# Anomaly Detection Using LSTM-Based Variational Autoencoder in Unsupervised Data in Power Grid

Dibyajyoti Guha, *Member, IEEE*, Rajdeep Chatterjee, *Member, IEEE*, and Biplab Sikdar, *Senior Member, IEEE*

***Abstract*—This article proposes a deep generative model for anomaly detection in unsupervised power grid data. One-class classifier-based methods suffer from performance degradation when training data contain anomalous samples. Due to the temporal characteristics in most of the power grid datasets, we explore a long short-term memory-variational autoencoder-based deep generative model that can tolerate the moderate presence of anomalous data during training instead of standard data. This work demonstrates the advantage of reconstruction-based methods over clustering-based methods. As part of the reparameterization of the latent layer, a method is proposed by employing wavelet decomposition of the wavelet coefficients found from the high and medium frequency representations of the input time-series data. For further improvement, we have incorporated a $\log\cosh$-based cost function instead of the traditional consideration of the $L_2$ norm-based cost function. The numerical results demonstrate an improvement of performance metrics, such as AUC by 0.1–0.2 of our method over other benchmark methods. The transient stability threshold ($\delta$) is an important system parameter in the performance assessment of power grid systems. Through time domain simulations, it has been shown that $\delta = 0.3$ obtains optimal accuracy for transient stability assessment in the IEEE NewEngland-39 bus.**

***Index Terms*—Autoencoder (AE), deep learning, long short-term memory (LSTM), power grid, recurrent neural networks (RNN), smart grid, variational autoencoder (VAE), variational Bayes.**

## I. Introduction

REAL-TIME anomaly detection is essential for smart grid and internet of things (IoT)-driven manufacturing plants. Anomalies in the power system are deviations from expected measurements resulting from grid faults, such as load fluctuations and system oscillations. In the last few years, machine learning has started to play a critical role in bad data detection, transient stability prediction for various kinds of microgrid faults. For example, one-class support vector machine (OCSVM), random forest, and artificial neural networks-based methods have been used in [1]. The one-class classification technique, such as OCSVM and support vector data description (SVDD), models the underlying data by building a hypersphere of minimum volume to enclose the majority of the data in the feature space. This hypersphere describes the normality profile.

The availability of labeled data remains a crucial challenge for anomaly detection in a power grid, where high-dimensional data are generated in large volumes. Various methods have been proposed in literature for unsupervised anomaly detection. These can be grouped into two approaches: clustering-based and reconstruction-based unsupervised anomaly detections. Some of the most popular techniques among the clustering-based methods are OCSVM, K-means, and Gaussian mixture modeling (GMM). The high computational complexity of distance computation in large-dimensional datasets makes the clustering technique less attractive. Reconstruction-based approaches rely on dimension reduction through feedforward neural networks. It assumes that anomalies are incompressible and remain challenging to reconstruct from reduced dimension latent space [2]. In addition, various researchers have applied fast Fourier transform and discrete wavelet transform (DWT)-based feature engineering to facilitate unsupervised learning [3].

One of the difficulties in one-class classification methods, such as OCSVM, is that they can only handle a tiny percentage of anomalies in the training data. The performance of OCSVM degrades with the increase in the size of a fraction of anomalies in training data [4]. Similarly, reconstruction-based methods, such as autoencoders (AEs), assume that only standard patterns are being used for training purposes, thereby making the AE to reconstruct the normal data points well and failing to reconstruct anomalous data as it has never seen them during training. Various researchers have studied data-driven learning of hyperparameters of OCSVM, random forest to overcome this limitation [5].

The preparation of labeled data in a large dataset for anomaly detection is an expensive task, which demands domain knowledge expertise and human intervention. In most of the existing unsupervised learning methods, it is assumed that training data are free from anomalous instances. The motivation of this article comes from a need to have tolerance for the presence of a relatively moderate fraction of anomalous data and not to be very restrictive about a low fraction of anomalous data during the training process as desired by unsupervised learning. The relevance of our work is that while allowing the training data to tolerate moderate presence of anomalous samples, the proposed method can achieve high accuracy, AUC, and low

Manuscript received 17 July 2022; revised 7 February 2023; accepted 4 April 2023. Date of publication 27 April 2023; date of current version 30 August 2023. *(Corresponding author: Dibyajyoti Guha.)*

Dibyajyoti Guha is with the Department of Computer Science and Engineering, GITAM School of Technology, Gandhi Institute of Technology and Management (GITAM) Deemed to be University, Hyderabad 502329, India (e-mail: dibya.guha@gmail.com).

Rajdeep Chatterjee is with the School of Computer Engineering, Kalinga Institute of Industrial Technology (KIIT) Deemed to be University, Bhubaneswar 751024, India (e-mail: cse.rajdeep@gmail.com).

Biplab Sikdar is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: bsikdar@nus.edu.sg).

average response time (ART). There is "no single winner" among standalone anomaly detection methods that can detect all types of anomalies with high accuracy. We have chosen a generative model: recurrent neural network (RNN)-based variational autoencoder (VAE) for our anomaly detection method. Although VAE is a robust generative algorithm, the baseline VAE does not consider the temporal relation among data. Hence, we have incorporated long short-term memory (LSTM) with VAE. Many research papers have used the wavelet decomposition method for feature engineering purposes. We propose to use wavelet coefficients to make the latent space resilient to the presence of anomalous data during training.

### A. Contributions

In this article, we have utilized the temporal relationships in the data by combining VAE LSTM-based RNN for unsupervised anomaly detection. To enrich the reparameterization trick in VAE, we fetch independent samples from an a priori distribution of multivariate Gaussian distribution. The respective covariance matrix is learned from the singular value decomposition (SVD) of the wavelet coefficients found by multiresolution analysis. We use high and medium frequency representations of the signal employing Daubechies (db) wavelet decomposition. We choose high and medium frequency representations of the signal because we assume that anomalous samples occur rarely as compared with standard samples.

The important contributions of this article can be summarized as follows.

1) We propose an anomaly detection mechanism that can tolerate a certain amount of anomalous samples.
2) We demonstrate the effectiveness of generative models over clustering models in the context of anomaly detection for a sequential dataset.

The rest of this article is organized as follows. Section II reviews the related works and Section III elucidates the underlying principles of VAE. In Section IV, we present our proposed method to incorporate wavelet decomposition and log hyperbolic cosine ($\log\cos h$) function in the reconstruction loss function instead of the default choice of $L_2$ loss function. Section V provides a description of datasets used in this article, followed by Section VI that demonstrates the evaluation of results. Finally, we conclude this article in Section VII.

## II. Related Works

This section presents a literature survey on anomaly detection for power grid data with unsupervised learning. Clustering and reconstruction-based methods are widely used techniques in unsupervised learning. These techniques assume that standard data occurs much more frequently than abnormal data [6], [7]. The models are trained to predict the regular patterns of the data and to point out anomalies as they are different from the learned distribution.

### A. Clustering Based Methods

Clustering analyses, such as GMM, K-means, and density estimation, have been preferred by researchers as some of the most popular techniques for anomaly detection. In order to avoid the curse of dimensionality, the traditional methods adopt a two-step approach where the dimension is reduced at first by an AE or deep belief network, and subsequently, clustering analysis or density estimation is performed on the reduced space.

The input data are passed through a bottleneck, such as deep AEs or deep belief networks, before applying the one-class classifiers (OCC) to the data. Several hybrid mechanisms have been proposed to overcome the limitations of one-class classification methods. Cao et al. [8] proposed to use regularized AEs and VAEs in the first stage to force the standard data into a very tight space centered at the origin (that is, bottleneck unit). In the second stage, various density estimation-based clustering algorithms have been used for anomaly detection in the reduced space.

