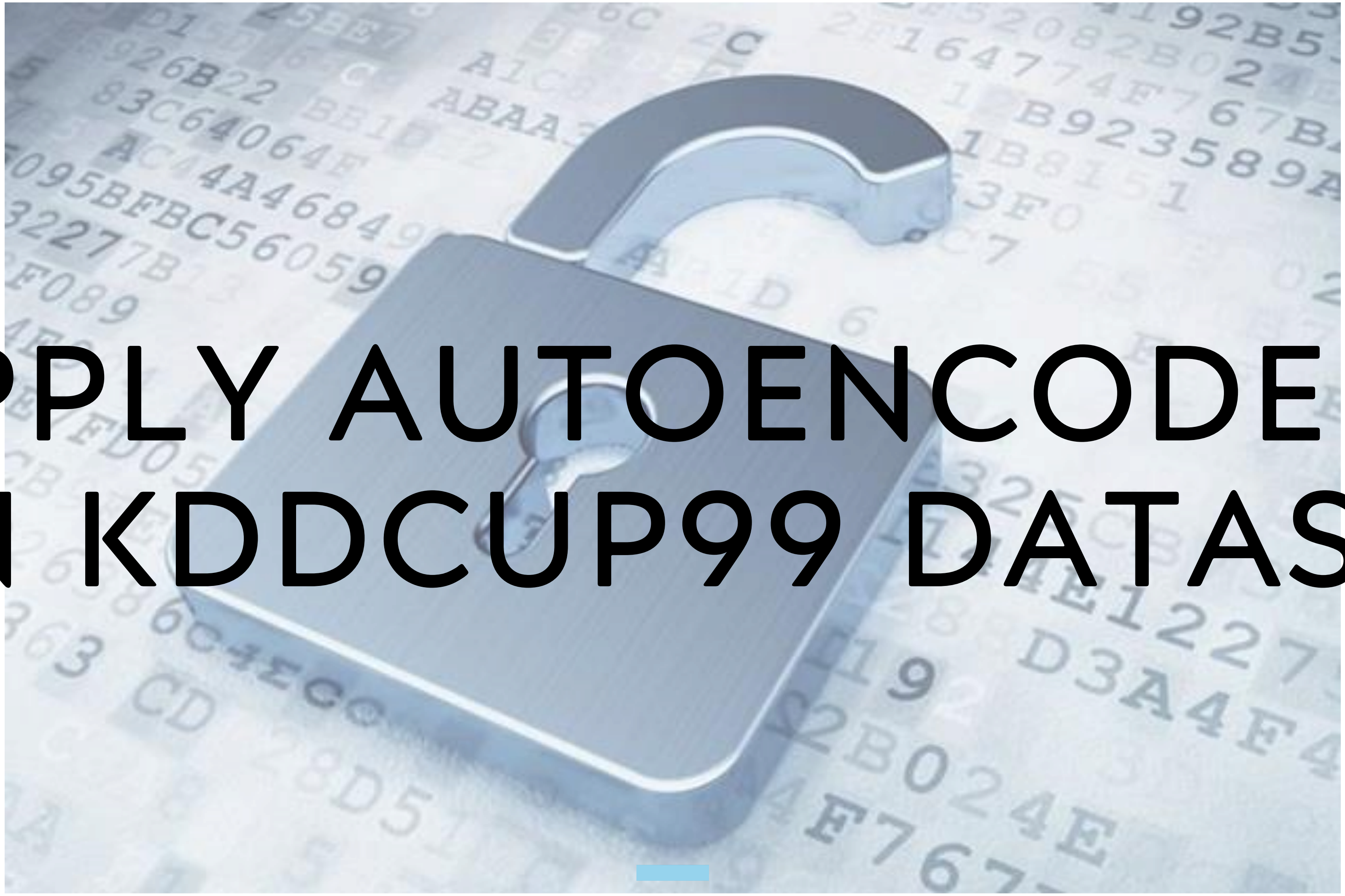


APPLY AUTOENCODERS ON KDDCUP99 DATASET

YIWEI YU, 2022 NOV

NETWORK INTRUSION DETECTOR

KDD Cup 1999 Data

- The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment.

