Anomaly Detection on the High Throughput Network of the ATLAS TDAQ System

Mitchell Phiri, Simon H. Connell, Pathmanathan Naidoo
Department of Mechanical Engineering Science
University of Johannesburg
Johannesburg - 2006, South Africa
201215553@student.uj.ac.za, shconnell@uj.ac.za, naidoop@uj.ac.za

Mikel E. Pozo Astigarraga
ATLAS Trigger and Data Acquisition
European Organisation for Nuclear Research
Meyrin - 1217, Switzerland
eukeni.pozo@cern.ch

Dave Nicholls
NESCA Board
South African Nuclear Energy Corporation
North West Province - 0240, South Africa
nicholdr@iburst.co.za

Abstract

As the volume of data recorded from systems increases, there is a need to effectively analyse this data to gain insights about the system. One such analysis requirement is anomaly detection. Data-driven approaches such as machine learning, are by construction, able to learn (to some degree) the underlying representations in the data and consequently identify a hyperplane which separates the normal point states from the anomalous ones. In most cases the data is not linear in the parameter space, does not possess apparent trends or periodic seasonality and is noisy. In this work, we develop models for anomaly detection analysing data obtained from the networking devices of the ATLAS Data Acquisition System (comprising approximately 10 000 interfaces polled at 30 seconds intervals). The selection of algorithms was based on robustness and interpretability of the models. Ultimately, the deep learning architectures as well as those inspired by biological networks and those that employ transformations that linearise the measurement space were chosen. Preliminary results indicate that we are able to model the system to some degree and the anomaly detection solution is generic for a multiple parallel suite of time series data, somewhat independent of its origin. As such these concepts and results are also applicable to the energy space, for example, monitoring data streams from a power station. Successful development would imply new insights into how anomalies occur in a system and/or when they will occur and would allow for in-depth analyses such as Root Cause Analysis. The combination of an interpretable model and Root Cause Analysis would lay foundations for developing a Reinforcement Learning based system in which the system could take active decisions on certain anomaly encounters.

Keywords: Anomaly Detection, Deep Learning, Hierarchical Temporal Memory, Koopman Operator

1. Introduction

The Large Hadron Collider (LHC) (Figure 1) located at the European Organization for Nuclear Research (CERN) is a circular particle accelerator providing proton to proton collisions at approximately forty million times per second (Aad *et al.*, 2008; Pozo Astigarraga, 2017). The protons circulate around the accelerator and interact at several points along the LHC ring where different particle detectors study the results of these interactions. ATLAS (A Toroidal LHC ApparatuS) (Figure 2) is one of the general purpose detectors and Trigger and Data Acquisition (TDAQ) is responsible for the transport and storage of the selected complex event data resulting from the collisions (Aad *et al.*, 2008; Pozo

Astigarraga, 2014; The ATLAS Collaboration, 2019). These selected events are transported to a data logging system for final packaging and transfer to permanent storage at approximately 2 GB/s following an input rate of 160 GB/s (Aad *et al.*, 2008; Pozo Astigarraga, 2014; The ATLAS Collaboration, 2019). In order to maximize efficiency and minimize downtime the system must be actively monitored.

Figure 3 illustrates the computer network of the ATLAS TDAQ system which is responsible for these high throughput data transfers between separate data centers located in the underground area of the experiment (USA 15) and on the surface 100 meters above (SDX 1). During the next LHC operational phase (2021-2024), data flow of several tens of Gigabytes per second is expected during data taking periods (Leahu, 2013; Collaboration, 2017; Pozo Astigarraga, 2018; Vandelli and Collaboration, 2019). These rates are expected to increase by two orders of magnitude in future upgrades and under these loads, the network needs to work efficiently, and a lot of effort is made to ensure that there are no downtimes due to hardware failures or system congestion (Pozo Astigarraga, 2014, 2018; Collaboration, 2017). This opens an avenue for developing a new monitoring system using smart techniques for anomaly detection such that the monitoring of the continuous flow of interface parameters (at least 10 metrics for around 10 000 interfaces polled every 30 seconds with the possibility of an increase in the polling frequency) is possible. In addition to this, the system should be able to differentiate between traffic anomalies considered as normal (for example, a detector resynchronization operation that will stop the data flow for a few seconds) and anomalies pointing to misbehaviour in the network and hence requiring the intervention of a system engineer.