Khaledian et al. [9] proposed anomaly detection using an unsupervised stacked ensemble learning with isolation forest, local outlier probability, and K-means in IEEE 14 and 68-based bus systems phasor measurement units (PMUs). Khaledian et al. [9] used ensemble learning for the faster detection of anomalies and then classifying anomalies into bad data, event data, and PDC error. In a similar vein, Zhou et al. [10] presented an unsupervised ensemble learning approach for online anomaly detection for PMU data. The ensemble method invokes three classes of base detectors, notably, Chebyshev-based, DBSCAN-based, and regression-based detectors, are selected to generate anomaly scores of the PMU data.

### B. Reconstruction Based Methods

This article focuses on the AE-based reconstruction method. AEs can extract the common factors of variation from normal samples but fail to do the same for anomalous samples, as they have never seen the abnormal examples during training. The VAE is used to infer the latent embedding and reconstruct the input data in a variational manner by optimizing the variational lower bound. We limit our literature survey to probabilistic modeling and generative modeling techniques.

A modification of VAE ($\beta$-VAE), which is aimed at unsupervised disentangled representation learning, is proposed by Chen et al. [11]. $\beta$-VAE includes an additional hyperparameter in the VAE cost function, squeezing the latent bottleneck and boosting the latent representation's factorization. A latent representation of input multivariate time series data by incorporating temporal dependence among stochastic variables is proposed by Su et al. [12], which is named as OmniAnomaly.

A fault detection mechanism for inverter-interfaced distributed generation (IIDG)-enabled microgrid based on DWT and gated recurrent unit (GRU)-based RNN is proposed by James et al. [13]. The branch current magnitudes of three phases in one cycle is considered as input in [13]. DWT-based feature engineering is performed before feeding the data as input to the GRU in [13]. Symbolic dynamic filtering-based feature extraction technique for cyber-attack detection in large-scale smart grids has been analyzed by Karimipour et al. [14]. Taking a different approach, Karimipour et al. [14] implemented a dynamic Bayesian network for revealing the causal relationship between the extracted symbolic features and the restricted Boltzmann

machine for capturing the distribution of regular operation of the grid system.

Another approach for detecting anomalies in power systems is to estimate the state by learning invariant relationships between system components. One of the most popularly used techniques is Kalman filter-based autoregressive state estimation. To this end, Muralidhar et al. [15] analyzed Kalman autoregressive state estimation. This method is extended to be used as structure aware invariant learning by leveraging the network topology of cyber-physical system given beforehand. The other method is to investigate the probabilistic modeling of heavy-tailed distribution to find anomalies since the anomalous data appear in the tails of the probability distributions [16]. Copula functions are well known for their ability to model the heavy-tailed distributions.

### C. Alternative to $L_2$-Based Cost Function

In the recent past, there has been a constant effort to explore the effectiveness of other cost functions instead of the $L_2$ norm in the context of VAEs. In this direction, Xu et al. [17] proposed log hyperbolic cosine ($\log \cosh$)-based cost function for intrusion detection using conditional VAE. Similarly, Zhao et al. [18] discussed the choice of $L_2$ norm in the context of image restoration. They evaluated several loss functions to observe the impact of perceptually motivated metrics: the structural similarity index (SSIM) and the multiscale structural similarity index (MS-SSIM). It has been shown empirically in [18] that the poor performance happens due to the local minima of the loss functions. At the same time, the insufficiency of SSIM and MS-SSIM is demonstrated in [18]. The motivating factor for a search of an alternate loss function is that the $L_2$ norm penalizes heavily for large reconstruction errors and lightly for small reconstruction errors. Most of the research works have been reported in the context of image restoration, denoising, deblurring, and demosaicking. In [17], $\log \cosh$-based cost function has been applied in the context of intrusion detection with conditional VAE, which is trained with class label information. Our research work explores the effectiveness of the $\log \cosh$-based cost function in unsupervised learning in the context of anomaly detection using VAE.

Although deep generative models, such as VAE, have produced promising results in anomaly detection for multidimensional time series, existing research has generally assumed that the training samples are free from anomalous data. This motivates us to carry out a study to accommodate the presence of a relatively moderate fraction of anomalous data and not to be very restrictive about a low fraction of anomalous data during the training process as desired by unsupervised learning.

## III. Methodology

In this section, we present an overview of the methodology of LSTM and VAE in a nutshell. In this article, synchrophasor measurement techniques, which record voltage magnitude, voltage angle, and voltage frequency for all the buses are used as input data. As a result, it becomes pertinent to use RNN for analyzing the multidimensional time series data.

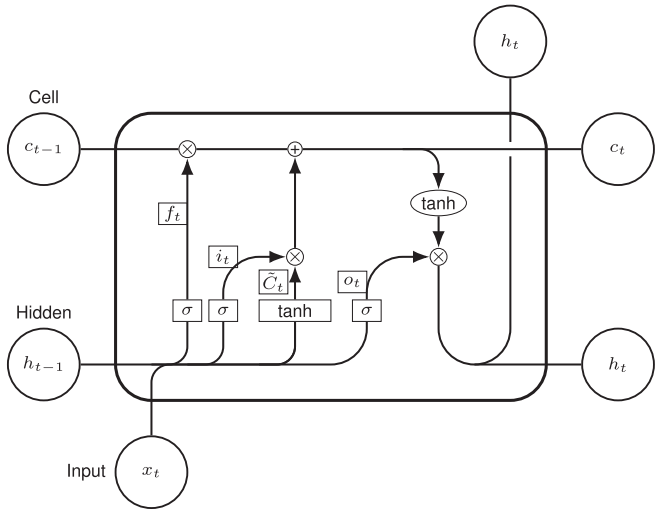


Fig. 1. Structure of an LSTM cell.

### A. Long Short-Term Memory

LSTM is a class of neural network that belongs to the family of RNN. It takes a temporal sequence of vectors $[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T]$ as input, and outputs a sequence of vectors $[\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T]$. An LSTM unit is comprised of three " gates": input, forget, and output, and a cell state in addition to a hidden state. An LSTM unit contains the weight parameters $\mathbf{W}^{x*}$, where $*$ denotes the symbol for one of the four matrices of the LSTM units ($\mathbf{f}$ for forget, $\mathbf{i}, \mathbf{g}$ for input, and $\mathbf{o}$ for output), the LSTM unit updates its cell state $\mathbf{C}_t$ according to the following formulation:

$$\mathbf{f}_t = \sigma\big(\mathbf{W}^{xf}\mathbf{x}_t + b^{xf} + \mathbf{W}^{hf}\mathbf{h}_{t-1} + b^{hf}\big)$$
$$\mathbf{i}_t = \sigma\big(\mathbf{W}^{xi}\mathbf{x}_t + b^{xi} + \mathbf{W}^{hi}\mathbf{h}_{t-1} + b^{hi}\big)$$
$$\mathbf{g}_t = \tanh\big(\mathbf{W}^{xg}\mathbf{x}_t + b^{xg} + \mathbf{W}^{hg}\mathbf{h}_{t-1} + b^{hg}\big)$$
$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$

and updates the output representation as

$$\mathbf{o}_t = \sigma\big(\mathbf{W}^{xo}\mathbf{x}_t + b^{xo} + \mathbf{W}^{ho}\mathbf{h}_{t-1} + b^{ho}\big)$$
$$\mathbf{h}_t = \tanh(\mathbf{C}_t) \odot o_t$$

where $\odot$ represents the elementwise product. The structure of an LSTM unit is shown in Fig. 1. One may refer to [19] for more details about LSTM.

GRU is another class of neural networks in the family of RNN. Unlike LSTM, GRU has only two gates, namely, update gate and reset gate. GRU has less training parameters, requires less memory, and executes faster than LSTM. There are two factors that dominate the comparison between LSTM and GRU: dataset size and length of time step. Comparison between GRU and LSTM in power grid datasets is an unexplored area. Yang et al. [20] conducted a performance comparison between LSTM and GRU in Yelp dataset. It was concluded by Yang et al. [20] that under long time step, GRU is more suitable for smaller

datasets than LSTM. We have chosen a small time step since we are interested in finding anomalies in the small-signal transient stability dataset. Hence, we have chosen LSTM over GRU as the RNN.