Try the following approaches to detect the intrusions (anomaly detection):

1. PCA (Principal Component Analysis)
2. AE: Simple Autoencoder, Deeper Autoencoder
3. VAE (Variational autoencoders)

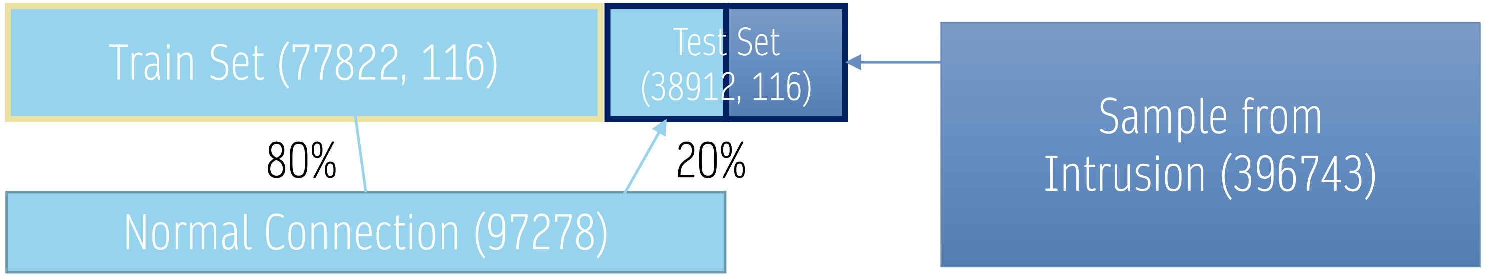
DATA PREPARATION

Download files [kddcup.data_10_percent.gz](data) and [kddcup.names](name of features) to build [Pandas Dataframe].

	duration	protocol_type	service	flag	src_bytes	dst_bytes
0	0	tcp	http	SF	181	5450
1	0	tcp	http	SF	239	486
2	0	tcp	http	SF	235	1337
3	0	tcp	http	SF	219	1337
4	0	tcp	http	SF	217	2032

