# Anomaly Detection for Sequential Data

**Click for Github code: Anomaly Detection for Sequential Data**

Georgios Sartzetakis
s151005@student.dtu.dk
Technical University of Denmark
Copenhagen, Denmark

Georgios Zefkilis
s151074@student.dtu.dk
Technical University of Denmark
Copenhagen, Denmark

Md Alamgir Kabir
s193074@student.dtu.dk
Technical University of Denmark
Copenhagen, Denmark

Leif Førland Schill
s153355@student.dtu.dk
Technical University of Denmark
Copenhagen, Denmark

## ABSTRACT

In this paper we are exploring the implementation of a version of a VAE for anomaly detection known as VELC [1]. VELC uses a re-encoder at the end of the decoder, and we compare the results with several versions of vanilla VAEs containing only an encoder and decoder. Initially, we built models that yield reliable results for the ECG dataset [2], and thereafter we used those models with the regular passenger cars dataset provided by the instructors of the course. We measure the performance of the models using the AUC. We obtained the AUC curve by thresholding different values of the mean squared error score between the actual and reconstructed time series. We obtained the best results with a bidirectional LSTM with re-encoder for a value of 0.997. The regular passenger cars data yielded poor results for this method of classification, which motivated a different approach thresholding the Lilliefors test statistic of the residuals. This gave better results, with the best AUC value 0.754 for a bidirectional GRU without re-encoding.

## KEYWORDS

Timeseries, Variational Inference, VAE, Anomaly detection

## 1 INTRODUCTION

Denmark spends approximately 5 million DKK per year on assessing road conditions and optimizing maintenance strategies with focus on safety, comfort, durability, etc. Apart from the high cost, limitations such as weather and road geometry make the monitoring and maintenance of the road a cumbersome process [3].

As an alternative way to monitor, maintain and manage the roads the Danish Innovation Fund has invested in the Live Road Assessment project (LiRA). The overall goal of the LiRA project is to collect data from sensors installed in green mobility cars (GM) [4] and develop associated models to assess road conditions. Road wear can thus be detected and improved much earlier, enabling the roads to be maintained more efficiently and for fewer resources [5].

The present study aims to analyse part of data collected from those sensors and develop an anomaly detection model able to spot roads that requires maintenance. Throughout the literature several ways have been used to detect anomalies from traditional machine learning (k-means clustering, Gaussian mixture models, PCA) to deep neural networks with the latter performing significantly better as the number of data is growing [6]. Recent studies from [1, 7–10] have applied variational auto-encoders (VAE) in different variations in order to detect anomalies in various datasets (time series, images etc.). The results of complex VAEs compared to conventional machine learning algorithms as well as vanilla VAEs have shown that they have great potential to detect anomalies much better. In the present study we have decided to reproduce the VAE model with constraint network and re-encoder as described in [1]. This so called VELC model has an interesting and different architecture compare to the other VAEs in the literature. The VELC model shows extremely good performance compare to the other state of the art models for anomaly detection. Additionally, the model has been tested on sequential datasets similar to those comes from the car sensors, therefore it constitutes an appealing approach for our case study. In our implementation though we will exclude the constraint network in the latent space due to discrepancies found in literature regarding its application.

## 2 DATASETS

This paper uses two datasets. The ECG dataset which contains the heart beats of 9499 samples over 140 time points. The dataset contains 5546 normal samples and 3953 anomaly samples. The models are trained on the normal data and tested on the anomaly data.

One can see from Figure 7( 7 7.1) that there is a clear distinction between the normal and anomaly data.

The GM dataset provided by the course instructors contain 5031 sample points and the features contain the vehicle acceleration for 369 time steps. The target columns contain the index of road roughness. We consider the anomaly data the ones with IRI (International Road Roughness Index) value $\geq 2$ and normal data the ones with IRI value $< 2$. Additionally, solely the perpendicular direction "z" considered for the modelling as it is the most informative one. Due to the noisy nature of the data we do not plot them all together but we can see how the series look like in Figure 2.

