TIME SERIES ANOMALY DETECTION WITH VARIATIONAL AUTOENCODER AND GRU

Nikolaj Espinoza s163848 — Clément E. Larsen s163669 — Jorge Sintes s202581

Technical University of Denmark
DTU Compute

ABSTRACT

This paper researches the potential of using variational autoencoders (VAE) and gated recurrent units (GRU) for anomaly detection on time series data. To do so various models combining these two architectures were tested on time series data collected by sensors installed on regular passenger cars, along with artificially generated signals. The results show that VAEs can be used to successfully detect anomalies and that the addition of GRU cells in the VAE architecture can greatly enhance this performance.

Index Terms— VAE, GRU, Variational Inference

1. INTRODUCTION

The concept of deviation from the norm became of interest to scientists as early as the XIXth century with the 1887 publication *On discordant observations* from Edgeworth [1]. The field of anomaly detection has since that time grown in interest, being facilitated by the advent of machine learning. A wide range of techniques has been developed over the years. One in particular, variational autoencoder (VAE) has proven its worth in various detection tasks [2]. Anomaly detection models based only on VAEs however often fail in detecting long-term anomalies. Recurrent neural networks (RNN), on the other hand, strive to capture features representing the dependence between data points, which is particularly interesting for time series data. One such widespread RNN is the LSTM, which was invented in 1997 [3], and has since then been used in various applications. In 2014, Cho et al. presented a new type of RNN, the GRU [4], which has been proven to be less computationally expensive than its counterpart from the XXth century. This paper will aim at testing whether VAEs can be successfully used to perform anomaly detection in time series data, and whether adding GRUs can benefit in this regard. This will be done using sensor data from Green Mobility (GM) cars to detect road anomalies. The proposed models will also be tested on artificial data to see how they perform on simpler signals.

2. BACKGROUND

2.1. VAE and Variational Inference

A Variational Autoencoder (VAE) is a type of autoencoder (AE). A basic AE is an artificial neural network that learns feature representations of data in a dimension different from the original dimension of the data (often lower). The goal of the AE is to encode the data into this new space referred to as the latent space and then reconstruct the original data from that encoding. For a given dataset the chosen encoding and decoding will thus be the ones giving the lowest reconstruction error in this process. The VAE differs from the AE by taking a more probabilistic approach. Instead of encoding an observation of the data into a point, it infers a probability distribution for it in the latent space, from which it then samples points that are passed through the decoder to obtain a probability distribution for each point of the original observation. This means that, contrary to the regular AE, where a reconstruction error is obtained, a reconstruction likelihood $p(X)$ is obtained instead (where X is the data trained on). The latter is more objective and principled for anomaly detection and does not require a threshold for classification purposes [2]. The goal of the VAE is to maximize this likelihood, i.e. maximize the integral in the equation

$$p(X) = \int p(X|z)p(z)dz, \tag{1}$$

where z is the latent variable. This integral is however intractable [5]. For most values of z, $p(X|z)$ will however nearly be 0, and thus not contribute much to the value of $p(X)$. A walk around here is therefore to find a function $q(z|X)$ giving a distribution over z values likely to produce X [6]. Chances are that the space of z values likely under that distribution will be a lot smaller than for those likely under the prior $p(z)$. A way to measure the similarity between these two distributions is the Kullback-Leibler divergence, denoted D_{KL}. Using the expression of $D_{KL}[q(z|X)||p(z|X)]$ together with Bayes' rule, the linearity of the expectation and properties of the log, one can get the following equation:

$$\log(p(X)) - D_{KL}[q(z|X)||p(z|X)] = E_{z\sim q(z|X)}[\log(p(X|z))] - D_{KL}[q(z|X)||p(z)] \tag{2}$$

