Improved Variational Autoencoder Anomaly Detection in Time Series Data

doi: https://doi.org/10.1109/SMC42975.2020.9283010

Improved Variational Autoencoder Anomaly Detection in Time Series Data

Umaporn Yokkampon[1,*], Sakmongkon Chumkamon[1], Abbe Mowshowitz[2], Ryusuke Fujisawa[1], Eiji Hayashi[1]
[1]Department of Computer Science and Systems Engineering, Kyushu Institute of Technology, Fukuoka, Japan
[2]Department of Computer Science, The City College of New York, New York, USA
Email: may@mmcs.mse.kyutech.ac.jp, m-san@mmcs.mse.kyutech.ac.jp, amowshowitz@ccny.cuny.edu, fujisawa@ces.kyutech.ac.jp, haya@mse.kyutech.ac.jp

***Abstract*— Uncertainty in observations about the state of affairs is unavoidable, and generally undesirable, so we are motivated to try to minimize its effect on data analysis. Detection of anomalies in data has become an important research area. In this paper, we propose a novel approach to anomaly detection based on the Variational Autoencoder method with a Mish activation function and a Negative Log-Likelihood loss function. The proposed method is validated with ten standard datasets, comparing performance on each of the various activation functions and loss functions. Experimental results show that our proposed method offers an improvement over existing methods. Statistical properties (i.e., F1 score, AUC, and ROC) of the method are also examined in light of the experimental results.**

Keywords—Anomaly Detection, Variational Autoencoder, Time Series Data, Activation Functions, Loss Functions

I. INTRODUCTION

Courtesy of developments in artificial intelligence, especially machine learning, large data sets can be analyzed effectively to detect anomalies. Confidence in data analysis is critical for all organizations, and such confidence can only be assured with quality data and metadata. To be useful in applications, data collected at a point in time or over an extended period must be accurate and reliable. Our current work is directed at improving accuracy and reliability of data with the help of anomaly detection.

Real-world data is often noisy, incomplete, and inconsistent since it derives from a variety of sources. This condition may reduce the performance and accuracy of data analysis. Thus it is important to ensure that anomalies in the data are detected and properly treated. Our research focuses on the development of methods for detecting anomalies, i.e., potentially unreliable data. Anomaly detection is essential as a foundation for ensuring the accuracy and reliability of data critical in so many practical settings such as factory automation, advertising, and financial transaction in banking, and insurance.

Anomaly detection aims to identify data that departs from what is expected in a dataset. Such detection plays an important role in network security, medical monitoring, social media monitoring, intrusion detection, production system monitoring, as well as other areas.

Several anomaly detection methods have been developed. Especially noteworthy are methods based on dimension reduction. These methods aim to reduce the dimension of the space defined by a data set, while retaining the important features of the original data. Dimension reduction methods differ according to their handling of feature selection and feature extraction; in addition these methods may be linear or non-linear. The Autoencoder is particularly important in this area. This method attempts to compress and thus map input data to a reduced dimensional space, and then use an encoding-decoding process to reconstruct the input data set. A newer method of dimensionality reduction is Variational Autoencoder (VAE), which evolved from Autoencoder. VAE is a type of neural networks that can learn to compress data in a completely unsupervised way. This method outperforms Autoencoder, by imposing a probability distribution, with given mean and variance, on the latent space, and using a sample from this distribution to reconstruct the data.

In this paper, we propose an anomaly detection method based on the Variational Autoencoder method using a Mish activation function and a Negative Log-Likelihood loss function to enhance the performance. The proposed method is validated using ten standard datasets, which contain sensor, ECG, and image data types. The validation is based on Area Under the Curve, Precision, Recall, and F1-score criteria.

II. BACKGROUND AND RELATED WORK

Anomaly detection is an important problem that has been investigated extensively in diverse areas, see [1], for example. It figures prominently in a variety of applications. Anomaly detection is a critical tool in the identification of potentially malicious intrusions in computer networks. Such intrusions might take the form of denial of service attacks, data breaches, etc. As mentioned in the previous section, many anomaly detection techniques have been developed, notably methods based on dimension reduction. One such method is Principal Component Analysis (PCA) [2], a linear algebra technique that can be used to achieve dimension reduction automatically. Vasan et al. [3] propose PCA using various classifier algorithms, and determine its reduction ratio in experiments on two benchmark datasets. This method offers good accuracy. PCA can reduce dimensions, but it is important to note that it results in loss of information, and cannot provide 100% accuracy.