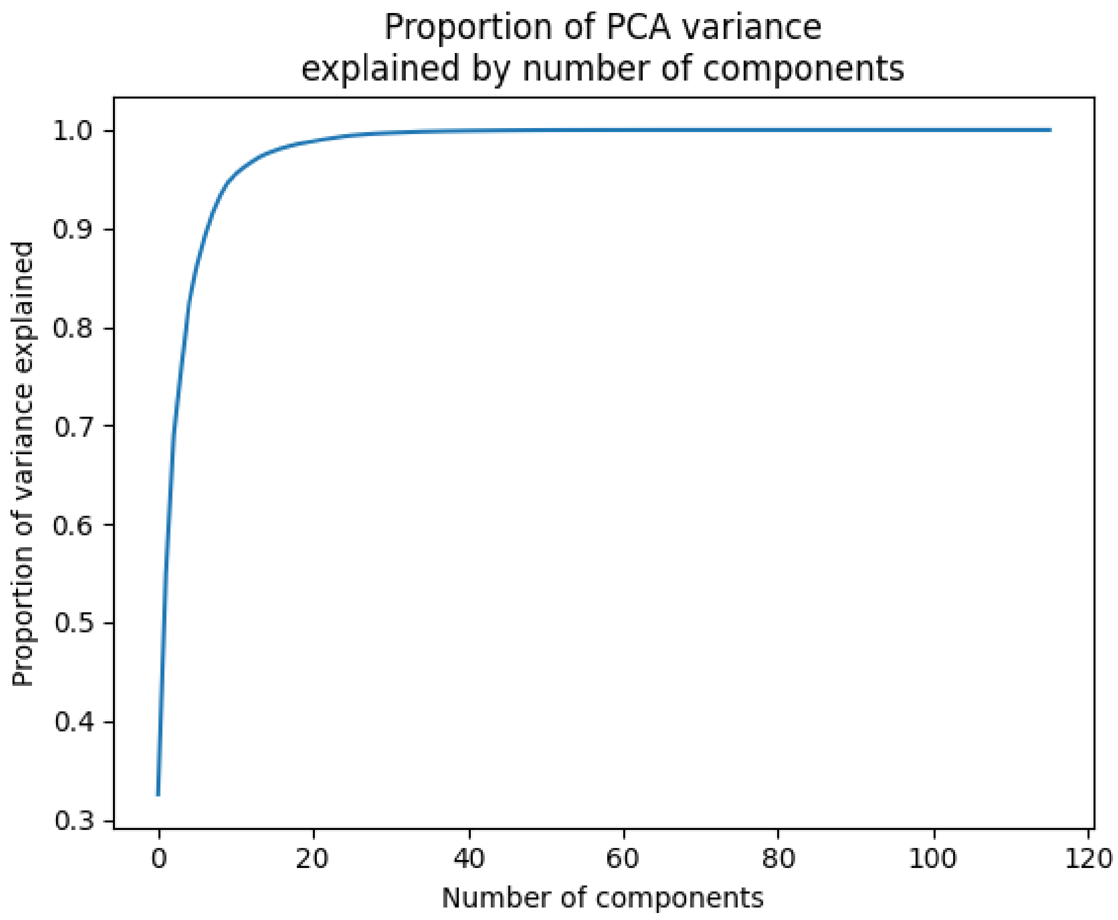
5 rows × 42 columns

- Remove columns which value is fixed, like 'num_outbound_cmds', 'is_host_login'.
- Transfer columns which contain different string value, to one-hot type, like 'protocol_type', 'service', 'flag'.
- Normalize numerical columns via [MinMaxScaler].



PCA (PRINCIPAL COMPONENT ANALYSIS)

- Take 8 components
  - Variance Explained = 0.92
  - MSE Reconstruction = 981.74
    - As the baseline while comparing results from AE and VAE.



SIMPLE AE (AUTOENCODER)

- Training
  - Epochs = 32
  - Batch size = 256
  - Accuracy = 93.77%
- Evaluation
  - Impressive outcome
- MSE Reconstruction = 680.99 (> Score on PCA)
  - Threshold = value at 99%

```
Model: "simple_ae"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 116)]             0

 encoder (Functional)        (None, 29)                3393
|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|
| input (InputLayer)        [(None, 116)]             0          |
|                                                                |
| encoder_out (Dense)       (None, 29)                3393       |
¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯
 decoder (Functional)        (None, 116)               3480
|¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯|
| decoder_in (InputLayer)   [(None, 29)]              0          |
|                                                                |
| reconstruction (Dense)    (None, 116)               3480       |
¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯
=================================================================
Total params: 6,873
Trainable params: 6,873
Non-trainable params: 0
```

Confuse Matrix

	precision	recall	f1-score	support
0.0	0.99	0.98	0.99	19456
1.0	0.98	0.99	0.99	19456
accuracy			0.99	38912
macro avg	0.99	0.99	0.99	38912
weighted avg	0.99	0.99	0.99	38912

DEEPER AE (AUTOENCODER)

- Training
  - Epochs = 32
  - Batch size = 256
  - Accuracy = 93.78%
- Evaluation
  - Impressive outcome
- MSE Reconstruction = 360.76 (> Score on Simple AE)
  - Threshold = value at 99%

```
Model: "deeper_ae"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input (InputLayer)          [(None, 116)]             0

 encoder (Functional)        (None, 29)                8497
|---------------------------------------------------------------|
| input (InputLayer)        [(None, 116)]             0         |
|                                                               |
| hidden-1 (Dense)          (None, 58)                6786      |
|                                                               |
| encoder_out (Dense)       (None, 29)                1711      |
-----------------------------------------------------------------
 decoder (Functional)        (None, 116)               8584
|---------------------------------------------------------------|
| decoder_in (InputLayer)   [(None, 29)]              0         |
|                                                               |
| hidden-2 (Dense)          (None, 58)                1740      |
|                                                               |
| decoder_out (Dense)       (None, 116)               6844      |
-----------------------------------------------------------------

=================================================================
Total params: 17,081
Trainable params: 17,081
Non-trainable params: 0
```