### B. Variational AE

VAE is a probabilistic graphical model that connects deep learning with variational inference. It aims to infer and learn a continuous latent variable $\boldsymbol{z}$ whose parameters have intractable posterior distributions. VAE is a directed probabilistic graphical model whose posterior is approximated by a neural network $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. The decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ represents the complex process of generating data $\boldsymbol{x}$ from the encoder's output, which is also modeled in the structure of a neural network. The objective function of a VAE is the variational lower bound of the marginal likelihood of data since the marginal likelihood is intractable. Interested readers may refer to [21] for derivation of VAE.

The marginal likelihood can be written as follows:

$$\log p_\theta(\boldsymbol{x}) = D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) + \mathcal{L}(\theta, \phi; \boldsymbol{x}). \quad (1)$$

The first right-hand side (RHS) term in (1) is the KL-divergence of the approximate from the true posterior. Since this KL-divergence is nonnegative, the second RHS term $\mathcal{L}(\theta, \phi; x^{(i)})$ is called the (variational) lower bound on the marginal likelihood of data point $\boldsymbol{x}$, and can be written as

$$\begin{aligned}\log p_\theta(\boldsymbol{x}) &\geq \mathcal{L}(\theta, \phi; \boldsymbol{x}) \\ &\geq -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{q_\phi(\boldsymbol{z}|\boldsymbol{x})}{p_\theta(\boldsymbol{z})}\right] + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \\ &\geq -D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z})\end{aligned} \quad (2)$$

where $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is the likelihood of the data $\boldsymbol{x}$ given the latent variable $\boldsymbol{z}$. The first term of (2) is the KL-divergence between the approximate posterior and the prior of the latent variable $\boldsymbol{z}$. This term forces the posterior distribution to be similar to the prior distribution, working as a regularization term. The second term of (2) can be understood in terms of the reconstruction of $\boldsymbol{x}$ through the posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and the likelihood $p_\theta(\boldsymbol{x}|\boldsymbol{z})$.

The approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ can be viewed as an encoder $f(\boldsymbol{x}, \boldsymbol{\phi})$ and the directed probabilistic graphical model $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ can be viewed as a decoder, from the perspective of AE. One of the most important features is that the VAE models the parameters of the distribution rather than the value itself. It implies that $f(\boldsymbol{x}, \boldsymbol{\phi})$ in the encoder outputs the parameter of the approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and to obtain the actual value of the latent variable $\boldsymbol{z}$, sampling from $q_\phi(\boldsymbol{z}; f(\boldsymbol{x}, \boldsymbol{\phi}))$ is required. Thus, the encoders and decoders of VAE can be called as probabilistic encoders and decoders with the complex relationship between the data $\boldsymbol{x}$ and the latent variable $\boldsymbol{z}$ represented by a neural network. The solution lies on differentiation and optimization of the lower bound $\mathcal{L}(\theta, \phi; x^{(i)})$ w.r.t. both the variational parameters $\phi$ and generative parameters $\theta$. The gradient of the lower bound w.r.t. $\phi$ is a bit problematic. To overcome this, Kingma and Welling [21] proposed a reparameterization trick that involves generating samples $z \sim q_\phi(\boldsymbol{z}|\boldsymbol{x})$. It is then often possible to express the random variable $\boldsymbol{z}$ as a deterministic variable $\boldsymbol{z} = g_\phi(\boldsymbol{\epsilon}, \boldsymbol{x})$, where $\boldsymbol{\epsilon}$ is an auxiliary variable with independent marginal $p(\boldsymbol{\epsilon})$, and $g_\phi(.)$ is some vector-valued function parameterized by $\phi$.

For example, in the univariate Gaussian case, let $z \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\mu, \sigma^2)$. In this case, a valid reparameterization is $z = \mu + \sigma \cdot \epsilon$, where $\epsilon$ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$. Therefore

$$\begin{aligned}\mathbb{E}_{\mathcal{N}(z;\mu,\sigma^2)}[f(z)] &= \mathbb{E}_{\mathcal{N}(\epsilon;0,1)}[f(\mu + \sigma \cdot \epsilon)] \\ &\simeq \frac{1}{L}\sum_{l=1}^{L} f(\mu + \sigma \cdot \epsilon^{(l)}) \text{ where } \epsilon^{(l)} \sim \mathcal{N}(0, 1).\end{aligned} \quad (3)$$

## IV. Proposed Method

In this article, we assume that the latent layer can be made resilient to the presence of anomalous data during training by forcing the covariance of latent variable to be guided by the high and medium frequency wavelet coefficients of the training data. Our assumption is in-line with the research work of [16], which assumes that the anomalous data appears in the tails of the probability distributions as they occur rarely. In this section, we propose a method to enhance reparameterization and reconstruction loss function instead of following the usual choice in the VAE.

### A. Reparameterization Trick

The heart of the VAE lies in the reparameterization of the latent space representation as given in (3). Since the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, in this case, is intractable, Kingma and Welling [21] demonstrated that the prior over the latent variables can be considered to be a centered isotropic multivariate Gaussian $p_\theta(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$, resulting into the output of the encoder network as

$$\log q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)}) = \log \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu_{(i)}}, \boldsymbol{\sigma^{2(i)}}\boldsymbol{I}) \quad (4)$$

where the mean and standard deviation (s.d.) of the approximate posterior, $\boldsymbol{\mu}_{(i)}$ and $\boldsymbol{\sigma}^{(i)}$, are outputs of the encoding neural network. A diagram of the architecture of modified VAE with feedforward neural network and LSTM is shown in Figs. 2 and 3, respectively. The architecture drawn in Fig. 3 is an enhanced version of the architecture described in Fig. 2 by incorporating LSTM units instead of the fully connected neurons in the encoding and decoding layers.

In accordance with Kingma and Welling [21], the sample from posterior $\boldsymbol{z}^{(i,l)} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})$ can be obtained using $\boldsymbol{z}^{(i,l)} = g_\phi(\boldsymbol{x}^{(i)}, \boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{\epsilon}^{(\boldsymbol{l})}$, where $\boldsymbol{\epsilon}^{(\boldsymbol{l})} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. We propose to enhance the assumption of distribution of $\boldsymbol{\epsilon}^{(\boldsymbol{l})} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{A})$, where $\boldsymbol{A}$ is a diagonal matrix consisting of the two largest eigenvalues found in the SVD of the wavelet coefficients from the high and medium frequency representations of the training data. The following steps are applied on each of the

