# Real-time Anomaly Detection for Multivariate Data Streams

AMLTS22: Applied Machine Learning Methods for Time Series Forecasting


## Kenneth Emeka Odoh

https://kenluck2001.github.io


October **21**, 2022

# Bio

- **Software Engineer** @ Microsoft.
  - About me: https://kenluck2001.github.io
- Authoring a **textbook** on Distributed Systems.
  - Link: https://kenluck2001.github.io/blog_post/authoring_a_new_book_on_distributed_computing.html
- Made significant **open source contributions** to a number of popular Software packages.
- Prolific technical **blogger**
  - Link: https://kenluck2001.github.io/blogs/1.html

**Paper:** https://arxiv.org/abs/2209.12398
**Blog:** https://kenluck2001.github.io/blog_post/real-time_anomaly_detection_for_multivariate_data_stream.html

# Anomaly Detection

This is the task of classifying patterns that depict abnormal behaviour.

**Anomaly detectio**n is well-suited for unbalanced data, where the ideal scenario is to predict the behaviour of the minority class.

Categorization of different anomaly types:

- Point anomaly
- Contextual Anomaly
- Collective Anomaly

Anomaly detection algorithms can operate in many settings:

- Static (batch)
- Online (real-time)
- Static + Online

Anomaly detection algorithm can work in **modes** which include:

- **Diagnosis method** finds the outlier in the data and removes it from the data sample to avoid skewing the distribution.
  - It is suitable when the distribution of expected behaviours is known.
  - The outliers get excluded when the estimating of the parameters of the distribution [3].
- **Accommodation method** finds the outliers in the data and incrementally re-estimating the parameters of the statistical model .
  - It is suitable for data streams that account for the effect of concept drift [2].

# Data Stream

- Time and space constraints.
- Online algorithms
  - Detecting concept drift.
  - Forgetting unnecessary history.
  - Revision of model after significant change in distribution.
- Time delay to prediction.

# Core Contributions

Probabilistic Exponentially Weighted Moving Average (**PEWMA**) was originally developed for online anomaly detection on **univariate** time series. See our paper for reason why **PEWMA** serves as improvements of Exponential Weighted Moving Average (**EWMA**).

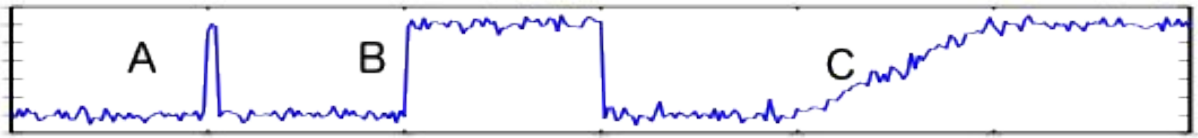


Figure 1: abrupt transient shift, abrupt distributional shift, and gradual distributional shift [1] labeled as "A", "B", and "C"

We provide extensions to support real-time anomaly detection of a **multivariate** data stream as follows. Our work involved a designing a few formulations as shown:

- Online Covariance Matrix in Section 3.2 of our paper.
- Online Inverse Covariance Matrix in Section 3.3 of our paper.
- Setting threshold  on Z-score on the PDF as shown in Section 3.4 of our paper.

# Online Multivariate Anomaly Detection

- Calculate incremental covariance and inverse covariance matrix.
- Make a running average.
- Use a Bayesian-like update method where the next prior was the previous posterior and update parameters as appropriate.
- Select a threshold and compare to the calculated p(x) and identify anomalies in the data stream.

(1) Use the covariance matrix, $C_{t+1}$ and inverse covariance matrix, ${C_{t+1}}^{-1}$.

(2) We increment the mean vector, $\mu$ as new data arrives. It is possible to simplify the Covariance matrix, $C$, which will capture a number of the dynamics of the system. Let $n$ represent the current count of data before new data has arrived. Also, $\hat{x}$: is the new data, $\mu_{t+1}$: moving average as shown in Equation 14.

$$\mu_{t+1} = \frac{(n * \mu_t) + \hat{x}}{n + 1} \tag{14}$$

(3) Set a threshold to determine the acceptance and rejection regions. Items in the acceptance region are considered to be normal behavior as shown in Equation 15.

$$p(x) = \frac{1}{\sqrt{(2\pi)^m |C|}} \exp\left(-\frac{1}{2}(x-\mu)^T C^{-1} (x-\mu)\right) \tag{15}$$

Where $\mu$ is mean vector, $C$ is covariance matrix, $|C|$ is the determinant of $C$ matrix, $x \in R^m$ is data vector, and $m$ is the dimension of $x$ respectively.