## 3 METHODS

### 3.1 Variational AutoEncoder (VAE)

Variational AutoEncoder (VAE), is a generative model, which combines Bayesian inference with the autoencoder framework. VAE is an architecture composed of both an encoder and a decoder that is trained to minimise the reconstruction error between the encoded-decoded data and the original data. In VAE, let $\mathbf{x}$ be the data we want to model, $\mathbf{z}$ the latent variable, $p(\mathbf{x})$ the probability distribution of data, $p(\mathbf{z})$ the probability distribution of the latent variable and $p(\mathbf{x}|\mathbf{z})$ the distribution of generating data given latent variable. So the generative process can be written as

$$p(\mathbf{x}) = \int_z p(\mathbf{x}, \mathbf{z})dz = \int_z p(\mathbf{x}|\mathbf{z})p(\mathbf{z})dz$$

which is analytically impossible to compute since the search space for $\mathbf{z}$ is continuous and computationally large. The trick is to approximate the posterior with $q(\mathbf{z}|\mathbf{x})$, in the encoder network which is more feasible to compute compared to our unknown true posterior, $p(\mathbf{z}|\mathbf{x})$ then the log likelihood $p(\mathbf{x}|\mathbf{z})$ is learned from decoder network. After series of statistical derivation the marginal log-likelihood of $\mathbf{x}$ can be written as follows

$$\log p(\mathbf{x}) = \mathbf{KL}\left[q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z} \mid \mathbf{x})\right] + \mathcal{L}(\mathbf{x})$$

where

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x}, \mathbf{z})\right] - \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log q(\mathbf{z} \mid \mathbf{x})\right]$$

is called variational lower bounds or evidence lower bound objective (ELBO). Kullback-Leibler divergence, $\mathbf{KL}\left[q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z} \mid \mathbf{x})\right]$ measured the similarity of the two distributions $p(\mathbf{z} \mid \mathbf{x})$ and $q(\mathbf{z} \mid \mathbf{x})$. Hence $\log p(\mathbf{x})$ is constant, to minimize the KL divergence we have to maximize the variational lower bound, $\mathcal{L}(\mathbf{x})$. With further statistical derivation we can rewrite the variational lower bound as follows

$$\mathcal{L}(\mathbf{x}) = -\mathbf{KL}\left[q(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})\right] + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z})\right] \tag{1}$$

## 3.2 Long short-term memory (LSTM)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture that has feedback connections. It cannot only process single data points but also entire sequences of data. A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell [11]. Figure 14 in Appendix 7.5 illustrates the architecture of an LSTM unit.

## 3.3 Gated Recurrent Unit (GRU)

The GRU's has a slightly different architecture than LSTM where it combines both the forget and input gate into a single gate called the update gate. Also, it merges the cell state and hidden state. The main advantage of GRU is the reduced number of parameters as compared to LSTM's without any compromise whatsoever which has resulted in faster convergence and a more generalized model [12]. See Appendix 7.6 for more details.

## 3.4 Methodology

The structure of the model has been inspired by [1] and is illustrated in Figure 1. The LSTM architecture refers to bi-directional LSTM which is two LSTM networks stacked on top of each other [1]. The model has also been tested with GRU architecture.

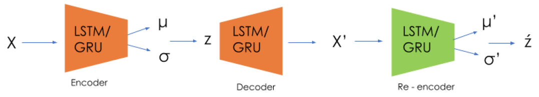

**Figure 1: The model architecture**

Following the same approach as in [1] the aim of our model is to detect the anomalies of sequential data (time series) based on the reconstruction error of a generative model. The model is trained with normal data. Hence, it will ideally record relatively small reconstruction errors for normal data, while large reconstruction errors for abnormal data.