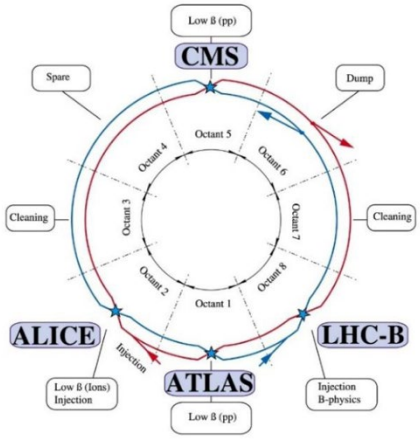

Figure 1: Layout of the LHC, and its four major experiments

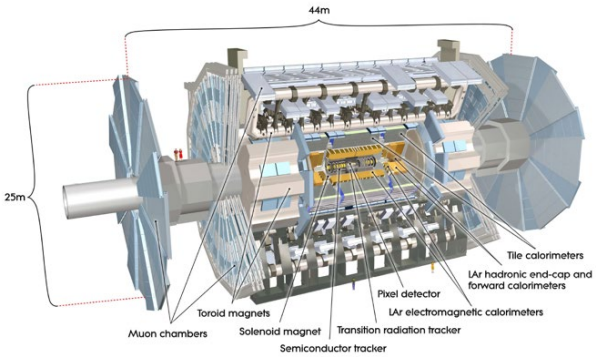

Figure 2: Cut-away view of ATLAS detector

For the problem at hand the data to be used is gathered from the interfaces and the measured parameters per interface include but are not limited to (for the flow into and out of the device) the load, the number of packets, the number of octets, the number of errors, the number of broadcasts and the number of discards. Due to the volume of devices being

polled, a window of 30 seconds is allowed between successive polling calls such that each device is polled twice per minute resulting in two measurements. One problem that becomes immediately apparent is that for one single polling window there exists a time difference of at most 30 seconds between the first and last interfaces polled. The other problem is that the poller is not guaranteed to poll all devices in a single 30 second window, but it will poll the same port twice in the subsequent 30 second window if it was missed in the preceding polling instance. This introduces asynchronicity in the data which poses a new challenge when carrying out tasks that require grouping of data based on a metric such as time or using the timestamp for different purposes.