Note here that, since the KL-divergence is always non-negative, maximizing the term on the right hand side of equation (2) by optimizing $q(z|X)$ is thus the same as minimizing the KL divergence on the left hand side [7]. Hence, the term on the right hand side is a lower bound for $\log(p(X))$, commonly called the evidence lower bound (ELBO). The ELBO noted $\mathcal{L}(q)$ is therefore defined as the objective function of the VAE, which the training will aim at maximizing. The first term in the ELBO measures the reconstruction quality while the KL-divergence provides regularization by forcing $q(z|X)$ to resemble the prior $p(z)$. A slight modification of the objective function yields the so called beta-Elbo:

$$\mathcal{L}_\beta(q) = E_{z\sim q(z|X)}[\log(p(X|z))] - \beta D[q(z|X)||p(z)] \quad (3)$$

where β is a hyperparameter allowing to tune the degree of regularization of the model. This more flexible version of the objective function was used in the presented work.

2.2. Gated Recurrent Unit (GRU)

GRU is a type of a recurrent neural network unit [4] that has a gating mechanism to remember dependencies in the data throughout the training phase. The mechanism runs the following calculations iteratively in each time step:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (4)$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (5)$$

$$h'_t = \tanh(W^{(h)}x_t + r_t \odot U^{(h)}h_{t-1}) \quad (6)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad (7)$$

Here, z_t is the update gate, r_t is the reset gate, h'_t is the current memory content, and h_t is the final output memory content, all at time t. W is the weights for the input, U is the weights for the hidden state, x_t is the current input, h_{t-1} is the hidden state from the previous time step, and $\odot$ is the element-wise multiplication operator.

The update gate (4) serves as a way to decide how much information from previous time steps should be kept and passed on. The reset gate (5) decides how much past information should be dropped and forgotten. The current memory content (6) stores what information to be kept from the past, based on (5), and (7) uses this along with (4) to calculate the final output.

GRU is often compared to LSTM [3] for its similar architecture. While LSTM has both a forget gate and a cell state to remember important features of previous information, GRU differs by only having a forget gate. Due to this, GRU often has a similar performance to LSTM, but with the added bonus of speed, due to less computations being done [8].

3. MODELS

This section will present the architectures of the models used to perform the anomaly detection. In the rest of the report the term feedforward neural network will designate a feedforward neural network with linear layers and *ReLU* activation functions unless otherwise stated. Furthermore all prior and posterior distributions, as well as all the probability distributions outputted by the decoder, will be a unimodal multivariate Gaussian distribution with diagonal covariance matrix.

3.1. Baseline

The chosen baseline model first uses GRU cells to capture the dependencies in the data and then passes the last hidden state through a feedforward network having a single neuron with a sigmoid in its output layer for classification purposes. The architecture is depicted in Figure 1.

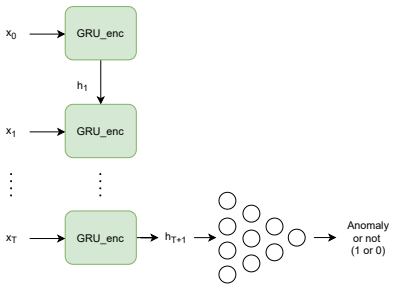

Fig. 1: Architecture for the baseline model.

3.2. VAE

The VAE is a basic VAE with a feedforward network in the encoder and decoder.

3.3. Annabelle

Annabelle will denote a type of VAE where the encoder is composed of a GRU cell which at each iteration takes one point x_t measured at time t from every fed time series, and where the last hidden state is split in two to obtain the parameters of the Gaussian posterior distribution in the latent space. As for the decoder, it remains the same as in a basic VAE. The architecture is shown in Figure 2.