As discussed in Section 1, the first part of the model consists of the VAE framework with an additional re-encoder network. The re-encoder will add complexity and more parameters in the network meaning the that the whole model can extract more features from original and latent space which will yield (according to [1]) better performance of the model. The loss function consisted of three parts as described in [1]. The first part is the loss function from VAE meaning the reconstruction loss $L_{recx}$ and the KL loss $L_{KL_1}$.

$$L_{rec_x} = ||X - X'||_2^2 \tag{2}$$

$$L_{KL_1} = \frac{1}{2}\sum_{i=1}^{z}[(\mu_i^2 + \sigma_i^2) - 1 - log(\sigma_1^2)] \tag{3}$$

The second part refers to the re-encoder's loss function which is the same as the KL loss $L_{KL_1}$ from the VAE framework.

$$L_{KL_2} = \frac{1}{2}\sum_{i=1}^{z'}[(\mu_i'^2 + \sigma_i'^2) - 1 - log(\sigma_1'^2)] \tag{4}$$

Lastly, the third part is the error of the latent vector

$$L_{latent} = ||Z - Z'||_2^2 \tag{5}$$

The loss function of the entire model is calculated as

$$L_{model} = L_{rec_x} + L_{KL_1} + L_{latent} + L_{KL_2} \tag{6}$$

We will use area under curve (AUC) as a metric of the overall performance of the model. In order to validate the performance of the architecture presented in Figure 1 we compare it against common VAE with LSTM and GRU architecture.

Additionally, we visualise the latent space of the re-encoder for the ECG and GM data using the PCA method to check the separability of the classes in the latent space. The latent space consists of 10 vectors in both encoder and re-encoder. The number of vectors was tested as hyper parameter along with values such as 15 and 20.

# 4 RESULTS

In order to ensure and illustrate the effectiveness of the VAE with re-encoder we carried out several tests on two kinds of time series datasets. Several variations of VAEs were built and tested during the process of building the model in Figure 1. The results of those models/variations are shown in Table 1 (The models named as VrAE refer to VAE with re-encoder).

**Table 1: Initial AUC for the tested models**

| *Model* | *ECG* | *Green Mobility* |
|---:|---|---|
| VAE LSTM | 0.928 | 0.542 |
| **VAE GRU** | 0.991 | **0.563** |
| VAE Bi-LSTM | 0.959 | 0.541 |
| VAE Bi-GRU | 0.992 | 0.482 |
| VrAE LSTM | 0.992 | 0.558 |
| VrAE GRU | 0.977 | 0.562 |
| VrAE Bi-LSTM | 0.996 | 0.559 |
| **VrAE Bi-GRU** | **0.997** | 0.480 |

The results for the ECG dataset are almost equally good for all the models. On the other hand when the models were tested on the GM data their performance was rather poor.

Figure 2 and 3 illustrates the reconstruction of normal and abnormal GM data and ECG data respectively for VrAE-BiGRU model.