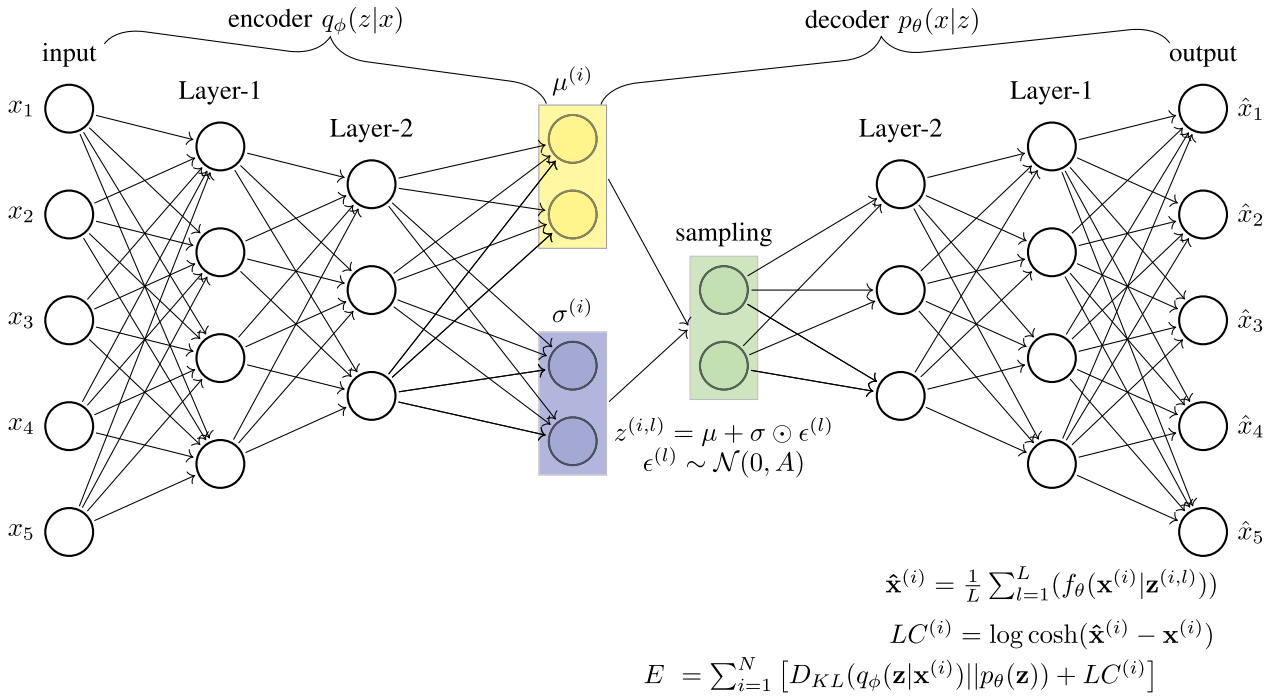


Fig. 2. MVAE with feedforward neural network.

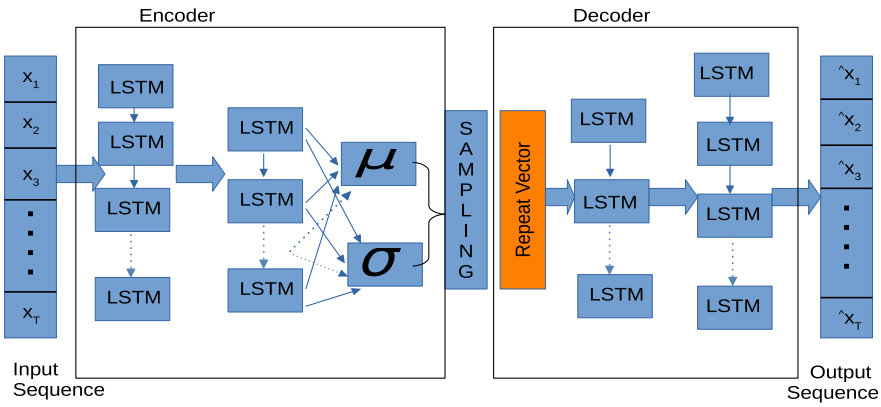


Fig. 3. LSTM with MVAE.

two principal components of the training data to construct the covariance matrix of the multivariate Gaussian distribution.

1) To obtain high and medium frequency wavelet coefficients, the db wavelet transformation up to the level of 4 is applied to the first principal component of the training dataset. The detail coefficients are stored as a column vector of a matrix. The lower level wavelet coefficients are padded with zero to maintain equal size for all the vectors in each column. Compute the average of the wavelet coefficient matrices generated from the first two principal components. Let us denote this resultant matrix as **B**.
2) Repeat the same process for the second principal component of the signal.
3) Develop a diagonal matrix **A** whose elements are the first two eigenvalues of the matrix **B**.

We have also incorporated the $\log\cosh$ function in the cost function, described in the following section. We have named this model LSTM modified variational autoencoder (MVAE). Thus, we use a different technique to find the approximate distribution in LSTM-MVAE, in contrast to the baseline VAE that uses an identity matrix. The algorithmic description of our method is given in Algorithms 1 and 2.

### *B. Reconstruction Loss Function*

The reconstruction loss function $\boldsymbol{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z})$ plays a pivotal role in all the deep generative models. It has become usual practice to use the $L_2$ loss as reconstruction loss as $p_\theta(\boldsymbol{x}|\boldsymbol{z}) \propto \exp(\|x - \hat{x}\|^2)$ while considering the samples from multivariate Gaussian distribution. The efficiency of $L_2$ loss

**Algorithm 1:** Training of LSTM-MVAE.

Input: $X = \{\boldsymbol{x}^{(1)} = \{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}, \ldots, \boldsymbol{y}^{(T)}\}, \boldsymbol{x}^{(2)} = \{\boldsymbol{y}^{(2)}, \boldsymbol{y}^{(3)}, \ldots, \boldsymbol{y}^{(T+1)}\}, \ldots, \boldsymbol{x}^{(N-T+1)} = \{\boldsymbol{y}^{(N-T+1)}, \boldsymbol{y}^{(N-T+2)}, \ldots, \boldsymbol{y}^{(N)}\}\}$, $X_{train} = 0.8 * X; X_{test} = 0.2 * X$
Output: probabilistic encoder and decoder
**repeat**
  **for all** $\boldsymbol{x^{(i)}} \in X_{train}$ **do**
    $\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)} = g_\phi(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})$
    Draw $L$ samples from $\boldsymbol{\epsilon^{(l)}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{A})$
    $\boldsymbol{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{\epsilon^{(l)}}$
  **end for**
  $\hat{\boldsymbol{x}}^{(i)} = \frac{1}{L}\sum_{l=1}^{L}(f_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)}))$
  $LC^{(i)} = \log\cosh(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{x}}^{(i)})$
  $\mathrm{E} = \sum_{i=1}^{N}[D_{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})||p_\theta(\boldsymbol{z})) + LC^{(i)}]$
  $\phi, \theta \leftarrow$ Update parameters using gradients of $E$
**until** convergence of parameters $\phi, \theta$
**return** $g_\phi, f_\theta$

**Algorithm 2:** Testing of LSTM-MVAE.

Input: Testing dataset $= X_{test}$, $\alpha$ : Using Chebyshev's inequality 99.1% confidence interval.
Output: reconstruction probability $p_\theta(\boldsymbol{x}|\hat{\boldsymbol{x}})$
**for all** $\boldsymbol{x^{(i)}} \in X_{test}$ **do**
  $\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)} = g_\phi(\boldsymbol{z}^{(i)}|\boldsymbol{x}^{(i)})$
  Draw $L$ samples from $\boldsymbol{\epsilon^{(l)}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{A})$
  $\boldsymbol{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \cdot \boldsymbol{\epsilon^{(l)}}$
**end for**
$\hat{\boldsymbol{x}}^{(i)} = \frac{1}{L}\sum_{l=1}^{L}(f_\theta(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,l)}))$
$rc[\boldsymbol{x}^{(i)}] = \log\cosh(\boldsymbol{x}^{(i)}, \hat{\boldsymbol{x}}^{(i)})$
**if** $((rc[\boldsymbol{x}^{(i)}] < \alpha))$ $\boldsymbol{x}^{(i)}$ is an anomaly
**else** $\boldsymbol{x}^{(i)}$ is not an anomaly

function has been investigated by Zhao et al. [18] and found to get confounded in local minima in the image processing context. In addition, the $L_2$ loss function is sensitive toward outliers. On the other hand, the $L_1$ loss function is not sensitive toward outliers. The necessity for $L_1$ loss function can arise if the decoder is chosen from a zero-mean Laplace distribution $p_\theta(\mathbf{x}|\mathbf{z}) \propto \exp(\|x - \hat{x}\|_1)$. But the $L_1$ loss function is disadvantageous as it is not differentiable for a data point with $x_i - \hat{x_i} = 0$. Zhao et al. [18] shown that $L_1$ loss function can outperform $L_2$ loss function in the context of image restoration. An interesting problem arising in this context is to find a suitable reconstruction loss function instead of the usual choice of $L_2$ loss for time series dataset in deep generative models, such as LSTM-based VAE.