Autoencoder is another method for reducing the dimension of a dataset. This method lowers dimensions through data compression, and then produces reconstructed data similar to the original input. Autoencoder outclasses PCA in that PCA is a linear transformation, whereas autoencoders use nonlinear transformations and thereby models relatively complex relationships. Yokkampon et al. [4] propose the autoencoder method with spiking raw data in the frequency domain to detect anomalies. This method offers good anomaly detection performance by analyzing time series data in the frequency domain.

The newest deep learning method is variational autoencoder (VAE) [5], which is a variant of autoencoder. VAE can design complex generative models of data, and fit them to large datasets. VAE outperforms autoencoders that do not use the encoder-decoder process to reconstruct the input. VAE determines a probability distribution on the latent space and learns the distribution so that the decoder's distribution of outputs matches that of the observed data. Then, it samples from this distribution to generate new data. Bayer and Osendorfer [6] proposed the recurrent neural networks with latent variables using four polyphonic musical data sets and motion capture data to model time series data and introduced Stochastic Recurrent Networks (STORNs). Soelch et al. [7] proposed a Stochastic Recurrent Network (STORN) to learn robot time series data. That method offers good anomaly detection performance, both off-line and on-line. An and Cho [8] proposed using the reconstruction probability from the variational autoencoder and a new probabilistic anomaly score to detect anomalies. This method outperforms autoencoder and PCA. Zhang et al. [9] proposed the variational autoencoder with re-Encoder and Latent Constraint network (VELC) and used LSTM as the encoder and decoder part of VAE to identify time series anomalies. This method offers excellent performance for anomaly detection. Wang et al. [10] proposed deep autoencoder networks and spectral clustering using the Mish activation function for acoustic scene analysis. This method outperforms many state-of-the-art methods.

In the research reported here, we propose the Variational Autoencoder method based on the Mish activation function and Negative Log-Likelihood loss function to analyze and detect anomalies in time series data. Performance is assessed by comparing results using various activation functions and loss functions designed to improve performance. The Mish activation function helps to avoid saturation, which generally causes training to drastically slow down performance due to near-zero gradients. Saturation also has undesirable regularization effects. We compare our results with research reported in [9], which also used variational autoencoder but did not use the same loss function. Our approach is designed to avoid overfitting and ensure that the latent space has suitable properties that facilitate the generative process. Therefore, the variational autoencoder can be defined as being an autoencoder whose training is regularized, which implies that it can design complex generative models of data and fit them to large datasets. This is where VAE works better than any other method currently available. To justify this claim, we briefly explain the VAE activation functions and loss functions in this section.

A. Variational Autoencoder

Autoencoder is a simple learning network designed to transform inputs into outputs with the minimum possible error so it can compress and decompress images or documents and reduce noise in the data. This system consists of three main components or layers. The first contains the code, encoder, and decoder. The encoder compresses the input and generates the code; the decoder reconstructs new input using only this code. One drawback of the autoencoder is a hidden layer which may not be continuous and could make interpolation difficult. Variational Autoencoder (VAE) was an outgrowth of autoencoder designed to address this problem.

VAE (Kingma and Welling, 2013) [5] is a deep learning technique for dimension reduction. VAE relies on probability distributions of observations in latent space and makes strong assumptions about the distribution of latent variables. Thus, rather than generate an encoder that outputs a single value to describe each latent variable, it defines an encoder to describe a probability distribution for each latent variable.

Like autoencoder, VAE also has three main layers. The first one consists of the encoder, decoder, and loss function. A variable x represents the data or input, and x is generated from a latent variable z, which is the encoded representation. The first process samples a latent representation z from the prior distribution $p(z)$ and then samples the data x from the conditional likelihood distribution $p(x|z)$.

The decoder is defined by $p(x|z)$ which corresponds to the distribution of the decoded variable given the encoded one, whereas the encoder is defined by $p(\mathrm{z}|\mathrm{x})$ which corresponds to the distribution of the encoded variable given the decoded one.