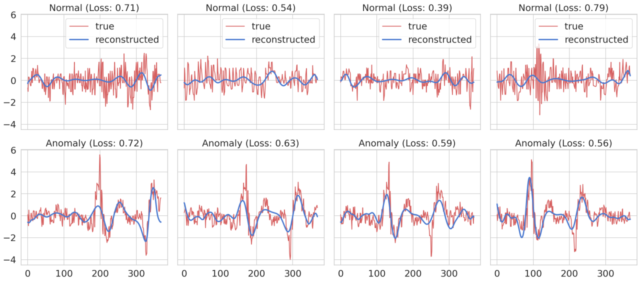

**Figure 2: Reconstruction error for GM data for VrAE-BiGRU (Top: Normal, Bottom: Anomalies)**

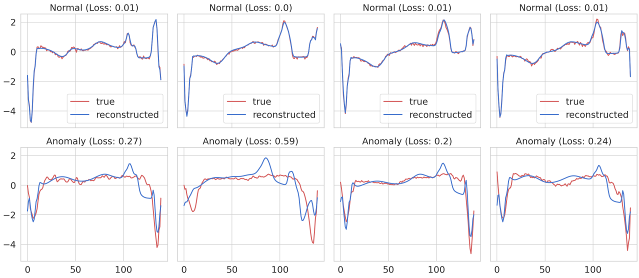

**Figure 3: Reconstruction error for ECG data for VrAE-BiGRU (Top: Normal, Bottom: Anomalies)**

Clearly the normal data (Figure 3, top row) form similar patterns for the ECG data thus the models could learn to reconstruct those patterns better. Additionally, in Figure 4 and Figure 5 we have visualised how the latent space of the means is distributed along two PCA components.

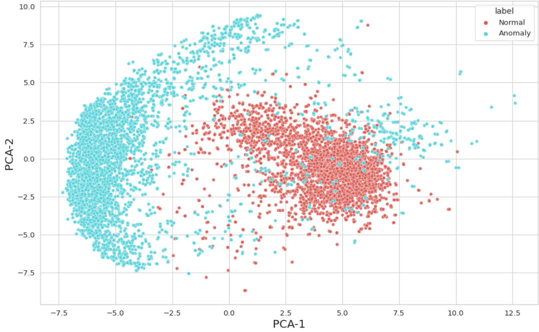

**Figure 4: Projection of the latent space (with PCA) of re-Encoder for ECG data using VrAE-BiGRU**

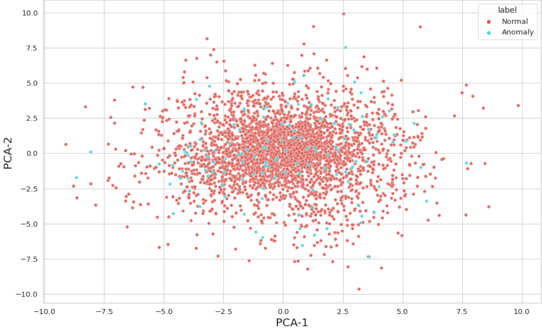

**Figure 5: Projection of the latent space (with PCA) of re-encoder for GM data using VrAE-BiGRU**

We can see for the case of the ECG data there is some obvious clustering between the normal and the and anomaly data. However, for the GM data it seems that this is not the case. The normal data have variations and noise that the model cannot easily learn to reconstruct successfully. Also, the model reconstructs the main trends in the anomaly data rather well (Figure 2). In order to overcome the issue of poor performance on GM data, we used the Lilliefors test statistic to classify instead of mean squared error. The Lilliefors test is used to test whether data come from a normally distributed population, when the null hypothesis does not specify which normal distribution [13]. In the present study we subtract the original data from the predicted ones and perform the Lilliefors test on the residuals. In case the residuals formed normal distribution (low test statistic) they were classified as normal or as anomaly otherwise. The AUC scores of the models after we implemented the statistical test are shown in Table 2.

**Table 2: AUC scores with Lilliefors statistical test**

| *Model* | *Green Mobility* |
|---:|:---|
| VAE LSTM | 0.625 |
| VAE GRU | 0.701 |
| VAE Bi-LSTM | 0.609 |
| **VAE Bi-GRU** | **0.754** |
| VrAE LSTM | 0.646 |
| VrAE GRU | 0.706 |
| VrAE Bi-LSTM | 0.525 |
| VrAE Bi-GRU | 0.731 |

We can see that the AUC scores have increased for almost all the models. VAE-GRU, VAE-BiGRU and VrAE-BiGRU record a significant increase of the magnitude of 0.25 on average. On the other hand, the VrAE-GRU shows solely an increase of 0.019. The models with LSTM architecture show an increase of 0.079 while the VrAE-LSTM records a decrease of 0.029.

We suspect that this is due to the nature of the data, namely that the high frequency oscillations from driving on a normal road can be seen as normally distributed noise, which is disrupted when the car travels over a rough patch.