3.4. Arthur

Arthur has the same encoder as Annabelle. The decoder is however composed of a GRU cell, followed by a feedforward network. When a sample is collected from the posterior, it is then split into two parts. One is fed to the decoder as the input of the first iteration, and has the same input size as the GRU cell of the encoder. The remaining part of the sample—which can be thought of as the ID of the encoded signal—is then fed as the initial hidden state of the cell. The outputted hidden state is then passed through a feedforward network to obtain the parameters of the Gaussian distribution of the corresponding point of the signal, whose mean is then fed as the

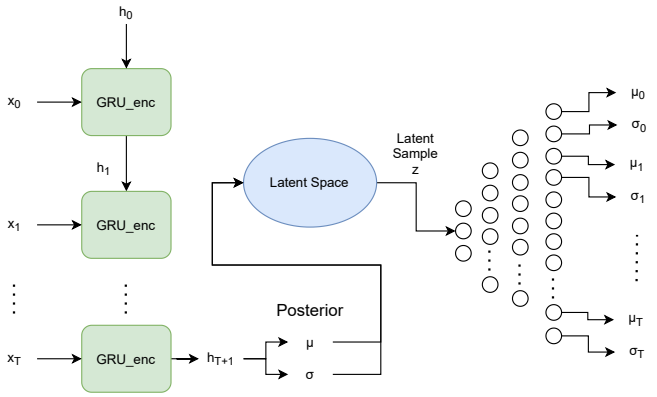

Fig. 2: Architecture for Annabelle.

input of the next iteration. The intuition behind this model came from *Sequence to Sequence Learning with Neural Networks* from Ilya Sutskever et al. [9] and *Generating Sentences from a Continuous Space* by Bowman et al.[10]. The model architecture is depicted in Figure 3.

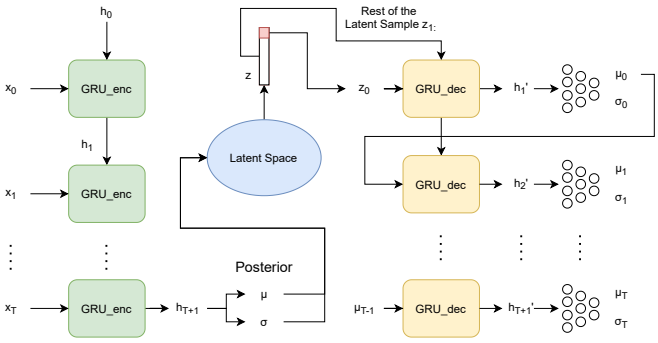

Fig. 3: Architecture for Arthur.

3.5. Betty

Betty is in essence trying to cope with the slow training of Arthur resulting from the feeding of the mean of the previous point as the input of the next GRU. This indeed requires the use of a for loop in practice, considerably slowing down the training. Here Betty is a form of vectorized alternative. The difference of the model lies in the decoder. Here the sample from the latent space is again split into two separate parts, one of which now has two times the dimension of the input size of the GRU encoder, and is fed as the initial hidden state of the GRU cell. While the remaining part of the sample—"ID of the signal"— is replicated sequence length times, and fed as an input to every GRU cell. This way the "ID" and the previous hidden state are passed over to the next iteration without the need of the extra for loop. Each hidden state outputted by the GRU cells is then split into two parts representing the means and the standard deviations of the corresponding input point(s).

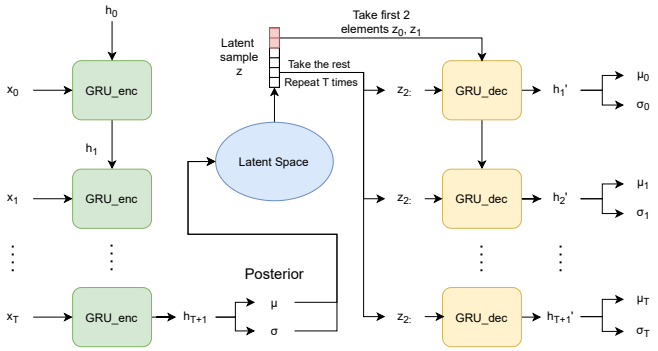

Fig. 4: Architecture for Betty, shown with input size of 1.