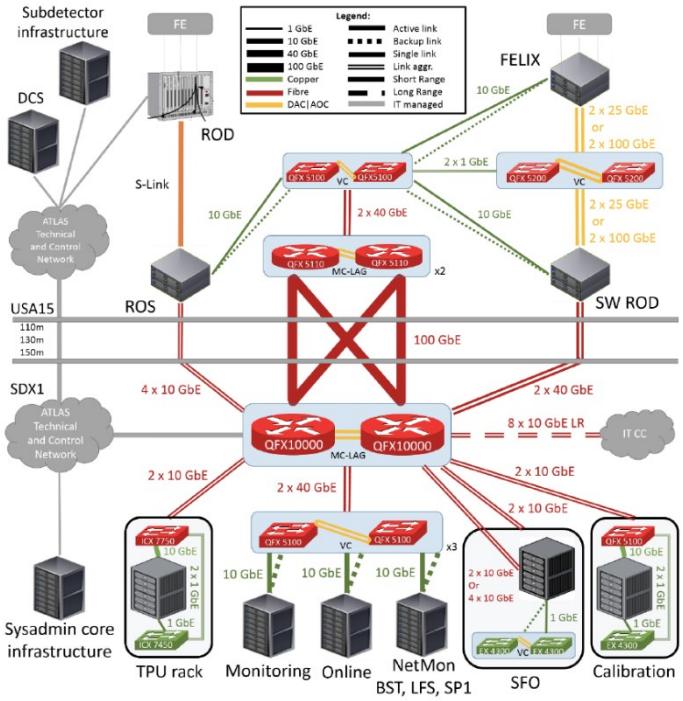

Figure 3: Overview of the TDAQ Physical Network

The collected data can be thought of as existing in a hyperspace of $n + 1$ dimensions where n is the number of metrics recorded and the additional dimension comes from the timestamp. We note that because the time is continuous, new data entries are added to the span of this hyperspace and that there exists a concept of clustering data in the spatial and temporal context. The data may also possess seasonal variations and trends arising from the operational and non-operational modes of the network. The asynchronicity problem introduces a complexity in that for the clustering of points (this clustering from a high level introduces a hyperplane separating normal and anomalous) to be successfully carried out there needs to be a regularity in the time dimension (that is, the recording of data points or their analysis thereof should happen at a single point in time) so that points from a single polling window can be accurately compared. The continuity of time also suggests that there exists two types of anomalies in the data which are spatial (anomalous with respect to the measurement domain of the data) and temporal (anomalous with respect to a finite closed time window but normal otherwise).

In order to gain more insights into the problem we developed a simple statistical model which calculated for each metric a rolling mean using the preceding 10 data points and consequently the threshold based on this mean as a fraction of the mean. We obtained a bounded operational window with bounds for lowermost and uppermost values in the range *[mean - threshold; mean+ threshold]* with values falling outside this range as anomalies. We tested this as a solution, and we noted that the model was over-predicting the number of anomalies. This statistical model, though not good enough for making predictions illustrated the general requirements which are that; the model should be predictive, there is a concept of statistical significance of the outliers, and there is a requirement of a threshold for classification. This model also indicated where weak points may occur (that is, the predictive power of the model is heavily decreased) and that the solution depends on several parameters such as the efficiency of the model at estimating the next value as well as the assignment of the local significance of that value, and the threshold. We also note that although the algorithm does adapt its mean to the values around a local domain, it is not able to capture

seasonal variations (underlying normal variation) or erratic behaviour which is appears anomalous but is not the kind of anomaly searched for.

The prediction of the normal could range from a simple statistical regression algorithm where several points from the past generate a forecast of the next point with the addition of some random variable (noise from some distribution). Alternatively, it could be done with a sophisticated approach which can learn the underlying normal variations and their noise distributions and then classify the points in the future. This paper aims to present our findings to that end as carried out on the ATLAS TDAQ Network highlighting the following contributions:

- Providing an intuitive way of handling hierarchical data streams, that is, input pipelines (properly batching the data for pre-processing and piping it to an algorithm continuously (streaming)) and output (yielding results from evaluation and post processing them into groups for classification).
- Showing that the algorithms presented in this paper generalize to problems from different domains.

2. Review of Some Relevant Work

In building systems, a realistic measurement of the system performance is necessary to determine whether a system is operating at maximum efficiency. Anomaly detection is a tool aimed at detecting operational inefficiencies in a system by identifying and highlighting parts of that system which operate outside the scope of what is regarded as normal at a particular instance or a defined range of time (Ahmad *et al.*, 2017). More generally, the ubiquity of sensors means that large amounts of data are easily collected and because of this we can capture patterns (that is, seasonal, weekly or daily trends etc. The granularity can be as far down to the accuracy of the order of fractions of seconds) in the data-stream(s) which aid in determining whether a point is anomalous or not. The volume of data usually involved in the anomaly detection process requires a lot of computation and a good model for representation and as such traditional machine learning and deep learning based methods offer an avenue for handling big data, and they have been shown to work well for multivariate systems (Song *et al.*, 2017; Chalapathy, Toth and Chawla, 2018; Shih, Sun and Lee, 2018; Wielgosz *et al.*, 2018; Gong *et al.*, 2019; Wyszynski and Pozo Astigarraga, 2019; Lin *et al.*, 2020). Other methods that work well include biologically inspired models for example the Hierarchical Temporal Memory (Cui, Ahmad and Hawkins, 2016; Ahmad *et al.*, 2017). This section serves to highlight some relevant work being done for complex systems somewhat related to anomaly detection and/or simplifying the anomaly detection process.