The loss function of the variational autoencoder consist of two terms: a reconstruction loss term, which can be thought of maximizing the reconstruction likelihood, and a regularizer term which encourages learn distribution $q(\mathrm{z}|\mathrm{x})$ to be similar to the true prior distribution $p(z)$. The loss function is defined as follows:

$$L(x_i) = -E_{z\sim q(z|x_i)}[\log p(x_i|z)] + KL\left(q(z|x_i) \,||\, p(z)\right) \quad (1)$$

For the regularizer term, VAE uses the Kullback-Leibler divergence to control the divergence between the probability distributions. The objective is to obtain the best fit between the mean and variance of the probability distribution and the corresponding parameters of the target distribution.

B. Activation Functions

Activation functions are mathematical equations used in neural networks for transforming the weighted sum of inputs from a node to its output. Activation functions can be either linear or non-linear. In this research, we use three activation functions defined as follows.

1) Leaky ReLu: Leaky ReLu is a recently developed activation function. It is designed to minimize sensitivity to the dying ReLU problem by having a small negative slope (in the neighborhood of 0.01).

The Leaky ReLu activation function is defined by (see Maas et al., 2013) [11]:

$$f(x) = \begin{cases} 0.01x, & \text{if } x < 0 \\ x, & \text{otherwise} \end{cases} \tag{2}$$

2) Swish: Swish is an activation function discovered by researchers at Google (Ramachandran et al., 2017) [12]. The swish activation function is a combination of the sigmoid activation function and the input function. The shape of Swish is similar to ReLu, but it performs better than ReLu.

More precisely, Sigmoid is defined as follows:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3}$$

The Swish function is defined by:

$$f(x) = x \cdot sigmoid(x) = \frac{x}{1+e^{-x}} \tag{4}$$

3) Mish: Mish is also one of the new activation functions in the deep learning world. It is a combination of hyperbolic tangent and softplus. Mish performs better than ReLu, sigmoid, and even Swish.

The Mish activation function is given by:

$$f(x) = x \cdot \tanh(softplus(x)) \tag{5}$$

where $softplus(x) = \ln(1+e^{x})$

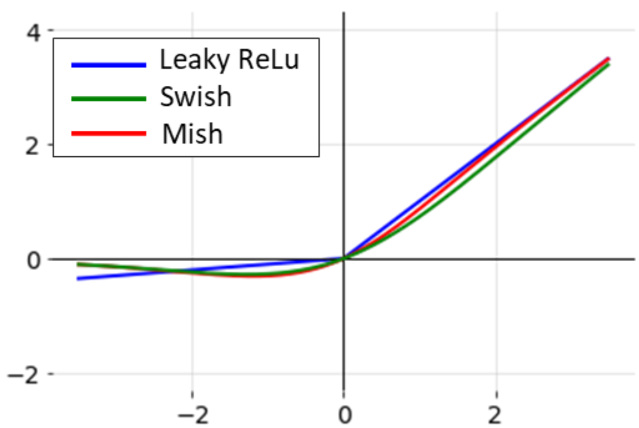

Fig. 1. Activation Functions

C. Loss Functions

Loss functions are used to optimize the parameter values in a neural network model and also can be used to measure the difference between the estimated and actual values for an instance of data. In this research, we use two loss functions.

1) Mean Square Error: Mean Square Error (MSE) is the most commonly used loss function. MSE is calculated as the average of squared difference between predictions and actual observations. The result is always positive.

It is given by the following:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{6}$$

2) Negative Log-Likelihood: The Negative Log-Likelihood loss function is widely used in neural networks. It is typically used as a measure of the accuracy of a classifier.

This function is defined by:

$$-\sum_{j=1}^{M} y_j \log \hat{y}_j \tag{7}$$

III. Proposed Mehod

This section describes our proposed method for improving anomaly detection and discusses the data sets and the evaluation metrics used. The structure of our proposed method is shown in Fig. 2. As explained earlier, this system uses the variational autoencoder method based on the Mish activation function and Negative Log-Likelihood loss function. The system is evaluated on time series data sets by comparing performance using different functions.