3.6. Classification method

All models except the Baseline are outputting a mean and standard deviation for the Gaussian distribution of each point of the inputted time series. These distributions allow to compute the log-likelihood of the entire signal for each observation. Knowing the ratio r of anomalous data in the training set, one can then pick the $(r \cdot 100)$th percentile of the log-likelihoods of the observations in the training set to be a threshold for classification [5]. Each point having a log-likelihood smaller than or equal to that threshold is then classified as being an anomaly and vice-versa.

4. RESULTS

The model was tested on data from a 3D accelerometer measuring the displacement of regular passenger cars—namely Renault ZOEs from the company Green Mobility—along three axes. Each observation is annotated with an international roughness index (IRI) measured by p79 vehicles from the Danish Road Directorate, on the M3 highway in Copenhagen. The vertical displacement measured by the sensor was however the only parameter considered in the presented work. The samples having an IRI of 2 or more were labelled as anomalies (approx. 3.5%), and the remaining ones as normal observations. Furthermore some of the data observations were interpolated, for all the time series to have the same length. Tests were also performed on artificially generated data, created to contain 5% of anomalies. (The data generation process is explained section 7.1 in the appendix). The two datasets were standardized using z-score normalization prior to training.

The models were then tested on the two datasets using cross-validations with 5 inner- and outer-folds, with the size of the latent space and the variable beta from the beta-ELBO as hyperparameters. The parameters selected in the inner-loop were set to be the ones giving the highest recall (see section 7.2 in the Appendix). As the final purpose of the model is to detect anomalies in road condition, it is arguably of higher interest to detect as many anomalies as possible, than miss-

classifying as little normal observations as possible. The average of the obtained results for the 5-folds are depicted in Table 1 and 2. One can see that for both the artificial and GM data, Arthur has the best accuracy, precision and F1-score, while the VAE dominates on the recall. It is also worth mentioning that Annabelle has a significantly higher F1-score on the GM data than the VAE, while being comparable for the sampled data. As for the baseline, its accuracy corresponds to the percentage of non-anomalous data, with a recall of 0.

Model	Accuracy	Precision	Recall	F1-Score
Baseline	0.95	-	0	0
VAE	0.972	0.6774	**1**	0.7981
Annabelle	0.975	0.7396	0.8255	0.7737
Arthur	**0.9870**	**0.8753**	0.8818	**0.8734**
Betty	0.9630	0.6421	0.7091	0.6694

Table 1: Performance measures of the results for all models on artificial signals.

Model	Accuracy	Precision	Recall	F1-Score
Baseline	0.9650	-	0	0
VAE	0.9261	0.3065	**0.8692**	0.4525
Annabelle	0.9698	0.5676	0.5848	0.5698
Arthur	**0.9722**	**0.6101**	0.6019	**0.5996**
Betty	0.9712	0.5876	0.5903	0.5852

Table 2: Performance measure of results for all models on GM data.

A ROC plot and precision vs recall plot were made for the GM data and are presented in Figure 5. The ROC plot clearly shows that the true positive rate is high for all thresholds and that the plot does not allow for a good differentiation of the models. The precision vs recall plot on the other hand shows that the GRU based VAEs (GRU-VAEs) i.e. Annabelle, Arthur and Betty, have a better precision to recall ratio than the basic VAE, for any chosen threshold. It is furthermore worth mentioning that the dots representing that threshold are very close to the optimal threshold.

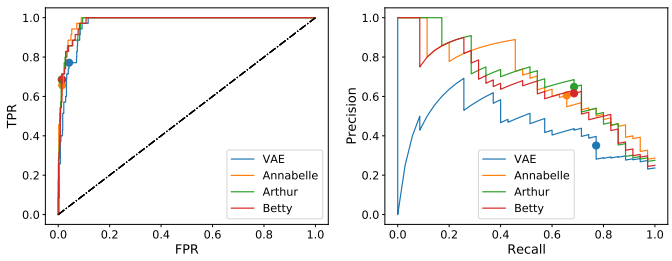

Fig. 5: ROC plot (left) and precision vs recall plot (right) for an outer-fold of the cross-validation. The dots correspond to the values obtained for the thresholds picked by the model.