2.1 Machine Learning

The sub-branch of Artificial Intelligence (AI) in which computers progressively learn from data is known as Machine Learning (ML) (Chollet, 2018). For simplicity purposes we can think of a dataset ($X: y \longmapsto$ where X is the data used to determine y and y is the value to be predicted) as being composed of features (X) and labels (y). This dataset is split into training and testing data (in the form X = $X_{train} + X_{test}$and $y = y_{train} + y_{test}$ in the ratio 70:30 typically) and the machine learning algorithms are subjected to training with the dataset $\{X_{train}, y_{train}\}$ in which the models learn to minimise the error between y_{train} and $y_{predicted}$ (the value it predicts y takes). On completing the learning process (known as training) the model is evaluated by feeding X_{test} and then comparing the values of y_{test} to $y_{predicted}$. This method of learning is termed supervised learning in that both the features and labels are made available to the model beforehand. Several other types of learning exist based on the level of supervision and of relevance to this work is unsupervised learning. In this type of learning the entire corpus of data is the feature set and the feature set is used to generate labels at defined intervals for training. Successful training of such a model allows for evaluations such as forecasting to be done, and predictions can be made based of this forecast. The number of metrics to be monitored suggest deep learning models (models that are powered by artificial neural networks which progressively extract higher level features as more layers are added to the network), because they are well suited for learning underlying representations in data. For our use case we study two types of artificial neural networks namely Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM) Networks (Karpathy, 2015; Chollet, 2018; Brunton and Kutz, 2019).

2.2 Transformations out of the time domain

In all instances where time is recorded for measurements, there is often a need to encode the time (be it measured synchronously or asynchronously) in such a way that it can be consumed by the models built for analysis. As an instructive example (Kazemi *et al.*, 2019) develops a model-agnostic vector representation for time (Time2Vec) and

their results indicate that performance is improved for all models employing the Time2Vec transformation when compared to their direct equivalents that do not. The paper also shows that the implemented time transformation removes the need for time alignment in that the learned vector representation becomes absolute time invariant.

2.3 Transformations that linearize the measurement space

As one example treatment, Koopman demonstrated that a non-linear dynamical system can be represented by an infinite-dimensional linear operator acting on a Hilbert space of measurement functions of the state of the system (Koopman, 1931; SCHMID, 2010; Brunton *et al.*, 2016, 2017; Brunton and Kutz, 2019). The resulting *Koopman operator* is linear and consequently its spectral decomposition fully characterizes the behaviour of the non-linear system. However, because there are infinitely many degrees of freedom required to describe the space of all possible measurement functions of the state space, the Koopman operator is infinite dimensional (Mezić, 2005; SCHMID, 2010; Brunton *et al.*, 2016). A lot of research (Mezić, 2005; ROWLEY *et al.*, 2009; SCHMID, 2010; Giannakis, 2015; Williams, Kevrekidis and Rowley, 2015; Brunton *et al.*, 2017; Kaiser, Kutz and Brunton, 2017; Champion, Brunton and Kutz, 2019) is being done on ways to obtain approximations of the Koopman operator, since a wealth of knowledge already exists for linear systems and being able to accurately approximate non-linear systems would imply the ease of approximating future states of the system as well as carrying out optimal control (Brunton *et al.*, 2016; Brunton and Kutz, 2019). (Lusch, Kutz and Brunton, 2018) show that we can use deep learning methods to learn universal embeddings of non-linear dynamic systems allowing for the combination of representation (handled by the deep learning network) as well as interpretability (handled by the approximate linear dynamics) (Lusch, Kutz and Brunton, 2018; Brunton and Kutz, 2019; Gin *et al.*, 2019). (Lange, Brunton and Kutz, 2020) show that for a finite long-range time span we can carry out forecasts with Fourier and Koopman transformations and the results obtained from this forecast show that forecasts done in this way are stable when compared to those done by a Long Short Term Memory (LSTM) Neural Network, and also that the further we forecast (that is, forecasting from a forecast) the worse the LSTM results get implying that the LSTM network becomes a bias frequency estimator.