In Appendix 7.2 we can see how the distribution of the reconstruction error for the normal and anomaly classes are highly overlapping, while the distribution of the Lilliefors' test statistic is more separated.

## 5 DISCUSSION/LIMITATIONS

Throughout the paper we tried to build a variation of the VELC model as described in [1]. Both ECG and GM data were tested in a numerous models/variations with LSTM and GRU architectures. The results for the ECG dataset recorded high and consistent AUC scores across all the models. The homogeneity of the normal/abnormal ECG data (see Figure 3) and lack of noise allowed the model to capture the normal and anomaly data very well. The GM data on the other hand recorded low AUC scores when using mean squared reconstruction error. The results though are more or less consistent along the models. In order to improve the performance of the models in GM data we applied another method of classifying normal and anomaly data. For that we used the Lilliefors test. The results showed significant improvements in few of the models (VrAE-BiGRU and VAE- BiGRU) while in others the deviations of AUC score were negligible compare to the initial results (VrAE-BiLSTM and VrAE GRU).

The overall results for the green mobility data indicate that our model has limitations which could be due to a variety of reasons. From the plot in figure 2 we observe that the data carry a lot of noise which may not allow to the model to capture the signal from the time series. We assume that if we remove the noise from the data the model we will extract solely the signal. Hence the model will learn better the underlying distribution. Figure 6 show how data looks like after de-noise and their corresponding AUC score is 0.638 (lillefors test) for VrAE-BiGRU

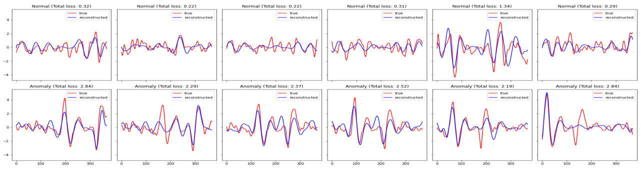

**Figure 6: Reconstruction of de-noised green mobility data for VrAE-BiGRU**

The AUC score doesn't seem to be improved even after smoothing up data. Further investigation and research needed in order to apply the smoothing process correctly with the right parameters such as sampling rate which in our case has been defined arbitrarily. However, due to lack of time no further investigation at this part carried out.

Other than denoising the data, more time series dataset should be used for simulations in order to ensure that our targeted models perform as they supposed to. The ECG dataset although comes from real data is a smooth and relatively easy dataset to work with. Therefore, regardless the high performance of the models on those data it doesn't constitute a strong indicator of how well they perform. Furthermore, our models score extremely high AUC scores on that dataset compare to even the VELC model itself (0.988). This might indicate that our implementations are prone to over-fitting.

Apart from limitations regarding the data, additional constrains in the performance of the models may come from the way the models have been implemented. The VrAEs with bidirectional GRU and LSTM architectures score the highest performance on the ECG data while this is not the case on GM dataset. Overall we observe that the performance of the VrAEs is slightly better than the vanilla VAEs while in few cases is worst. According to [1] a complete implementation of VELC model is able to outperform other state of the art models. Meaning that either solely the implementation of the re-encoder without the constraint network doesn't add significant value to the model or that our implementation of the re-encoder requires further investigation.

## 6 CONCLUSION

Throughout the study a VAE with Re-encoder was chosen as the main model that could potentially provide the optimal solutions to fulfil the scope of the study. The outcome of the study though revealed that vanilla bidirectional GRU VAE has slightly better performance on detecting the anomalies correctly. The limitations of the model though have significant influence on the performance of all models and especially on the VrAE ones. Therefore, no certain conclusion can be made on which model is indeed the best. Future steps to ensure a consistent and robust performance of the models involves further data pre-processing such as removing noise. Extensive research to fully understand whether the implementation of model has been applied correctly and how the constraint network could potentially affect the overall performance of the VrAEs compare to vanilla VAEs.

Click for Github code: Anomaly Detection for Sequential Data

## REFERENCES

[1] Chunkai Zhang, Shaocong Li, Hongye Zhang, and Yingyang Chen. Velc: A new variational autoencoder based model for time series anomaly detection, 2020.