5. DISCUSSION AND CONCLUSION

The obtained results show that the baseline simply classifies all the points as being normal. This is most likely due to the low percentage of anomalies in the datasets, which could be coped with by using data augmentation techniques. This however highlights a very important point, namely that all VAE based models can overcome this problem, clearly showing the added value of the VAE architecture. Regarding the rest of the models, it was noticed that the VAE obtained the best recall on both datasets. This however appeared to be at the expense of the precision, which was lower than in most of the other models. From a safety perspective finding as many anomalies as possible for road conditions should be preferred. From a financial perspective however, making sure that work force is mainly sent out to verify the road condition when there is actually an anomaly is preferred. In that regard, finding a trade-off between precision and recall is important. Taking this into consideration F1-score could be a more suited performance measure.

Looking at the ROC-plot in Figure 5, it can be seen that the curves for all the proposed models hug the y-axis, as one would typically like to see. The representation of classes in the GM dataset must be taken into consideration though. In this case there is a low percentage of anomalies ($\leq 5\%$), indicating a class imbalance. This means that as the models begin to classify more signals as anomalies, the true positive rate increases more rapidly than the false positive rate, even though the model isn't necessarily performing well. There is therefore not much information to be gained from looking at the ROC-plot, and it is much more interesting to look at the precision-recall plot instead. This plot gives more insight into the true classification performance of the models. Ideally, the curves will hug the top of the plot, giving a precision and recall of 1. The higher precision to recall ratio of the GRU-VAEs means that their guess of anomalies have a higher rate of being correct, when more anomalies are predicted. This indicates an overall better performance of the GRU-VAEs when compared to the VAE in this aspect, which points to the conclusion that adding GRU cells to the VAE does in fact help with detecting the anomalies. The points representing the recall and precision for the chosen log-likelihood thresholds are close to the optimal thresholds, showing the robustness of the models.

All in all, the results of this paper show that VAE architectures are a good tool for unsupervised anomaly detection problems, that are able to perform better than regular classification techniques. Furthermore, when working with time series data, the addition of GRUs, a newer and lesser known alternative to LSTMs, can be helpful in this detection. The results of adding GRUs to the VAE proved to be valuable and showed the benefits of using the two together for the purpose of anomaly detection.

6. REFERENCES

[1] F. Y. Edgeworth, "On discordant observations," The London, Edin-burgh, and Dublin Philosophical Magazine and Journal of Science, 2003, vol. 23, no. 5, p. 364–375.

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

[3] Sepp Hochreiter and Jürgen Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[4] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014.

[5] 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 *Proceedings of The 10th Asian Conference on Machine Learning*, Jun Zhu and Ichiro Takeuchi, Eds. 14-16 Nov 2018, vol. 95 of *Proceedings of Machine Learning Research*, pp. 97–112, PMLR.

[6] Carl Doersch, "Tutorial on variational autoencoders," 2021.

[7] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[8] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[9] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, "Sequence to sequence learning with neural networks," 2014.

[10] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew M. Dai, Rafal Jozefowicz, and Samy Bengio, "Generating sentences from a continuous space," 2016.

7. APPENDIX

7.1. Artificial data generation

The purpose behind having a self-generated artificial data-set is to create a well-known set of base signals. Different types of anomalies can then be added on top. This comes in great help in the beginning phases of model testing, as it becomes easier to evaluate their signal reconstruction, sensitivity when detecting anomalies...

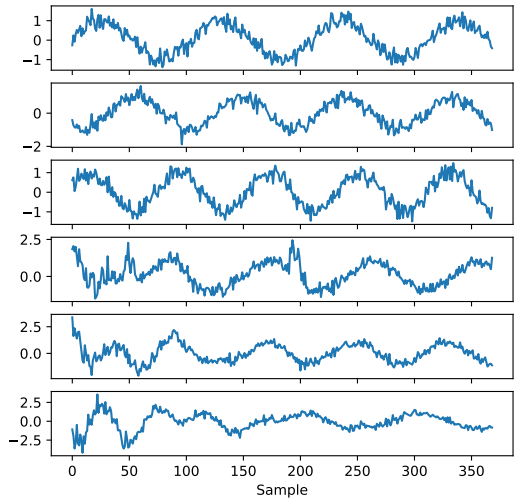

Fig. 6: 6 samples of the artificially generated data, last 3 including anomalies.

