ABSTRACT

Autoencoders are a very popular approach in detecting anomalies in a system, where reconstruction error is generally used as an anomaly score. However, the reconstruction errors, generated in such manners, contain external noises of the system, making reconstruction errors as anomaly scores less effective. In this brief, we present an additional hypothesis that autoencoders may introduce additional statistical noise in the reconstruction errors as well.

Our proposal includes a design of an autoencoder, lays out a theoretical basis of designing a noise filter for reconstruction errors, and outlines various aggregation methods to reduce the effect of the noise. While further work is still needed, we are able to show the accuracy improvement by using various aggregation methods.

1. INTRODUCTION

Anomaly detection is a field of detecting outlier data indicating deviation from the normal behavior with generally some bad connotation. Anomalous events happen relatively infrequently but are disastrous in nature.

A popular approach in deep learning-based anomaly detection is to build a deep learning model, Autoencoder, to reduce the dimensionality of the data, and then reconstruct the input sample. An autoencoder belongs to a family of machine learning models, called neural networks, and more specifically deep neural networks. An autoencoder consists of an encoder, and a decoder.

And anomalous event is when anomaly score $A_n$ is more than a threshold $th$:

$$f = \begin{cases} 
0, & \text{normal if } A_n < th \\
1, & \text{anomaly if } A_n \geq th 
\end{cases}$$
This issue can be solved by A) properly training the autoencoder, B) by regularizing the training process, and C) by analyzing the data over a longer time etc.

Because of the noise as mentioned above, the anomaly detection system may be updated as the following:

2. AUTOENCODER ARCHITECTURE

The proposed autoencoder is inspired by the Wavenet architecture. It consists of an encoder and a decoder with a bottleneck layer and a self-attention layer. Unlike the Wavenet architecture, all time steps are preserved in the bottleneck layer.

* Input features
As in the baseline system, we use a log-mel-spectrogram of the input $X_n$
  - FFT calculation over 4096 or 8192 samples depending on the machine type, mentioned in the challenge.
  - Analysis frame size 64 ms
  - Log mel-band energy bands fn: 128 bands
  - Input time-steps to the autoencoder ts: 4 or 32 Analysis frames depending on the machine type.

* Hyperparameters
The following are the additional model parameters apart from ts and fn
  - Number of layers $M$
  - Number of bottlenecks $b_n$
  - Use machine ID or not at the bottleneck layer.

3. NOISE FILTERS AND AGGREGATION METHODS

Developing effective noise filters are dependent on deployment and other factors. For lack of time, we have not investigated the effective noise filters, and ML based approaches to noise filtering yet. However, following anomaly scores are evaluated and manually selected.

A. Calculate the Mean Square Error over the complete sample (default and state-of-art)

$$MSE = \frac{1}{N} \sum_{1}^{N} (\bar{X}_n - X_n)^2$$

Where N is the number of the frames per sample.

B. Calculate the Median Square Error – this allows to filter sudden onset of the excessive noise for a short duration.

$$Median((\bar{X}_n - X_n)^2)$$

C. The proposed error E1 calculation which involves adding the frequency mel bands over all the frames per sample, and then calculating MSE.

$$E1 = \frac{1}{NL} \sum_{1}^{L} \left( \sum_{1}^{N} \bar{X}_n - X_n \right)^2$$

Where N is number of frames per sample, and L is the number of the log-mel-bands. This is helpful if the present noise in the system behaves like the white noise, and cancels itself over the period of time, resulting in only minor left-over noise in all of the bands.

D. The proposed error E2 calculation which involves adding frequency bands over all of the frames, and then calculating the mean absolute error (MAE) over all frequency bands.

$$E2 = \frac{1}{NL} \sum_{1}^{L} \left\| \sum_{1}^{N} \bar{X}_n - X_n \right\|$$

Where N is number of frames per sample, and L is the number of the log-mel-bands. This is helpful if the noise present in the system behaves like white noise and cancels itself over a period of time, and still there is significant left-over noise in one or more of the bands.
4. RESULTS

Baseline results are generated after running for 100 epochs.

Table 1. Baseline results

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>AUC</th>
<th>pAUC</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToyCar</td>
<td>78.77</td>
<td>67.58</td>
<td>MSE</td>
</tr>
<tr>
<td>ToyConveyor</td>
<td>72.53</td>
<td>60.43</td>
<td>MSE</td>
</tr>
<tr>
<td>Fan</td>
<td>65.83</td>
<td>52.45</td>
<td>MSE</td>
</tr>
<tr>
<td>Pump</td>
<td>72.89</td>
<td>59.99</td>
<td>MSE</td>
</tr>
<tr>
<td>Slider</td>
<td>84.76</td>
<td>66.53</td>
<td>MSE</td>
</tr>
<tr>
<td>Valve</td>
<td>66.26</td>
<td>50.98</td>
<td>MSE</td>
</tr>
</tbody>
</table>

All the machine types were trained for 500 epochs with early stopping patience of 50 epochs, and run on the test data.

Table 2. Accuracy results using proposed scheme

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>AUC</th>
<th>pAUC</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToyCar</td>
<td>95.64</td>
<td>85.99</td>
<td>E2</td>
</tr>
<tr>
<td>ToyConveyor</td>
<td>86.52</td>
<td>70.81</td>
<td>E2</td>
</tr>
<tr>
<td>Fan</td>
<td>86.71</td>
<td>70.58</td>
<td>E1</td>
</tr>
<tr>
<td>Pump</td>
<td>88.71</td>
<td>72.04</td>
<td>E1</td>
</tr>
<tr>
<td>Slider</td>
<td>92.36</td>
<td>76.11</td>
<td>E1</td>
</tr>
<tr>
<td>Valve</td>
<td>88.61</td>
<td>75.34</td>
<td>MSE</td>
</tr>
</tbody>
</table>

Table 3. Improvements

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>AUC</th>
<th>pAUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToyCar</td>
<td>16.87</td>
<td>18.41</td>
</tr>
<tr>
<td>ToyConveyor</td>
<td>13.99</td>
<td>10.38</td>
</tr>
<tr>
<td>Fan</td>
<td>20.88</td>
<td>18.13</td>
</tr>
<tr>
<td>Pump</td>
<td>15.82</td>
<td>12.05</td>
</tr>
<tr>
<td>Slider</td>
<td>7.6</td>
<td>9.58</td>
</tr>
<tr>
<td>Valve</td>
<td>22.35</td>
<td>24.36</td>
</tr>
</tbody>
</table>

5. REFERENCES

Search and Data Mining (WSDM ’20). Association for Computing Machinery, New York, NY, USA, 894–896. DOI:https://doi.org/10.1145/3336191.3371876