Our proposed system consists of three main parts. First is the input; second is the variational autoencoder method; and the last is the set of results. The input part contains ten time series data sets. The procedure is to input all ten time series data sets to the variational autoencoder in order to identify anomalies by constructing the encoder and decoder. Six cases are distinguished. The results are then obtained from the variational autoencoder reconstruction value.

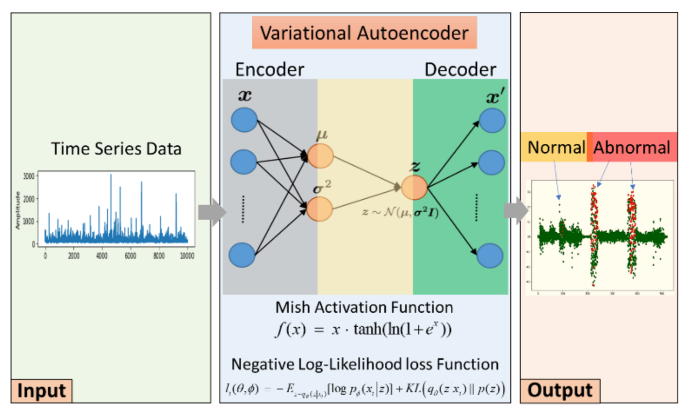

Fig. 2. Concept of the proposed method

A. Data Sets

To illustrate the effectiveness of the proposed method, we conducted experiments using three types of time series data (sensor, ECG, and image), obtained from a UCR public data set [13] and a UCI public data set [14]. All datasets are given in time series format, and every data point is labeled. For all datasets, we chose the minority class as an anomaly class. The summary of datasets is shown in Table 1. For each dataset, 80% of normal data has been used for the training phase; the remaining 20% and the anomalies have been used for testing.

TABLE I. SUMMARY OF TIME SERIES DATA SETS

Datasets	Data type	Length	Number of instances	Anomaly Ratio
ItalyPowerDemand	Sensor	24	1096	0.49
Wafer	Sensor	152	7164	0.11
SonyAIBORobotSurface2	Sensor	65	980	0.38
ECGFiveDays	ECG	136	884	0.50
TwoLeadECG	ECG	82	1162	0.50
MoteStrain	Sensor	84	1272	0.46
Arrhythmia	Sensor	274	452	0.40
DistalPhalanxOutlineCorrect	Image	80	876	0.38
MiddlePhalanxOutlineCorrect	Image	80	891	0.38
PhalangesOutlinesCorrect	Image	80	2658	0.36

B. Performance Evaluation

To evaluate the accuracy of our proposed method for improving anomaly detection, we use Area under the curve of the receiver operating characteristic (AUC), Precision (Pre), Recall (Rec), and F1-Score criteria, which are defined as follows:

$$\text{Pre} = \frac{TP}{TP+FP} \tag{8}$$

$$\text{Rec} = \frac{TP}{TP+FN} \tag{9}$$

$$\text{F1} = 2 \times \frac{\text{Pre}\times\text{Rec}}{\text{Pre}+\text{Rec}} \tag{10}$$

where TP is the correctly detected anomaly, FP is the falsely detected anomaly, TN is the correctly assigned normal, and FN is the falsely assigned normal.

IV. RESULTS AND DISCUSSION

In this section, we evaluate the anomaly detection performance for ten time-series data sets using the Variational autoencoder method. The experiments are divided into 6 cases which consist of our proposed VAE using the Mish activation function and Negative Log-likelihood compared with Leaky ReLu and Swish activation functions, and MSE loss function. The experiment with the standard dataset is designed to benchmark the method and investigate its characteristics.

TABLE II. CASE I: ACTIVATION FUNCTION: LEAKY RELU, LOSS FUNCTION: MEAN SQUARE ERROR

Datasets	Performance Evaluation			
	AUC	*Precision*	*Recall*	*F1-Score*
ItalyPowerDemand	0.7227	0.7807	0.5394	0.6380
Wafer	0.6148	0.7567	0.8995	0.8219
SonyAIBORobotSurface2	0.8364	0.8729	0.7007	0.7774
ECGFiveDays	0.7627	0.9022	0.6288	0.7411
TwoLeadECG	0.6905	0.8376	0.5632	0.6735
MoteStrain	0.8404	0.9084	0.6230	0.7391
Arrhythmia	0.6000	0.8200	0.6029	0.6949
DistalPhalanxOutlineCorrect	0.5968	0.8190	0.6515	0.7257
MiddlePhalanxOutlineCorrect	0.4811	0.7436	0.6493	0.6932
PhalangesOutlinesCorrect	0.5355	0.7819	0.6291	0.6972