2.4 Biologically inspired algorithms

According to (Hawkins and Ahmad, 2016; Hawkins *et al.*, 2016), the Hierarchical Temporal Memory (HTM) is a biologically constrained theory of intelligence based on the mammalian brain. Numenta (Numenta, no date) has published several papers showing the capabilities of HTM based algorithms and recently they showed in (Ahmad and Scheinkman, 2019) that sparse representations can be combined with artificial neural network algorithms to yield more robust results when compared to using only the artificial neural networks. In relation to the work presented in this paper, they show that anomaly detection can be carried out in an unsupervised fashion (they show their results for a univariate system) and that the HTM algorithm is able to capture and handle various phenomena such as concept drift, continuous learning, online predictions, data retention to name a few (Ahmad *et al.*, 2017).

2.5 Self-attention

Attention, first introduced for simultaneous encoding and decoding in neural sequence-to-sequence models by (Bahdanau, Cho and Bengio, 2014), was developed to capture the mapping of tokens between two sequences. Contrarily, self-attention applies attention to a single context instead of across multiple sequences and its ability to directly encode long-range dependencies and parallelize has led to state-of-the-art performance for varied tasks such as (Vaswani *et al.*, 2017; Devlin *et al.*, 2018; Parmar *et al.*, 2018; Shaw, Uszkoreit and Vaswani, 2018; Huang *et al.*, 2019; Brown *et al.*, 2020; Dhariwal *et al.*, 2020; Wang *et al.*, 2020). Naturally because attention models can capture long-term dependencies and because of their success in language processing models, attention models are slowly being adopted for time series forecasting and classification with promising results (Godfried, 2019). (Song *et al.*, 2017) leveraged self-attention for multivariate medical time series data in which they employed 1D kernels for each of the patient variables measured, and at the time of publishing their results were state-of-the-art, but they were soon surpassed by TimeNet (Malhotra *et al.*, 2017). This can be attributed to the effectiveness of transfer-learning based pretraining rather than model architecture (Raina *et al.*, 2007; Malhotra *et al.*, 2017; Godfried, 2019). (Li *et al.*, 2019) focused on addressing problems faced when applying attention models to time series data particularly looking at the effects of self-attention in dynamic time series data which have a variance in seasonality and the memory footprint associated with computing self-attention for long data sequences. To remedy these two problems, they introduced a new method to calculate the query and value vectors, and they also proposed a new attention mechanism with improved computational efficiency.

Models based on the LSTM, CNNs and HTM were benchmarked using the Numenta's Anomaly Benchmark Dataset (Numenta, no date; Ahmad *et al.*, 2017; Lin *et al.*, 2020) and all these methods perform exceptionally well for their generic cases.

3. Methodology

For modelling purposes, we use data obtained from a **ATLAS TDAQ Technical Run**. The run effectively carries out the same physics as produced by the detector (that is, the data which is sent through the TDAQ system is previously recorded data) and is used to test how systems will behave during an actual run. The sampling that occurs is a result of a data collection manager fetching physics data at an interval that follows a negative exponential distribution. Numerous events (from the physics data) are discarded very quickly whereas the remaining trailing events have a longer discard lead time, and this manifests as extended network idle times. What is observed on sampling (Figure 4 and Figure 5) the system is a cumulative sum of the event gathering for all the events that follow a certain distribution. This results in data points oscillating randomly about a constant average. The average is expected to be constant because the sampling interval is much larger than the average event processing time (order of milliseconds) (Pozo Astigarraga, 2014; The ATLAS Collaboration, 2019). The patterns observed in Figure 4 and Figure 5 are normal and the noise associated with the measurement of the system are also normal. We observe the same data visualized under two different axes, one in which the time is absolute and one in which it is relative. This is an indicator to how time would influence the results we obtained in each of the two instances. Figure 4 indicates windows of successive measurement and no measurement whereas Figure 5 indicates a single window of continuous measurement.

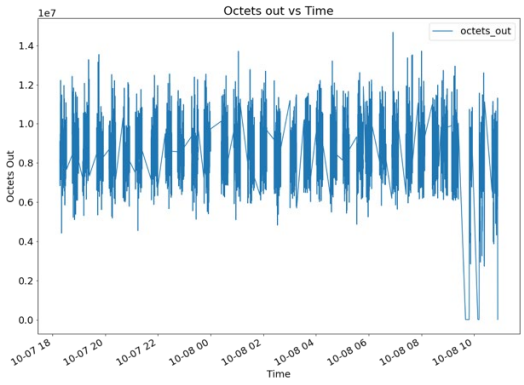

Figure 4: Overview of metric data of a technical run observed with absolute time

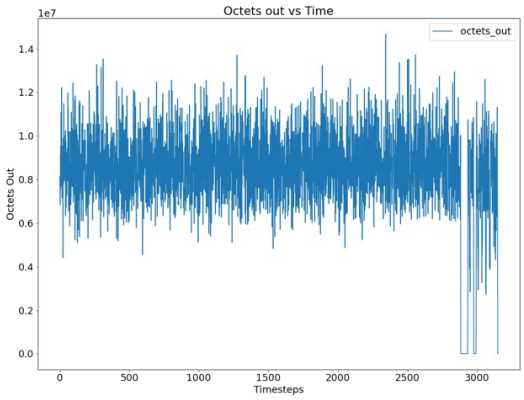

Figure 5: Overview of metric data of a technical run observed with relative time.

The data possesses a hierarchical structure(Figure 7), as such we determined that we need to have the following pattern for resolution:

- A function takes as input the streaming data as pulled from the server at that point in time.
- A pre-processing step to transform the data into the format the is required for the anomaly detection.
- A computing step that calculates the anomaly score for each data point in the dataset.

For model development we used a fraction of the data that came from one of the technical runs (Figure 6), and we limit the scope to this singular technical run so that we are able to estimate the resolution capabilities of each model. The fraction of data used came from the device *'sw data tpu 44'* and port *'1/1/100'* and we term it $\hat{x}$. We develop several models under three different headers namely:

1. Artificial Neural Networks
    - One Dimensional Convolutional Neural Network (1D CNN)
    - Long Short-Term Memory (LSTM)
2. Biologically Inspired Models
    - Hierarchical Temporal Memory (HTM)
3. Spectral Methods
    - Hankel's Alternative view of Koopman (HAVOK)