[2] Ary Goldberger, Luís Amaral, L. Glass, Shlomo Havlin, J. Hausdorg, Plamen Ivanov, R. Mark, J. Mietus, G. Moody, Chung-Kang Peng, H. Stanley, and Physiotoolkit Physiobank. Components of a new research resource for complex physiologic signals. *PhysioNet*, 101, 01 2000.

[3] Milena Bajic and Tommy Sonne Alstrøm. Machine learning for road condition data, 2020.

[4] Green mobility. https://www.greenmobility.com.

[5] Donald Knuth. The lira project official website, http://lira-project.dk/.

[6] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey, 2019.

[7] Yifan Guo, Weixian Liao, Qianlong Wang, Lixing Yu, Tianxi Ji, and Pan Li. Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In Jun Zhu and Ichiro Takeuchi, editors, *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 97–112. PMLR, 14–16 Nov 2018.

[8] Adrian Alan Pol, Victor Berger, Gianluca Cerminara, Cecile Germain, and Maurizio Pierini. Anomaly detection with conditional variational autoencoders, 2020.

[9] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability, 2015.

[10] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection, 2019.

[11] Long short-term memory-Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/longshorttermmemory.

[12] Gated Reccurent Unit(GRU) Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/gatedrecurrentunit.

[13] Lilliefors test. https://en.wikipedia.org/wiki/lillieforstest.