TABLE III. CASE II: ACTIVATION FUNCTION: LEAKY RELU, LOSS FUNCTION: NEGATIVE LOG-LIKELIHOOD

Datasets	Performance Evaluation			
	AUC	*Precision*	*Recall*	*F1-Score*
ItalyPowerDemand	0.8537	0.9612	0.6000	0.7388
Wafer	0.9899	0.8353	**1.0000**	0.9102
SonyAIBORobotSurface2	0.9510	0.9735	0.7483	0.8462
ECGFiveDays	0.8948	0.9890	0.6818	0.8072
TwoLeadECG	0.7217	0.9561	0.6229	0.7543
MoteStrain	0.8876	0.8904	0.6806	0.7715
Arrhythmia	0.7424	0.9000	0.6618	0.7627
DistalPhalanxOutlineCorrect	0.7726	0.8929	0.7576	0.8197
MiddlePhalanxOutlineCorrect	0.5990	0.8083	0.7239	0.7638
PhalangesOutlinesCorrect	**0.6812**	0.8713	0.7293	0.7940

The results of anomaly detection and comparisons are summarized in Table 2. - Table 7. We presented the results with ten data sets for discussion. Table 2 shows case I, which uses the Leaky ReLu activation function and Mean Square Error loss function. Table 3. shows case II, which uses Leaky ReLu activation function that is normally used in VAE and Negative Log-Likelihood loss function. The results indicate that case II could improve outcomes in AUC and F1-Score values higher than case I on all the data sets. Cases I and II draw a comparison with the same activation function but different loss functions. Thus, the Negative Log-Likelihood loss function can be used to improve the effectiveness of anomaly detection for time series data. Note that in case II, performance on the Wafer data set exhibits perfect recall results.

TABLE IV. CASE III: ACTIVATION FUNCTION: SWISH, LOSS FUNCTION: MEAN SQUARE ERROR

Datasets	Performance Evaluation			
	AUC	*Precision*	*Recall*	*F1-Score*
ItalyPowerDemand	0.7921	0.7768	0.5686	0.6566
Wafer	0.6941	0.7653	0.9372	0.8426
SonyAIBORobotSurface2	0.9161	0.9421	0.7651	0.8444
ECGFiveDays	0.8334	0.9432	0.6288	0.7545
TwoLeadECG	0.7709	0.9333	0.6328	0.7542
MoteStrain	0.8628	0.9453	0.6402	0.7634
Arrhythmia	0.7057	0.8704	0.6912	0.7705
DistalPhalanxOutlineCorrect	0.7430	0.8899	0.7462	0.8117
MiddlePhalanxOutlineCorrect	0.5164	0.8318	0.6496	0.7295
PhalangesOutlinesCorrect	0.5905	0.7972	0.7093	0.7507

TABLE V. CASE IV: ACTIVATION FUNCTION: SWISH, LOSS FUNCTION: NEGATIVE LOG-LIKELIHOOD

Datasets	Performance Evaluation			
	AUC	*Precision*	*Recall*	*F1-Score*
ItalyPowerDemand	0.8962	0.9244	0.6667	0.7746
Wafer	0.9903	0.8358	**1.0000**	0.9106
SonyAIBORobotSurface2	0.9637	0.9318	0.8367	0.8817
ECGFiveDays	0.9468	0.9878	0.7232	0.8351
TwoLeadECG	0.7451	0.9643	0.6316	0.7633
MoteStrain	**0.8935**	0.9333	0.6597	0.7730
Arrhythmia	0.7408	0.8936	0.6176	0.7304
DistalPhalanxOutlineCorrect	0.7832	0.9417	0.7348	0.8255
MiddlePhalanxOutlineCorrect	0.6123	0.9192	0.6791	0.7811
PhalangesOutlinesCorrect	0.6603	0.8459	0.7257	0.7812

Table 4. shows case III, which uses Swish activation function and Mean Square Error loss function. Table 5 presents case IV, which uses Swish activation function and Negative Log-Likelihood loss function. The results show that case IV could improve outcomes in AUC values to a greater extent than case III on 9 data sets, just as indicated for the F1-Score. Moreover, in case IV, performance on the Wafer data set shows perfect recall results. Thus, Negative Log-Likelihood loss function can be used to improve the effectiveness of anomaly detection for time series data.