Confuse Matrix

	precision	recall	f1-score	support
0.0	0.99	0.98	0.99	19456
1.0	0.98	0.99	0.99	19456
accuracy			0.99	38912
macro avg	0.99	0.99	0.99	38912
weighted avg	0.99	0.99	0.99	38912

VAE (VARIATIONAL AUTOENCODER)

- Training
  - Epochs = 50
  - Batch size = 256

```
Model: "vae_mlp"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 encoder_input (InputLayer)  [(None, 116)]             0

 encoder (Functional)        (None, 29)                10208
|---------------------------------------------------------------|
| encoder_input (InputLayer)  [(None, 116)]            0        |
|                                                               |
| dense_7 (Dense)            (None, 58)                6786     |
|                                                               |
| z_mean (Dense)             (None, 29)                1711     |
|                                                               |
| z_log_var (Dense)          (None, 29)                1711     |
|                                                               |
| z (Lambda)                 (None, 29)                0        |
-----------------------------------------------------------------
 decoder (Functional)        (None, 116)               8584
|---------------------------------------------------------------|
| z_sampling (InputLayer)    [(None, 29)]              0        |
|                                                               |
| dense_8 (Dense)            (None, 58)                1740     |
|                                                               |
| dense_9 (Dense)            (None, 116)               6844     |
-----------------------------------------------------------------

=================================================================
Total params: 18,792
Trainable params: 18,792
Non-trainable params: 0
```

- Evaluation
  - Impressive outcome

Confuse Matrix

	precision	recall	f1-score	support
0.0	0.99	0.98	0.99	19456
1.0	0.98	0.99	0.99	19456
accuracy			0.99	38912
macro avg	0.99	0.99	0.99	38912
weighted avg	0.99	0.99	0.99	38912

- MSE Reconstruction = 863.42 (> Score on PCA)
  - Threshold = value at 99%

LATENT SPACE (ERROR TERM)

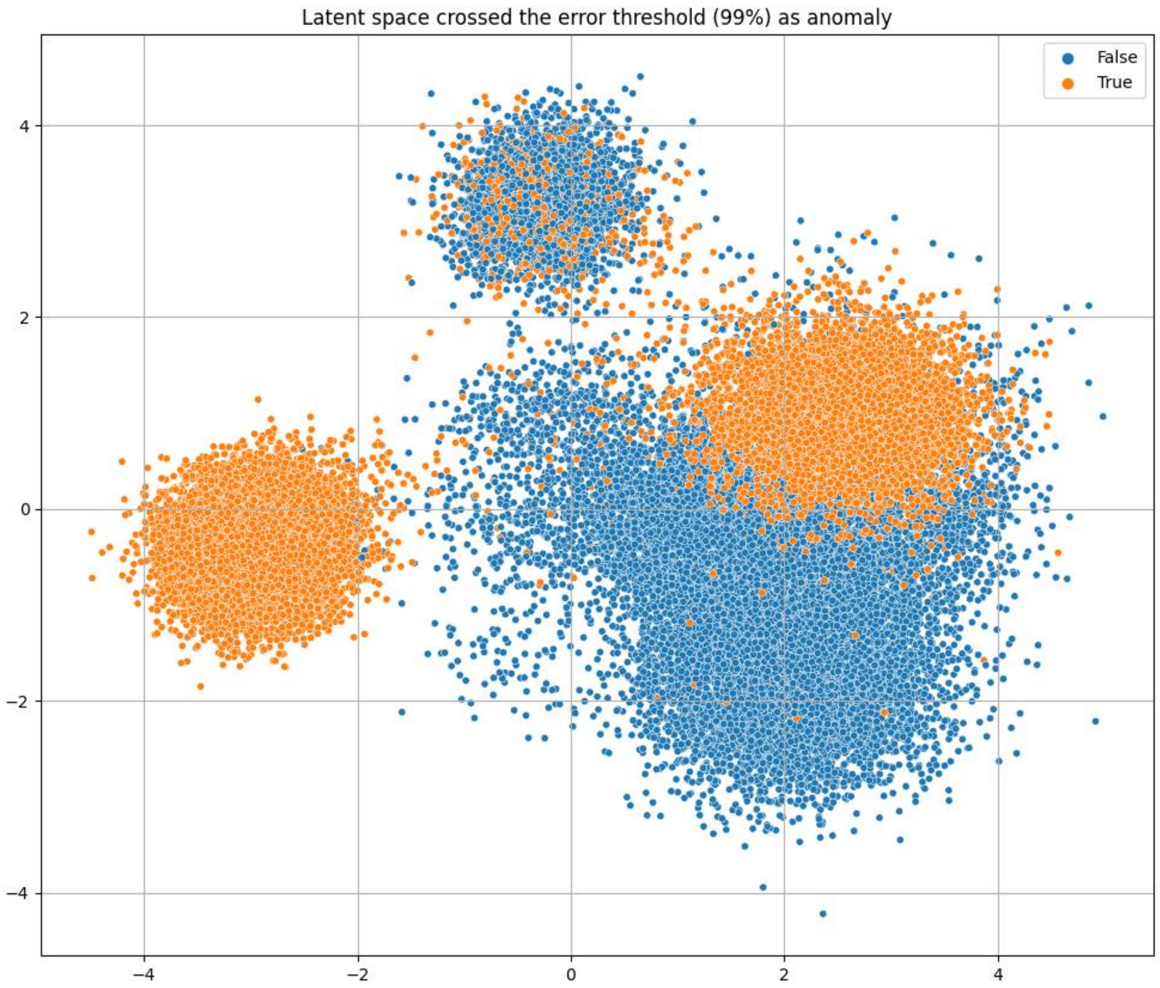
Examine the latent space generated by the encoder, by only using the encoder model without the decoder.