# 7 APPENDIX

## 7.1 ECG dataset

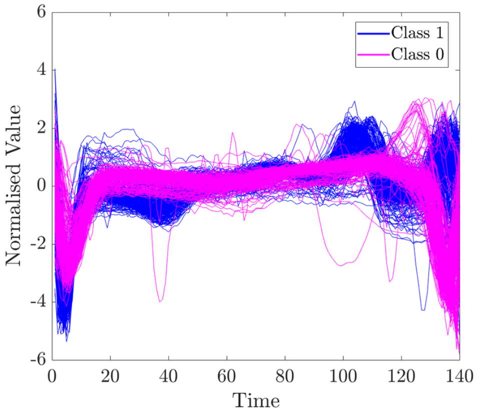

**Figure 7: Class 0 represents the anomaly heartbeat and class 1 the normal heartbeat for ECG dataset**

## 7.2 MSE distributions

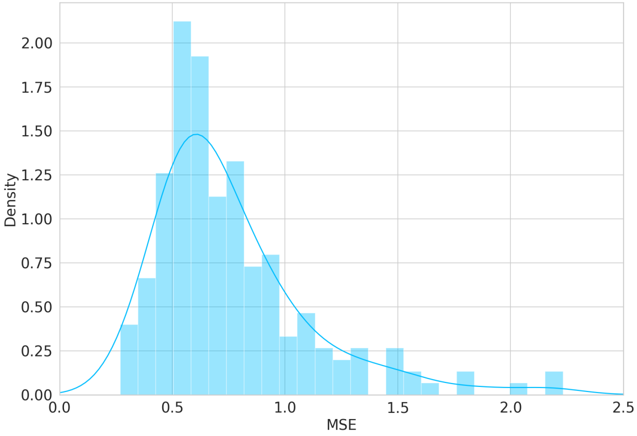

**Figure 8: MSE distribution for Bi-GRU model on normal green mobility data**

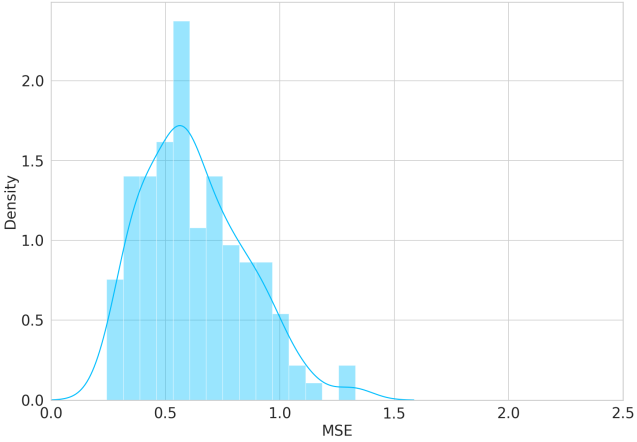

**Figure 9: MSE distribution for Bi-GRU model on anomaly green mobility data**

## 7.3 Lilliefor's test statistic distributions

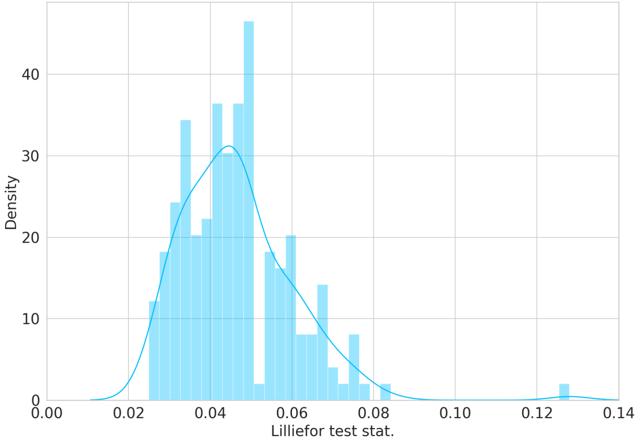

**Figure 10: Lilliefor's test statistic distribution for Bi-GRU model on normal green mobility data**

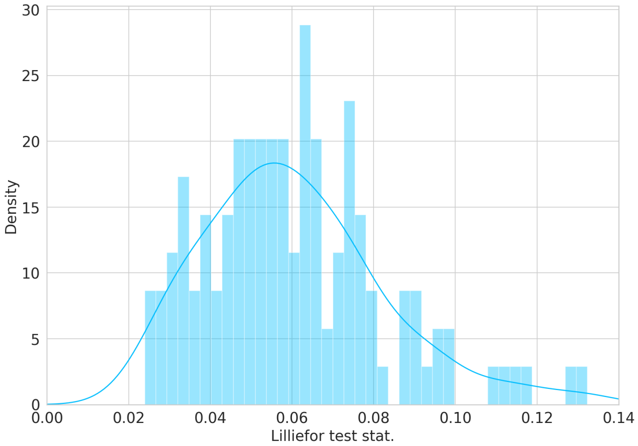

**Figure 11: Lilliefor's test statistic distribution for Bi-GRU model on anomaly green mobility data**

## 7.4 VrAE-BiGRU losses

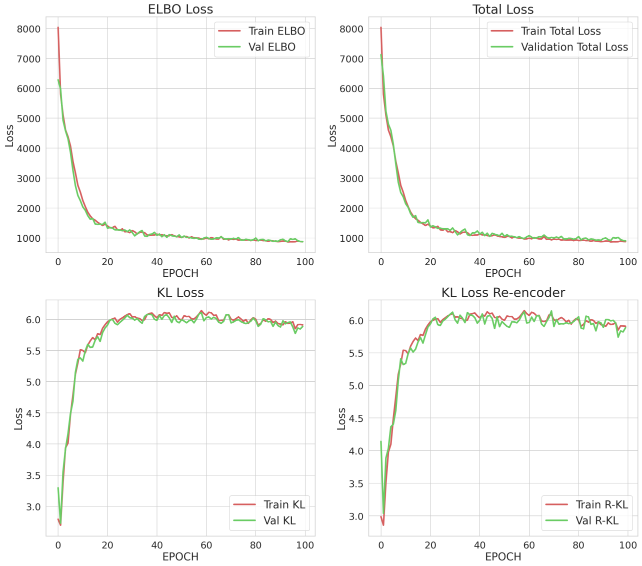

**Figure 12: Losses for ECG for the VrAE-BiGRU**

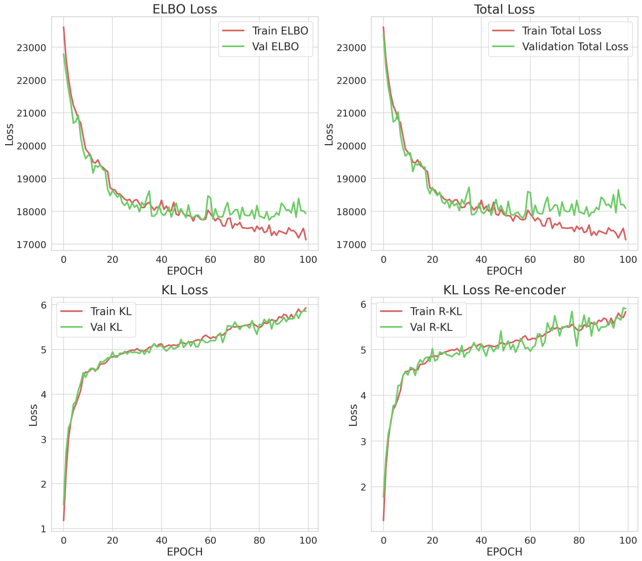

**Figure 13: Losses for Green Mobility for the VrAE-BiGRU**

## 7.5 LSTM Details

Figure 14 illustrates a single LSTM cell. $X_t$ is the input vector, $H_{t-1}$ is the Previous cell Output, $C_{t-1}$ is the Previous Cell Memory, $H_t$ is the Current cell output and $C_t$ is the Current cell Memory. The equations below indicates the various operations that take place in the LSTM cell.