TABLE VI. CASE V: ACTIVATION FUNCTION: MISH, LOSS FUNCTION: MEAN SQUARE ERROR

Datasets	Performance Evaluation			
	AUC	***Precision***	***Recall***	***F1-Score***
ItalyPowerDemand	0.8404	0.8333	0.6164	0.7087
Wafer	0.7111	0.7843	0.9520	0.8601
SonyAIBORobotSurface2	0.9344	0.9389	0.8425	0.8881
ECGFiveDays	0.8866	0.9368	0.6593	0.7739
TwoLeadECG	0.8179	0.9444	0.7000	0.8041
MoteStrain	0.8852	0.9420	0.6806	**0.7903**
Arrhythmia	0.7713	0.9388	0.7188	0.8142
DistalPhalanxOutlineCorrect	0.7596	0.9000	0.7500	0.8182
MiddlePhalanxOutlineCorrect	0.5610	0.8087	0.6940	0.7470
PhalangesOutlinesCorrect	0.6072	0.8210	0.7261	0.7707

TABLE VII. CASE VI: ACTIVATION FUNCTION: MISH, LOSS FUNCTION: NEGATIVE LOG-LIKELIHOOD

Datasets	Performance Evaluation			
	AUC	***Precision***	***Recall***	***F1-Score***
ItalyPowerDemand	**0.9492**	0.9730	0.6545	**0.7826**
Wafer	**0.9956**	0.8478	**1.0000**	**0.9176**
SonyAIBORobotSurface2	**0.9693**	0.9688	0.8435	**0.9018**
ECGFiveDays	**0.9767**	0.8804	0.9759	**0.9257**
TwoLeadECG	**0.8955**	0.9231	0.7742	**0.8421**
MoteStrain	0.8788	0.8973	0.6859	0.7774
Arrhythmia	**0.8023**	0.8852	0.7941	**0.8372**
DistalPhalanxOutlineCorrect	**0.8129**	0.9107	0.7727	**0.8361**
MiddlePhalanxOutlineCorrect	**0.7797**	0.8761	0.7388	**0.8016**
PhalangesOutlinesCorrect	0.6556	0.8504	0.7494	**0.7967**

Table 6. shows case V which uses the Mish activation function and Mean Square Error loss function. Table 7 shows case VI which is our proposed method. Case VI uses the Mish activation function and Negative Log-Likelihood loss function. The results show that case VI could improve outcomes in AUC and F1-Score values more than case V on almost every data set. Moreover, in case VI, performance on the Wafer data set also shows perfect recall results. Thus, the Negative Log-Likelihood loss function can be used to improve the effectiveness of anomaly detection for time series data.