In the present paper, the data is created to resemble the GM signals. It consists of a set of 1000 sinusoidal signals with a varying frequency between $0.2 - 0.25$ Hz, all with the same sequence length as the GM ones. The signals are then shifted time-wise and some Gaussian noise is added on top, to recreate measurement noise.

In order to account for anomalies, a dampened sinusoidal signal is created, by multiplying a basic sinusoidal of 0.5 Hz with an inverse exponential. This signal is also shifted and then added, scaled by a random number between 0.5 and 1, to a base signal. This is supposed to recreate the behaviour of a bump in the road. On top of this, some random defects are added to the signal. Figure 6 shows some of the signals generated by this method.

7.2. Cross-validation parameters

All models were tested using 2-layer cross-validation with 5 inner and outer folds, with the size of the latent space and the variable beta from the beta-ELBO as hyperparameters. The optimal parameters chosen are the ones that maximize the recall. The preferred parameters of each model on the GM data are shown in table 3.

Model	Latent space size	β	No. epochs
VAE	6-8	0.5	30
Annabelle	2-4	0.5-1	30
Arthur	10-15	0.5-1-2	5
Betty	10	15-20-25	35

Table 3: Performance measure of results for all models on GM data.

7.3. Other proposed architectures

Several different architectures were proposed for GRU-VAEs when incorporating GRU cells in the decoder.

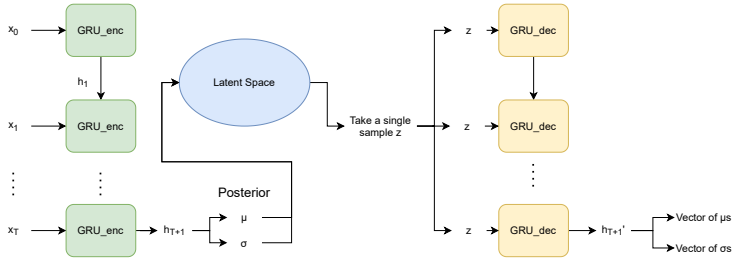

Fig. 7: Architecture for Barney.

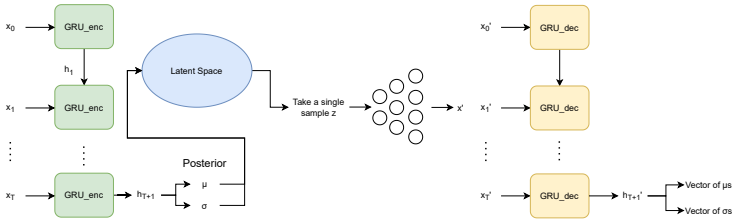

Fig. 8: Architecture for Claire.

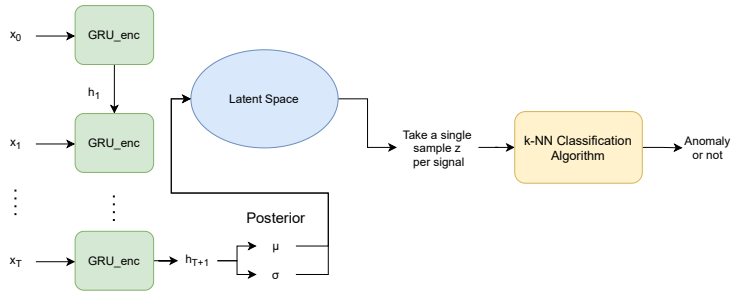

Fig. 9: Architecture for Charles.