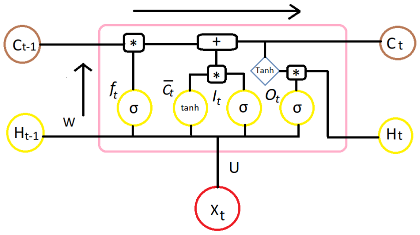

**Figure 14: LSTM unit architecture**

$$f_t = \sigma(X_t * U_f + H_{t-1} * W_f)$$
$$\bar{C_t} = tanh(X_t * U_c + H_{t-1} * W_c)$$
$$I_t = \sigma(X_t * U_f + H_{t-1} * W_f)$$
$$O_t = \sigma(X_t * U_o + H_{t-1} * W_o)$$
$$C_t = f_t * C_{t-1} + I_t * \bar{C_t}$$
$$H_t = O_t * tanh(C_t)$$

$W$ and $U$ are weight vectors for the various gates.

## 7.6 GRU Details

Figure 15 illustrates a single GRU cell. $x_t$ is the input vector, $h_t$ is the output vector, $\bar{h_t}$ is the candidate activation vector, $z_t$ is the update gate vector, $r_t$ is the reset gate vector. The below equations indicates the various operations that take place in the GRU cell.

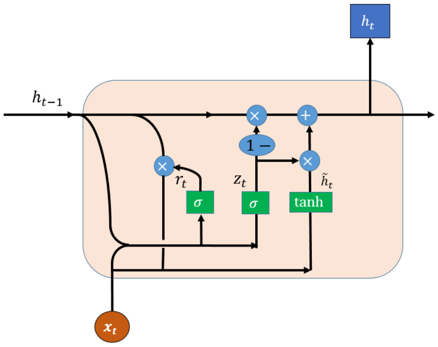

**Figure 15: GRU unit architecture**

$$z_t = \sigma(W_z * x_t + U_z * h_{t-1} + b_z)$$
$$r_t = \sigma(W_r * x_t + U_r * h_{t-1} + b_r)$$
$$\bar{h_t} = tanh(W_h * x_t + U_h * (r_t \odot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \bar{h_t}$$

$W$ and $U$ are weight matrices and $b$ is the bias vector.