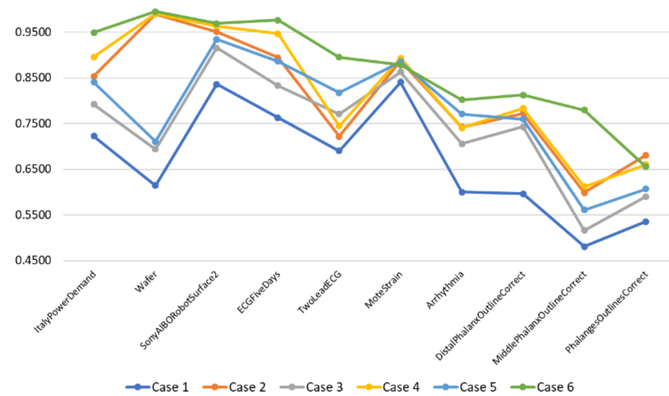

Fig. 3. AUC comparisons all six cases on Variational autoencoder

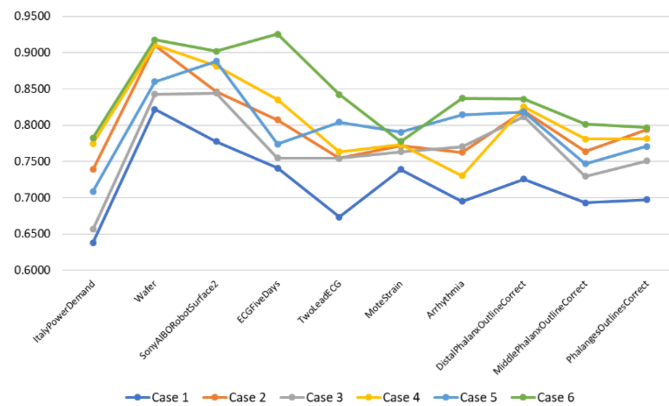

Fig. 4. F1-Score comparisons all six cases on Variational autoencoder

Figures 3 and 4 show the chart of AUC and F1-Score comparisons, respectively, for all six cases for Variational Autoencoder. It is quite clear that our method in case VI which uses the Mish activation function and Negative Log-Likelihood loss function outperforms on all metrics for all the data sets. For the Wafer data set, the result of AUC of case I yields 61.48%, but in case VI the results are improved to 99.22%. In addition, the F1-Score of case I yields only 84.26% compared with 91.76% for case VI. The Wafer data set results for recall in case II, IV, and VI, which uses Negative Log-Likelihood loss function, show perfect recall.

TABLE VIII. COMPARING AUC OF VAE RESULTS FROM THE RECENT RESEARCH

Datasets	OUR*	ANOGAN	ALAD	MLP-VAE	IF
ItalyPowerDemand	0.761	0.516	0.538	**0.768**	0.763
Wafer	**0.965**	0.558	0.587	0.790	0.847
ECGFiveDays	**0.970**	0.970	0.694	0.910	0.678
TwoLeadECG	**0.891**	0.554	0.515	0.731	0.760
MoteStrain	**0.840**	0.746	0.504	0.750	0.762
Arrhythmia	**0.758**	0.576	0.515	0.747	0.530
KDD99	**0.958**	0.887	0.950	0.622	0.929
GunPointAgeSpan	**0.881**	0.515	0.547	0.821	0.612
ToeSegmentation2	**0.846**	0.547	0.544	0.816	0.787
Herring	0.659	0.488	0.569	0.627	**0.698**

We also compare our results to the latest reported in “Time Series Anomaly Detection with Variational Autoencoders” [10] in 2019. There are six common datasets utilized in our method, namely, ItalyPowerDemand, Wafer, ECGFiveDays, TwoLeadECG, MoteStrain, and Arrhythmia. The results in terms of AUC are shown in Table VII. These results show that our method could improve AUC outcome values compared to the approach taken in [10] for all six datasets. In general, the results show that our method constitutes an improvement in anomaly detection performance for time series data relative to the results reported in the literature.

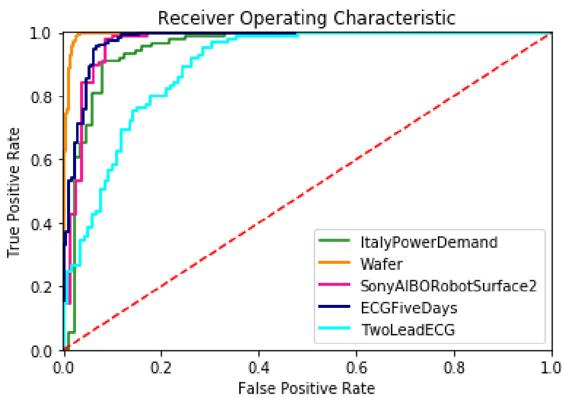

Fig. 5. ROC results of ItalyPowerDemand, Wafer, SonyAIBORobotSurface2, ECGFiveDays, TwoLeadECG data sets