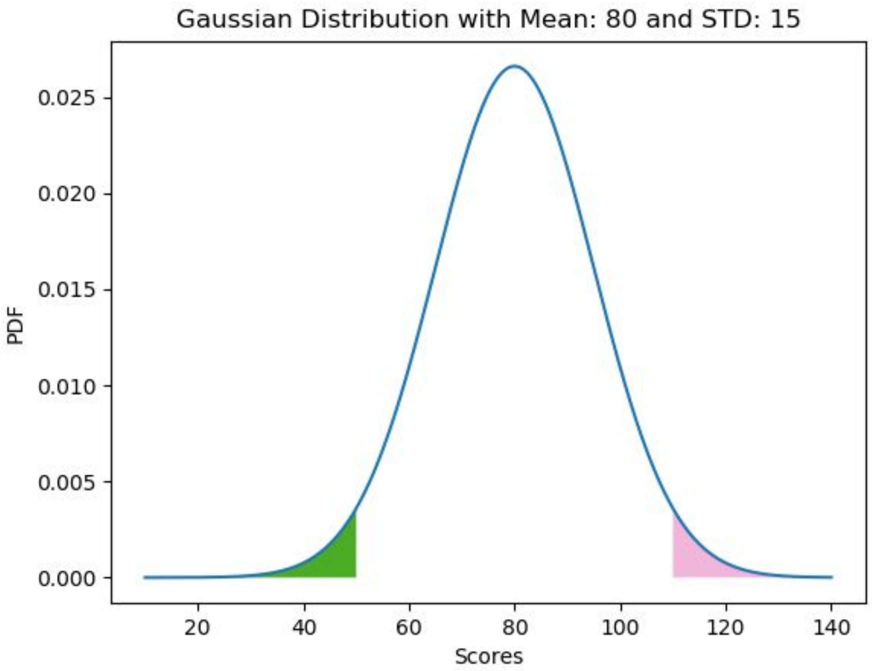




Figure 2: Deciding threshold on Normal Distribution Curve (mean ± 2 * std)

# Result Analysis

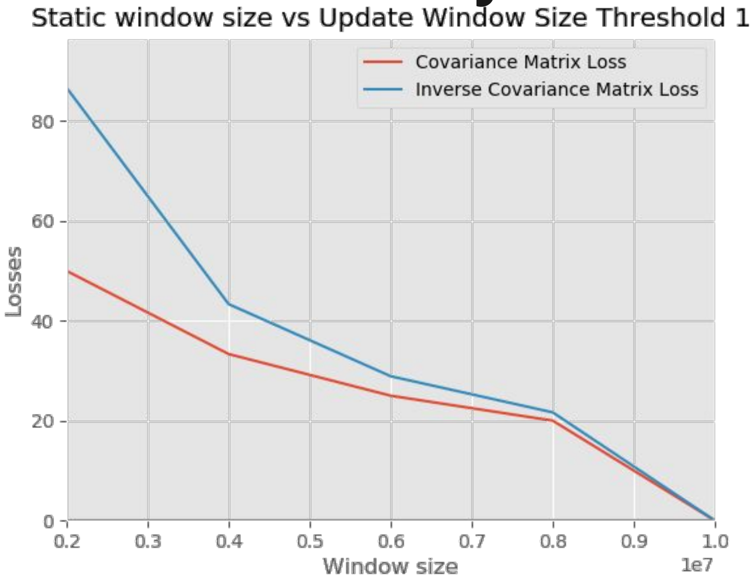


Figure 3: Compare the threshold of static vs incremental impact performance of anomaly detection

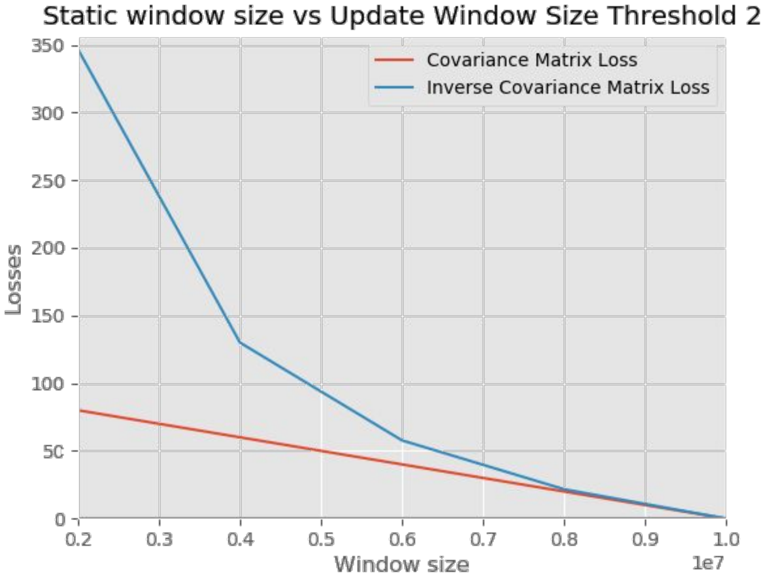


Figure 4: Compare the threshold of static vs incremental impact performance of anomaly detection (Version 2)

# Conclusions / Future Work

- For anomaly detection to work properly.
    - Highly informative features must be chosen that captures  the dynamics of the system

# Limitations

- Strong assumption on Gaussian distribution.
- Inability to handle non-stationarity distributions in data streams.

# References

1. Kevin M. Carter and William W. Streilein. 2012. Probabilistic reasoning for streaming anomaly detection. In Proceedings of the Statistical Signal Processing Workshop. 377–380.

2. Gregory Ditzler and Robi Polikar. 2013. Incremental Learning of Concept Drift from Streaming Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 25, 10 (2013), 2283–2301.

3. Victoria Hodge and Jim Austin. 2004. A Survey of Outlier Detection Methodologies. Artificial Intelligence Review 22, 2 (2004), 85–126.