The image below shows a scatter plot of the latent space generated by the encoder (after dim reduction to 2 dims). The color of each point reflects its associated reconstruction error term.

A darker dot implies a larger error term. We can clearly see one large cluster of points that seem quite on the normal side (with a relatively small error term), surrounded by 3 main clusters with a relatively high error term.

LATENT SPACE (ANOMALY)

We can confirm this with the plot below which plots the same points above after marking each point that has crossed the error threshold as an anomaly (in orange).

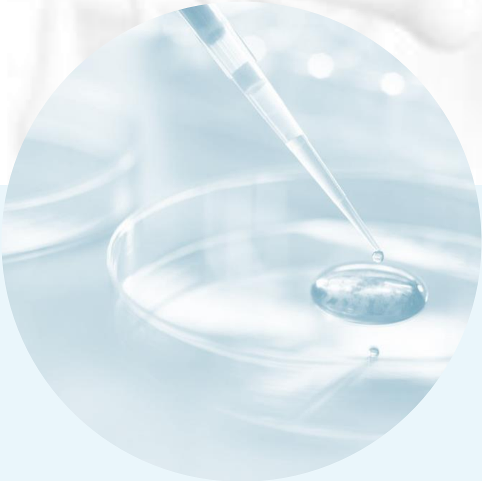


LATENT SPACE (GROUND-TRUTH)

Finally, we can compare the above to the ground-truth plot below which actually shows the true labels of the data. That is the points colored in orange in this plot are in fact anomalies — network packets sent during a network cyber attack. We can see that while we correctly identified the vast majority of the anomalies (98% of them), still there is a small group which we failed to identify, as the plot shows, probably because of some similarity to the normal points.

SUMMARY

Variational autoencoders are widely perceived as extremely effective for a variety of machine learning tasks. There is a lot of writing about variational autoencoders, but not too many practical examples in the areas of anomaly detection. The purpose of this post was to help to fill this gap by providing a simple example that can be used to prototype and test it. The greatest advantage of VAEs derives from the regularization imposed on the generated latent space. The ability to work with smoother and more continuous vector space can lead to more stable and accurate results, as it ensures similar data points lie closer together and makes similarity measures more reliable. After putting forward a simple architecture, I have shown how these qualities of VAE networks can generate pretty impressive results with relatively little tweaking, though I tried to highlight where tweaks and experimentation will be required if results are not satisfactory.



THANK YOU

[YUYW (GITHUB.COM)]