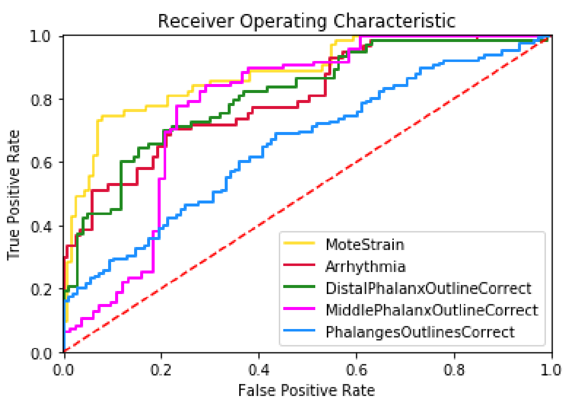

Fig. 6. ROC results of MoteStrain, Arrhythmia, DistalPhalanxOutlineCorrect, MiddlePhalanxOutlineCorrect, PhalangesOutlinesCorrect data sets

Finally, we investigate the characteristic of our method. Figures 5 and 6 illustrate the diagnostic ability of our method from case VI illustrated by the ROC curve. The charts show that the performance on the Wafer data set is better than on the other datasets. The performance results on the Phalanges Outlines Correct data set are worse, but could be improved the result based on recent research and other methods. Moreover, there are three image-based datasets, i.e., Distal Phalanx Outline Correct, Middle Phalanx Outline Correct, and Phalanges Outlines Correct, for which accurate anomaly detection is more difficult to achieve. This may be a consequence of the fact that image data is more complex than those based on signal or sensor data.

V. Conclusion

The system proposed here uses the Variational autoencoder method based on various activation functions and loss functions together. Performance has been evaluated by means of six different cases, and computation of the metrics AUC, Precision, Recall, and F1-Score. The experiments on anomaly detection show that our proposed method, which uses Mish activation function and Negative Log-Likelihood loss function could improve the accuracy of detecting anomalies on all the usual metrics for time series data. Therefore, the Mish activation function and loss function have been shown to improve anomaly detection on the criteria of precision, recall, F1 score, AUC and ROC.

In future research, we intend to use various data transformations in variational autoencoder to detect anomalies in time series data.

Acknowledgment

This research was partially supported by the Religion revitalization project in Kitakyushu by Japanese Government.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Computing Surveys, vol. 41(3), 2009.

[2] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, “Efficient GAN-Based Anomaly Detection,” arXiv preprint arXiv:1802.06222, 2018.

[3] K. Vasan, and B. Surendiran, “Dimensionality reduction using Principal Component Analysis for network intrusion detection,” Perspectives in Science, 8, pp. 510-512, 2016.

[4] U. Yokkampon, S. Chumkamon, A. Mowshowitz, and E. Hayashi, “Autoencoder with Spiking in Frequency Domain for Anomaly Detection of Uncertainty Event,” Journal of Robotics, Networking and Artificial Life, vol. 6, no. 4, pp. 231-234, 2020.

[5] D. P. Kingma, M. Welling, “Auto-encoding variational Bayes,” Proc. 2nd Int. Conf. Learn. Represen., April 2014.

[6] J. Bayer, C. Osendorfer, “Learning stochastic recurrent networks,” Proc. NIPS 2014 Workshop Advances Variational Inf., 2014.

[7] M. Sölch, J. Bayer, M. Ludersdorfer, P. van der Smagt, “Variational inference for on-line anomaly detection in high-dimensional time series,” Proc. ICML 2016 Anomaly Detection Workshop, 2016.

[8] J. An, and S. Cho, “Variational autoencoder based anomaly detection using reconstruction probability,” Special Lecture on IE 2, no. 1, 2015.

[9] C. Zhang, and Y. Chen, “Time Series Anomaly Detection with Variational Autoencoders,” CoRR, abs/1907.01702, 2019.

[10] M. Wang, X.L. Zhang, and S. Rahardja, “An Unsupervised Deep Learning System for Acoustic Scene Analysis,” Applied Sciences, Vol. 10, No. 6, p. 2076, 2020.

[11] A. L. Maas, A. Y. Hannun, & A. Y. Ng, “Rectifier Nonlinearities Improve Neural Network Acoustic Models,” In Proc. Icml. Vol. 30, No. 1, p. 3, June 2013.

[12] P. Ramachandran, B. Zoph, and Q. V. Le, “Searching for Activation Functions,” ArXiv, 2017. [Online]. Available: 1710.05941; http://arxiv.org/abs/1710.05941.

[13] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, “The ucr time series classification archive,” October 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[14] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml