencoder, variational methods for machine learning with applications to deep networks," Springer, pp. 111–149, 2021. https://doi.org/10.1007/978-3-030-70679-1

[12] Y. Duan, L. Wang, Q. Zhang, and J. Li, "Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 4, pp. 4468–4476, 2022. https://doi.org/10.1609/aaai.v36i4.20369

[13] P. C. Collins, and D. G. Harlow, "Probability and statistical modeling: Ti-6Al-4V produced via directed energy deposition," Journal of Materials Engineering and Performance, vol. 30, pp. 6905–6912, 2021. https://doi.org/10.1007/s11665-021-06062-y

[14] A. Petersen, C. Zhang, and P. Kokoszka, "Modeling probability density functions as data objects," Econometrics and Statistics, vol. 21, pp. 159-178, 2022. https://doi.org/10.1016/j.ecosta.2021.04.004

[15] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," Neurocomputing, vol. 470, pp. 443–456, 2022. https://doi.org/10.1016/j.neucom.2021.05.103

[16] K. Atz, F. Grisoni, and G. Schneider, "Geometric deep learning on molecular representations," Nature Machine Intelligence, vol. 3, pp. 1023–1032, 2021. https://doi.org/10.1038/s42256-021-00418-8

[17] S. Yong, and Z. Linzi, "Robust deep auto-encoding network for real-time anomaly detection at nuclear power plants," Process Safety and Environmental Protection, vol. 163, pp. 438–452, 2022.

https://doi.org/10.1016/j.psep.2022.05.039

[18] C. Ricciardi, A. S. Valente, K. Edmund, and M. Cesarelli, "Linear discriminant analysis and principal component analysis to predict coronary artery disease," Health Informatics Journal, vol. 26, no. 3, pp. 2181–2192, 2020. https://doi.org/10.1177/1460458219899210

[19] S. Winther, S. E. Schmidt, L. D. Rasmussen, L. E. J. Orozco, F. H. Steffensen, H. E. Bøtker, J. Knuuti, and M. Bøttcher, "Validation of the european society of cardiology pre-test probability model for obstructive coronary artery disease," European Heart Journal, vol. 42, no. 14, pp. 1401–1411, 2021. https://doi.org/10.1093/eurheartj/ehaa755

[20] H. S. Bakouch, A. S. Nik, A. Asgharzadeh, and H. S. Salinas, "A flexible probability model for proportion data: Unit-half-normal distribution," Communications in Statistics: Case Studies, Data Analysis and Applications, vol. 7, pp. 271–288, 2021. https://doi.org/10.1080/23737484.2021.1882355

[21] K. Sharifani, and M. Amini, "Machine learning and deep learning: A review of methods and applications," World Information Technology and Engineering Journal, vol. 10, pp. 3897–3904, 2023.

[22] A. Mohammed, and R. Kora, "A comprehensive review on ensemble deep learning: Opportunities and challenges," Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 2, pp. 757–774, 2023. https://doi.org/10.1016/j.jksuci.2023.01.014

[23] P. Notin, J. M. Hernández-Lobato, and Y. Gal, "Improving black-box optimization in VAE latent space using decoder uncertainty," Advances in Neural Information Processing Systems, NeurIPS, pp. 802–814, 2021. https://doi.org/10.48550/arXiv.2107.00096

[24] C. Liu, R. Antypenko, I. Sushko, and O. Zakharchenko, "Intrusion detection system after data augmentation schemes based on the VAE and CVAE," IEEE Transactions on Reliability, vol. 71, no. 2, pp. 1000–1010, 2022. https://doi.org/10.1109/TR.2022.3164877