In a nutshell, the contributions of our method are given as follows.

1) Enhancing the reparameterization trick for sampling latent variable in VAE. It works by employing SVD of the wavelet coefficients from the input time-series data's high and medium frequency representations.
2) Incorporating the log hyperbolic cosine ($\log\cosh$) function in the reconstruction loss function instead of the default choice of $L_2$ loss function.

## V. Datasets

In this section, we briefly describe the datasets that have been used in this article. Various experiments of our anomaly detection method have been carried out on different kinds of datasets, such as transient stability, PMU voltage measurement data, industrial control system (ICS) cyber-attack, and data integrity attack.

In order to prepare the training datasets for anomaly detection, we allow 5%–15% of the training data to contain anomalous samples. In the testing datasets, 50% of the data are anomalous. The labels of the dataset have been suppressed to the algorithm during training, while it has been used during the evaluation of our predictions. The experiments are implemented on a machine with Python 3.7, Intel Core i7 CPU at 3.6 GHz, and 8 GB RAM, with Keras and Tensorflow frameworks. The python codes are made available with private access at Github.[1] The following benchmark datasets have been used for our experiments.

### A. Transient Stability Prediction Datasets

In our work, we have chosen phasor-based simulator and hybrid-type simulator, which was investigated by Behdadnia and Parlak [22]. For simplicity, the dataset is termed as "TSAT_PMU" in the rest of this article. The outputs of the phasor-based simulator, including voltage magnitude, voltage angle, and voltage frequency, were named "Bus_Mag," "Bus_Ang," and "Bus_Freq," respectively. The dataset is a cell array where each cell has a single-precision array of dimension $39 \times 17$, where 39 indicates the number of buses equipped with PMUs, and 17 is the number of phasor values, including prefault, during-fault, and postfault data. It has a sample size of 5000. Interested readers may refer to [22] for further details of the phasor-based simulator.

### B. IEEE 118-Bus Transient Event Data

Transient stability assessment is very important for reliable power distribution. Transient data are used to simulate the PMU voltage measurements. In this article, we have chosen the transient event data of IEEE 118-bus, which is described in [23]. This dataset is termed as "TED_118" in the rest of this article. The data are collected from the TSAT software (DSATool) containing postfault voltage magnitude transient data simulation from the IEEE 118-bus system. Here, the training and testing samples are 8000 (50% stable and another 50% unstable) and 2000 (50% stable and another 50% unstable), 25 indicates the length of the data, which indicates the number of cycles of PMU voltage magnitudes after the clearance of the fault. The sampling rate of PMUs is 120.

[1] https://github.com/dguhanus/LSTM_MVAE

### C. ICS Cyber-Attack Datasets

The next dataset that has been considered is the cyber-attack dataset in ICS for the synchrophasor-based measurement data for broader area monitoring in a smart grid [24]. This dataset is referred to as "ICS_DB1" in the rest of this article. The synchrophasor measurement data include frequency, current phasors, voltage phasors, and sequence components. Different scenarios are simulated, such as power system disturbances, normal operations, and power system cyber-attacks are applied against the simulated power system and its components. However, we have used the binary classes for preparing the anomaly dataset.

### D. Data Integrity Attack in Cyber-Physical Systems

In order to verify the capacity to detect anomalies caused by data integrity attacks, we have chosen a false data injection attack. We have used the dataset from Goel and Swarup [25], wherein the IEEE 24 bus system is modeled in digisilent's power factory, and normal operating data points are generated under quasi-dynamic simulation and N-2 contingencies. Data integrity attacks are critical due to their difficulty in analyzing the normality profile. Hereafter, this dataset is referred to as "Integrity_Att" in the rest of this article.

### E. Hyperparameters of Proposed LSTM-MVAE

The encoder and decoder parts of LSTM-VAE are embedded with LSTM cells. The architecture of our proposed LSTM-MVAE is composed of two hidden layers and a latent layer, which is shown in Fig. 2, for an example of five dimensional input. The encoder path of LSTM-MVAE has 32 and 16 LSTM units in their hidden layers in succession, and two neurons for the latent layer. The decoder path of LSTM-MVAE has 16 and 32 LSTM units in their hidden layers in succession. The minibatch size is set to be 256, and fourfold cross validation, " relu" activation, and Adam optimizer [26] are used. The time step value for the LSTM is chosen to be 10. The method was run for 200 epochs with a learning rate of $1 \times 10^{-3}$ and early stopping enabled if the loss for the validation set observes no decrease for 50 epochs.

## VI. Discussion of Results

This section presents the detailed results for LSTM-MVAE for the abovementioned datasets and compares the performance against both the methods of generative and hybrid clustering-based techniques. As part of the reconstruction method, we have chosen various kinds of VAE, such as MMD-VAE, $\beta$-VAE, and OmniAnomaly. The construction of the hybrid clustering technique is composed of a combination of VAEs and multiple kinds of OCC [8]. Anomaly detection techniques use a threshold value that allows the user to control the tradeoff between false positive (FP) and false negative (FN). The performance of our model has been evaluated using the accuracy and the area under the resulting ROC curve (AUC). The performance metrics are reported by choosing the best threshold.

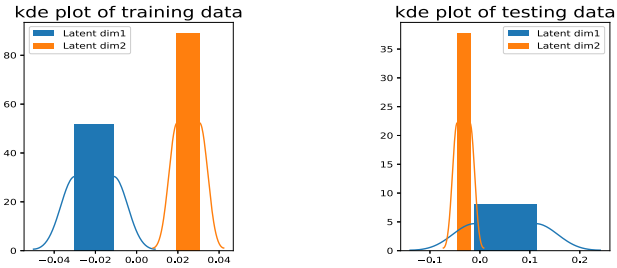


Fig. 4. Embedding of training data and testing in latent space for LSTM-MVAE in TSAT_PMU dataset.

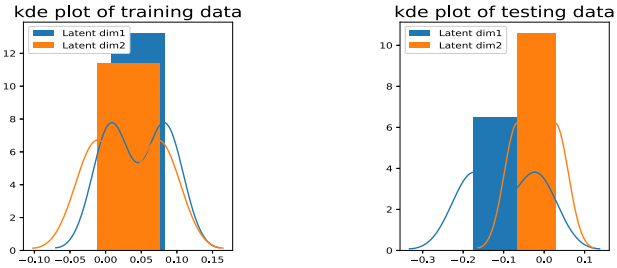


Fig. 5. Embedding of training data and testing in latent space for baseline LSTM-VAE in TSAT_PMU dataset.

### A. Comparison With LSTM-Baseline VAE

The histogram of latent layer encoding of training and testing data is shown in Figs. 4 and 5 for LSTM-MVAE and LSTM-VAE models, respectively, for the TSAT_PMU dataset. The data in Figs. 4 and 5 have been plotted by Kernel density estimation of "seaborn" packages. The $y$-axis values are produced by the internal algorithms of binning and "yticks" of "matplotlib," which makes it difficult to unify the scale of values in the $y$-axis. The data are plotted in 2-D latent space after being fed into the encoder part of the VAE. Fig. 4 shows that the separation of testing data in the latent space is more visible for LSTM-MVAE than for the LSTM-baseline VAE. These figures provide an intuitive explanation of the improved performance of LSTM-MVAE over baseline LSTM-VAE, as we can distinguish the two classes in the latent space whose mean are separated by some distance. This can be attributed as one of the causes behind the improved performance of LSTM-MVAE over baseline LSTM-VAE.

### B. Anomaly Detection With Various VAEs

In the first set of experiments, we compare our method with the state-of-the-art latent generative models, such as $\beta$-VAE, MMD-VAE, and OmniAnomaly, and the results are given in Table I. The AUC has been used to evaluate our method's predictive capacity. In the case of the IEEE 118-bus transient event dataset [23], it can be seen in Fig. 6 that our proposed model of LSTM-MVAE produces an improvement of 0.1–0.2 in AUC values, when compared with other latent generative models of MMD-VAE, $\beta$-VAE, and OmniAnomaly. The reconstruction fidelity of LSTM-MVAE is better than $\beta$-VAE as $\beta$-VAE introduces a tradeoff that punishes reconstruction quality for encouraging disentanglement within the latent representations.

TABLE I
PERFORMANCE COMPARISON WITH DIFFERENT KIND OF VAEs

| Method | Metric | datasets | | | |
|---|---|---|---|---|---|
| | | Integrity_Att | ICS DB1 | TSAT_PMU | TED_118 |
| MMD-VAE | Accuracy | 0.8 | 0.863 | 0.985 | 0.748 |
| | Precision | 0.97 | 0.985 | 0.984 | 0.956 |
| | F1 score | 0.76 | 0.926 | 0.991 | 0.853 |
| | AUC | 0.8 | 0.560 | 0.990 | 0.580 |
| $\beta$-VAE | Accuracy | 0.71 | 0.848 | 0.926 | 0.725 |
| | Precision | 0.9 | 0.954 | 0.960 | 0.903 |
| | F1 score | 0.66 | 0.900 | 0.934 | 0.827 |
| | AUC | 0.74 | 0.520 | 0.950 | 0.560 |
| OmniAnomaly | Accuracy | 0.72 | 0.875 | 0.989 | 0.832 |
| | Precision | 0.94 | 0.990 | 0.983 | 0.954 |
| | F1 score | 0.68 | 0.932 | 0.982 | 0.892 |
| | AUC | 0.77 | 0.670 | 0.990 | 0.700 |
| LSTM-MVAE | Accuracy | 0.73 | 0.880 | 0.990 | 0.854 |
| | Precision | 0.96 | 0.990 | 0.980 | 0.973 |
| | F1 score | 0.7 | 0.930 | 0.990 | 0.919 |
| | AUC | 0.79 | 0.680 | 0.990 | 0.720 |

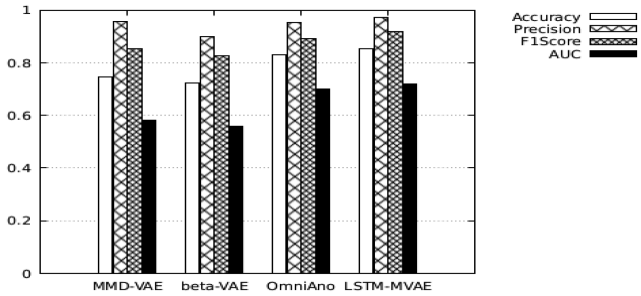


Fig. 6. Comparison with generative models for TED_118 dataset with 10% anomaly.

We have compared four different generative models across four different performance metrics: accuracy, precision, F1 score, and AUC. Hence, it is necessary to conduct statistical test, such as Wilcoxon test, to find the significance of the difference among the values produced in Table I. Among the chosen state-of-the-arts methods of MMD-VAE, $\beta$-VAE, and OmniAnomaly for anomaly detection, MMD-VAE and OmniAnomaly have more promising results. At first, we chose MMD-VAE and LSTM-MVAE for performing the Wilcoxon test. All the AUC values for the abovementioned datasets for MMD-VAE and LSTM-MVAE methods are collected as two different variables, and subsequently, the Wilcoxon test is performed between these two variables. The abovementioned test produced a $p$-value of 0.285. While performing the Wilcoxon test between the AUC values reported in Table I for LSTM-MVAE and OmniAnomaly, the $p$-value becomes 0.1. We have shown that the Wilcoxon test produces $p$-value $\geq 0.1$, which is sufficient for measuring significance of difference.

### C. Comparison With OCC Models in Various Kinds of VAEs

In the second set of experiments, we compare the performance of our model with hybrid models of VAE with various flavors of OCC by following Cao et al. [8], and the results are given in Table II. Interested readers may find the definition of clustering methods, such as local outlier factor (LOF), centroid (CEN), mean distance (MDIS), and kernel density estimation (KDE), variant of AEs, such as shrink autoencoder (SAE), and dirac delta variational autoencoder (DVAE) in [8]. The bottleneck layers of the trained DAE, VAE, SAE, and DVAE are used as latent representations for six OCC, LOF, CEN, MDIS, KDE, and OCSVM. Each column represents the AUCs reported by several classifiers.

It can be seen from Table II that CEN, MDIS, and KDE-based density estimation on the reduced latent space of various VAEs outperforms LOF and OCSVM. It can also be seen that hybrid model of DVAE and KDE produces better AUC than other AEs, such as VAE and DAE [8], in the presence of anomalous instances during training. This suggests that we can compare our proposed LSTM-MVAE performance with the hybrid model of DVAE-KDE. Wilcoxon test has been conducted between DVAE-KDE and LSTM-MVAE to find the significance of the difference among the values produced in Table II. All the AUC values for the abovementioned datasets for DVAE-KDE and LSTM-MVAE methods are collected as two different variables, and subsequently, the Wilcoxon test is performed between these two variables. The abovementioned test produced a $p$-value of 0.375. For the Wilcoxon test between OmniAnomaly and LSTM-MVAE, the $p$-value is 0.1. We have shown that the Wilcoxon test produces $p$-value $\geq 0.1$, which is sufficient for measuring significance of difference. The values in both Tables I and II suggest that our method not only achieves the performance of state-of-the-art algorithms but also produces an improvement of AUC values by 0.1–0.2.

### D. Comparison of Cost Functions for Logcosh and $L_2$ Loss

Next, we consider $\log\cosh$ function to understand its behavior relative to $L_1$ and $L_2$ functions. From the properties of $\log\cosh$ function in the following, we know that

$$f(x) = \log\cosh(x) = \log\frac{e^x + e^{-x}}{2\,s}. \tag{5}$$

As $x \to \infty$, the $\log\cosh$ function tends toward $\log\cosh(x) \approx \mid x \mid -\log(2)$. Therefore, when $x$ is far from 0, the function behaves like $L_1$ norm. At the same time, it behaves close to $L_2$ norm for small $\mid x \mid$. Thereby, $\log\cosh$ function

TABLE II
AUC COMPARISON AMONG HYBRID MODELS OF VAE AND OCC

| Methods | Classifiers | dataset | | | |
|---|---|---|---|---|---|
| | | Integrity_Att | ICS DB1 | TSAT_PMU | TED_118 |
| DAE | LOF | 0.64 | 0.51 | 0.71 | 0.61 |
| | CEN | 0.69 | 0.60 | 0.73 | 0.60 |
| | MDIS | 0.7 | 0.59 | 0.77 | 0.61 |
| | KDE | 0.72 | 0.61 | 0.82 | 0.64 |
| | $OCSVM5_{\nu=0.5}$ | 0.63 | 0.61 | 0.83 | 0.60 |
| | $OCSVM1_{\nu=0.1}$ | 0.61 | 0.57 | 0.83 | 0.58 |
| VAE | LOF | 0.67 | 0.51 | 0.82 | 0.62 |
| | CEN | 0.7 | 0.60 | 0.84 | 0.59 |
| | MDIS | 0.71 | 0.58 | 0.86 | 0.61 |
| | KDE | 0.72 | 0.62 | 0.88 | 0.64 |
| | $OCSVM5_{\nu=0.5}$ | 0.65 | 0.62 | 0.81 | 0.60 |
| | $OCSVM1_{\nu=0.1}$ | 0.62 | 0.59 | 0.80 | 0.56 |
| SAE | LOF | 0.65 | 0.53 | 0.92 | 0.72 |
| | CEN | 0.67 | 0.63 | 0.93 | 0.66 |
| | MDIS | 0.7 | 0.62 | 0.93 | 0.70 |
| | KDE | 0.72 | 0.66 | 0.94 | 0.73 |
| | $OCSVM5_{\nu=0.5}$ | 0.66 | 0.67 | 0.91 | 0.69 |
| | $OCSVM1_{\nu=0.1}$ | 0.61 | 0.63 | 0.90 | 0.67 |
| DVAE | LOF | 0.67 | 0.53 | 0.91 | 0.72 |
| | CEN | 0.7 | 0.64 | 0.92 | 0.64 |
| | MDIS | 0.72 | 0.62 | 0.94 | 0.70 |
| | KDE | 0.75 | 0.66 | 0.97 | 0.74 |
| | $OCSVM5_{\nu=0.5}$ | 0.67 | 0.68 | 0.85 | 0.66 |
| | $OCSVM1_{\nu=0.1}$ | 0.63 | 0.64 | 0.85 | 0.61 |
| OmniAnomaly | | 0.77 | 0.67 | 0.99 | 0.70 |
| LSTM-MVAE | | 0.79 | 0.68 | 0.99 | 0.72 |

TABLE III
LOG COSH VERSUS $L_2$ FOR RECONSTRUCTION LOSS FUNCTION FOR BOTH BASELINE LSTM-VAE AND LSTM-MVAE

| Method | Metric | Integrity_Att | | ICS_DB1 | | TSAT_PMU | | TED_118 | |
|---|---|---|---|---|---|---|---|---|---|
| | | log cosh | $L_2$ | log cosh | $L_2$ | log cosh | $L_2$ | log cosh | $L_2$ |
| LSTM-VAE | AUC | 0.76 | 0.72 | 0.64 | 0.62 | 0.98 | 0.98 | 0.71 | 0.68 |
| LSTM-MVAE | AUC | 0.79 | 0.74 | 0.75 | 0.66 | 0.99 | 0.98 | 0.72 | 0.7 |

is essentially a combination of both the $L_1$ and $L_2$ functions, and the derivative of $\log\cosh$ is the $\tanh$ function that makes the training of neural network more efficient. Our motivation to use $\log\cosh$-based cost function stems from the work of Ronchetti and Huber [27] who proposed to combine $L_1$ and $L_2$ by using $L_2$ in the vicinity of the origin and then invoking $L_1$ at a certain distance away from origin. To the best of our knowledge, $\log\cosh$-based cost function has not been used before in the context generative modeling in time series data.

The comparison between LSTM-VAE and LSTM-MVAE is given in Table III, which contains the AUC of the receiver operating characteristics curve. Furthermore, the performance comparison between $\log\cosh$-based cost function and $L_2$ norm-based cost function is given in Table III. It is evident that $\log\cosh$-based cost function produces better results than the $L_2$ norm-based cost function.

### *E. Comparison With Related Transient Assessment Methods*

In this section, we compare the performance of transient stability assessment of our method with that of James et al. [28]. For transient stability assessment, the generator rotor angle difference $\delta$ is used to determine whether any generator in the system is out of synchronism or not. This index is a widely used transient stability indicator for power grid systems [29]. By following the methodology as described by James et al. [28], a time domain simulation is conducted using TSAT. Different $\delta$ values in the range [0.05, 0.45] with a step size of 0.05 are simulated, and accuracy and ART are plotted. We refrain from comparing with CNN-LSTM-based models for the sake of maintaining similarity in model assumptions between compared methods.

The accuracy and ART of transient prediction is plotted across different values of $\delta$ for our proposed LSTM-MVAE and [28] in Figs. 7 and 8, respectively, for the IEEE New England-39 bus dataset. It can be seen from Fig. 7 that our proposed model provides better accuracy when $\delta$ is below 0.3 and it maintain higher accuracy compared with [28] when $\delta$ is between 0.4 and 0.45. However, LSTM-MVAE produces lesser accuracy in the region from 0.3 to 0.4. One may infer that the $\delta$ value 0.3 can be chosen for getting optimal performance for transient assessment accuracy. In Fig. 8, it can be seen that ART values are lower for LSTM-MVAE than that of [28]. The reduction of ART can be attributed by the reduced number of neurons in intermediate layers for LSTM-MVAE.

### *F. Robustness in the Presence of Anomalous Data*

After comparing the performance of anomaly detection with the state-of-the-art methods, we now turn our attention toward verifying the resilience of our proposed method while we assume that anomalous data are contained in the training data with

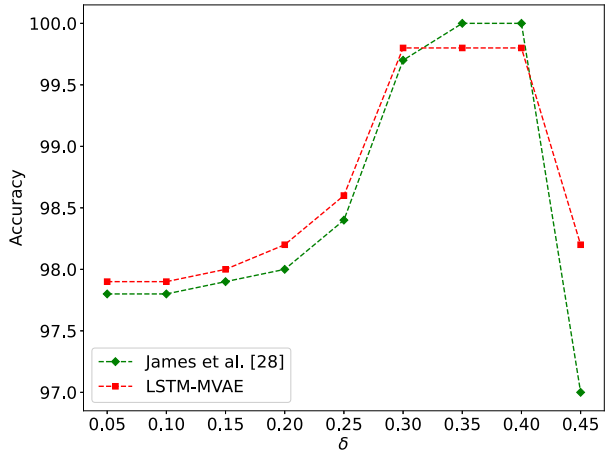


Fig. 7. Accuracy versus $\delta$ in IEEE New England-39 bus dataset.

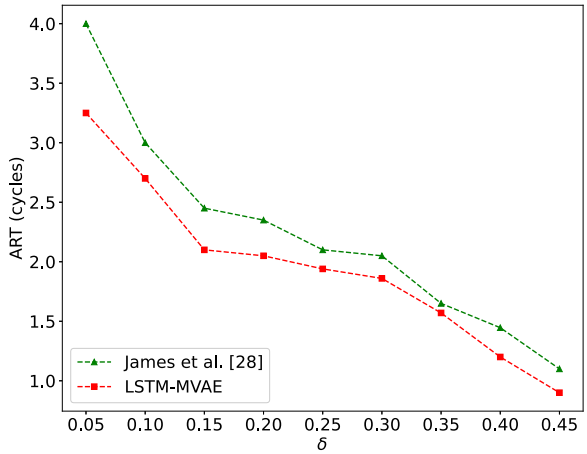


Fig. 8. ART versus $\delta$ in IEEE New England-39 bus dataset.

proportion of 5%–15%. The OmniAnomaly method [12] has been chosen for comparison purpose as OmniAnomaly outperforms the other methods of $\beta$-VAE, and MMD-VAE as given in Table I. Fig. 9 compares the AUC between LSTM-MVAE and OmniAnomaly for various fractions of anomalous data for the TED_118 dataset. The results suggest that LSTM-MVAE remains less sensitive to the presence of increased anomalous data during training. To highlight the strengths of LSTM-MVAE, the results are summarized as follows.

1) The separability of normal and anomaly classes of data in the latent layer is more clearly visible for our proposed model than the baseline LSTM-VAE.
2) By the help of Wilcoxon testing to verify the significance of the difference of our proposed method's performance metrics over the state-of-the-arts methods, as reported in Tables I and II, it has been shown that the $p$-value $\geq 0.1$.
3) The resilience of our proposed model is better than the state-of-the-art methods when anomalous data are contained up to 15% of the training data.
4) In the context of anomaly detection, it is demonstrated that $\log\cosh$-based loss function outperforms the $L_2$-based loss function.

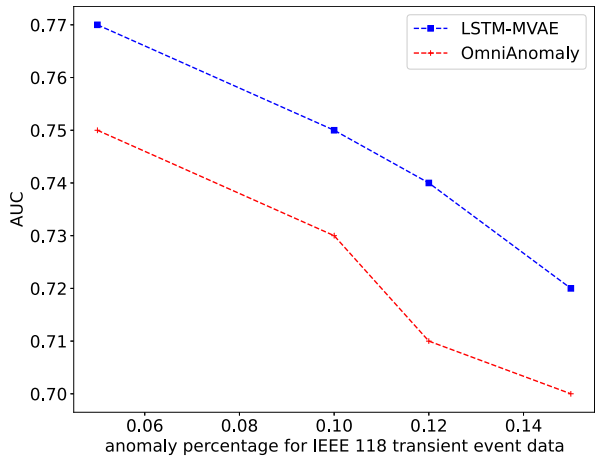


Fig. 9. Comparison of LSTM-MVAE and OmniAnomly with different percentage of anomaly in TED_118 dataset.

5) In the context of transient stability assessment, our proposed method produces better accuracy and ART for various values of transient stability index ($0.05 \leq \delta \leq 0.45$) when it is compared with [28]. Through time domain simulations, it has been shown that $\delta = 0.3$ obtains optimal accuracy for transient stability assessment in IEEE NewEngland-39 bus.

## VII. Conclusion

This article considered a generative model based on an LSTM-VAE that remains relatively insensitive to moderate presence of anomalous data during training, in contrast to the prevalent existing methods with absence of anomalous data during training. In this article, we introduced a method to enhance the reparameterization trick for sampling latent variables in VAE by employing singular value decomposition of the wavelet coefficients found from the input's high and medium frequency representation time-series data. At the same time, we also incorporated a $\log\cosh$-based cost function instead of the traditional use of the $L_2$ norm-based cost function. The numerical results demonstrate an improvement of performance metrics, such as AUC by 0.1–0.2 for our method over other benchmark methods.

## References

[1] E. Casagrande, W. L. Woon, H. H. Zeineldin, and D. Svetinovic, "A differential sequence component protection scheme for microgrids with inverter-based distributed generators," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 29–37, Jan. 2014.

[2] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2018.

[3] D. P. Mishra, S. R. Samantaray, and G. Joos, "A combined wavelet and data-mining based intelligent protection scheme for microgrid," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2295–2304, Sep. 2016.

[4] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proc. ACM SIGKDD Workshop Outlier Detection Description*, 2013, pp. 8–15.

[5] Z. Ghafoori, S. M. Erfani, S. Rajasegarar, J. C. Bezdek, S. Karunasekera, and C. Leckie, "Efficient unsupervised parameter estimation for one-class support vector machines," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 10, pp. 5057–5070, Oct. 2018.

[6] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. no. 15.
[7] J. E. Zhang, D. Wu, and B. Boulet, "Time series anomaly detection for smart grids: A survey," in *Proc. IEEE Elect. Power Energy Conf.*, 2021, pp. 125–130.
[8] V. L. Cao, M. Nicolau, and J. McDermott, "Learning neural representations for network anomaly detection," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 3074–3087, Aug. 2019.
[9] E. Khaledian, S. Pandey, P. Kundu, and A.K. Srivastava, "Real-time synchrophasor data anomaly detection and classification using isolation forest, kmeans, and loop," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2378–2388, May 2021.
[10] M. Zhou, Y. Wang, A.K. Srivastava, Y. Wu, and P. Banerjee, "Ensemble-based algorithm for synchrophasor data anomaly detection," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2979–2988, May 2019.
[11] R. T. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Annu. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2610–2620.
[12] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2828–2837.
[13] J. J. Q. Yu, Y. Hou, A. Y. S. Lam, and V. O. K. Li, "Intelligent fault detection scheme for microgrids with wavelet-based deep neural networks," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1694–1703, Mar. 2019.
[14] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80778–80788, 2019.
[15] N. Muralidhar et al., "illiad: InteLLigent invariant and anomaly detection in cyber-physical systems," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, pp. 1–20, 2018.
[16] G. Horváth, E. Kovács, R. Molontay, and S. Nováczki, "Copula-based anomaly scoring and localization for large-scale, high-dimensional continuous data," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 3, pp. 1–26, 2020.
[17] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward effective intrusion detection using log-cosh conditional variational autoencoder," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6187–6196, Apr. 2021.
[18] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.
[19] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
[20] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: Taking Yelp review dataset as an example," in *Proc. Int. Workshop Electron. Commun. Artif. Intell.*, 2020, pp. 98–101.
[21] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *stat*, vol. 1050, p. 1, 2014.
[22] T. Behdadnia and M. Parlak, "Mitigating erroneous PMU measurements via an ANN to enhance ML-based transient stability prediction accuracy," Aug. 2021, doi: 10.21227/b9kg-6c55.
[23] R. Ma, "IEEE 118-bus transient event data," Sep. 2021, doi: 10.21227/6f5v-q924.
[24] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.
[25] H. Goyel and S. Swarup, "Data integrity attack in cyber physical systems," Mar. 2021, doi: 10.21227/g4th-qz60.
[26] D. P. Kingma and J. L. Ba, "ADAM: A method for stochastic optimization."
[27] E. M. Ronchetti and P. J. Huber, *Robust Statistics*. Hoboken, NJ, USA: Wiley, 2009.
[28] J. James, D. J. Hill, A. Y. Lam, J. Gu, and V. O. Li, "Intelligent time-adaptive transient stability assessment system," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018.
[29] M. Pavella, D. Ernst, and D. Ruiz-Vega, *Transient Stability of Power Systems: A Unified Approach to Assessment and Control*. Berlin, Germany: Springer, 2000.

**Dibyajyoti Guha** (Member, IEEE) received the B.Tech. degree in computer science and engineering from the Kalyani Government Engineering College, University of Kalyani, Kalyani, India, in 2003, the M.S. (by Research) degree in telecommunication engineering from IIT Kharagpur, Kharagpur, India, in 2012, and the Ph.D. degree in mathematics from IIT Bhubaneswar, Bhubaneswar, India, in 2016.

He is currently an Associate Professor with the Department of CSE, Gitam Deemed to be University, Hyderabad, India. He was a Postdoctoral Research Fellow with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. He was with HCL-Cisco Offshore Development Center, Chennai, India, and Sipera Systems, Inc., (a startup company that was acquired by Avaya in 2011) as a Software Engineer in VoIP/data/telecom routing and SSL protocols in Cisco and Juniper high-end routers. He has authored or coauthored articles published in premier journals, such as IEEE/ACM TRANSACTION ON NETWORKING, *Performance Evaluation* (Elsevier), and *Applied Mathematical Modelling* (Elsevier). His research interests include machine learning, variational Bayes learning, computer vision, and brain computer interface.

**Rajdeep Chatterjee** (Member, IEEE) received the Bachelor of Engineering degree in computer science and engineering from The University of Burdwan, Bardhaman, India, in 2008, and the Master of Technology and Ph.D. degree in computer science and engineering from KIIT Deemed to be University, Bhubaneswar, India, in 2011 and 2020, respectively.

He is currently an Associate Professor with the School of Computer Engineering, KIIT Deemed to be University. He has authored or coauthored research articles in many reputed international conferences and journals. His research interests include brain.computer interface, machine learning, deep learning and computer vision.

Dr. Chatterjee was the recipient of MHRD (Govt. of India) scholarship in his masters for possessing All India Rank 1410 in GATE-2008. He is a Co-Founder of the international conference series on Computational Intelligence & Networks.

**Biplab Sikdar** (Senior Member, IEEE) received the B.Tech. degree in electronics and communication engineering from North Eastern Hill University, Shillong, India, in 1996, the M.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1998, and the Ph.D. degree in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2001.

He is currently a Professor with the Department of Electrical and Computer Engineering, National University of Singapore (NUS), Singapore, where he also the acting Head of Department for the Department of Electrical and Computer Engineering. With NUS, he also directs the US$40 million corporate research lab with Cisco Systems.

Dr. Sikdar was the recipient of the NSF CAREER Award, Tan Chin Tuan Fellowship from NTU Singapore, Japan Society for Promotion of Science Fellowship, and Leiv Eiriksson Fellowship from the Research Council of Norway. He is a Distinguished Lecturer of IEEE and ACM and is or was an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY, and IEEE INTERNET OF THINGS JOURNAL.