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 1138565 entries, ('sw-data-tpu-44', '1/1/1', Timestamp('2020-10-05 13:40:38')) to ('sw-data-tpu-44', 'management', Timestamp('2020-10-09 19:31:59'))
Data columns (total 21 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   ID              1138565 non-null  int64
 1   idx             1138565 non-null  int64
 2   speed           1138565 non-null  int64
 3   status          1138565 non-null  int64
 4   load_in         1138565 non-null  float64
 5   load_out        1138565 non-null  float64
 6   octets_in       1138565 non-null  float64
 7   octets_out      1138565 non-null  float64
 8   errors_in       1138565 non-null  float64
 9   errors_out      1138565 non-null  float64
 10  discards_in     1138565 non-null  float64
 11  discards_out    1138565 non-null  float64
 12  packets_in      1138565 non-null  float64
 13  packets_out     1138565 non-null  float64
 14  unicast_in      1138565 non-null  float64
 15  unicast_out     1138565 non-null  float64
 16  multicast_in    1138565 non-null  float64
 17  multicast_out   1138565 non-null  float64
 18  broadcasts_in   1138565 non-null  float64
 19  broadcasts_out  1138565 non-null  float64
 20  NP_discards_in  1138565 non-null  float64
dtypes: float64(17), int64(4)
memory usage: 190.4+ MB
```

Figure 6: Overview of Technical Run dataset description. Typical runs possess approximately 1.2 million entries.

			ID	idx	speed	status	load_in	load_out	octets_in	octets_out	errors_in	errors_out	discards_in	discards_out	packets_in	packets_out	unicast_in	unicast_out	multicast_in	multicast_out	broadcasts_in	broadcasts_out	NP_discards_in
hostname	name	time																					
		2020-10-05 13:40:38	3157669690	1	10000	3	0.000001	0.000037	8.300000	468.133000	0.000000	0.000000	0.000000	0.000000	0.033333	6.100000	0.000000	0.000000	0.033333	4.966670	0.000000	1.133330	0.000000
		2020-10-05 13:41:08	3157685894	1	10000	3	0.000001	0.000034	8.300000	423.867000	0.000000	0.000000	0.000000	0.000000	0.033333	5.466670	0.000000	0.000000	0.033333	4.833330	0.000000	0.633333	0.000000
sw-data-tpu-44	1/1/1	2020-10-05 13:41:38	3157702132	1	10000	3	0.000001	0.000035	8.300000	439.133000	0.000000	0.000000	0.000000	0.000000	0.033333	5.633330	0.000000	0.000000	0.033333	5.033330	0.000000	0.600000	0.000000
		2020-10-05 13:42:08	3157718524	1	10000	3	0.000001	0.000032	8.300000	400.467000	0.000000	0.000000	0.000000	0.000000	0.033333	5.100000	0.000000	0.000000	0.033333	4.766670	0.000000	0.333333	0.000000
		2020-10-05 13:42:38	3157734598	1	10000	3	0.000002	0.000037	23.233300	460.400000	0.000000	0.000000	0.000000	0.000000	0.266667	6.000000	0.133333	0.466667	0.033333	4.866670	0.100000	0.666667	0.000000

Figure 7: Hierarchical structure of the dataset. We observe the data after it has been manipulated to take the form of device $\mapsto$ port $\mapsto$ time.

4. Results

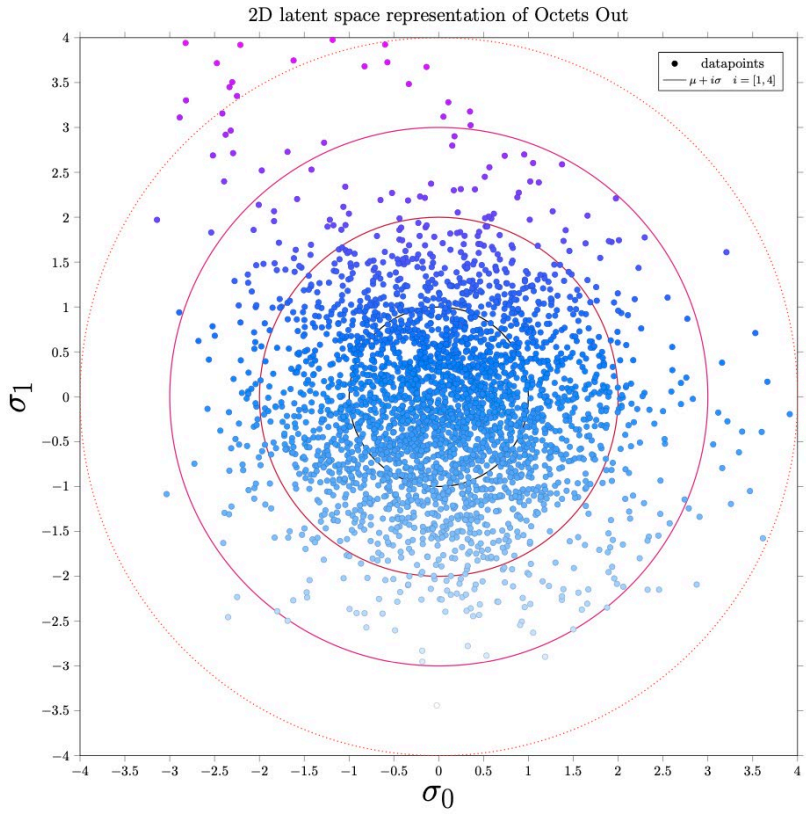

Figure 8**:** Decomposing the input signal to study the underlying patterns in the data. Transforming the signal into a 2D representation we observe that the data can be accurately represented with a Gaussian distribution. This is seen in that the data points follow a multi1variate normal distribution with a mean of 0 and a standard deviation of 1. In applying a threshold, the points lying outside the regions bounded by the selected value of the number of standard deviations automatically become anomalies and the points inside this region automatically become regarded as normal.

4.1 LSTM

Several LSTM networks with varying time windows of 60, 120, 240 and 300 timesteps on it $\hat{x}$ were tested and the results yielded a straight line through the dataset indicating that these LSTM models were able to learn the underlying mean of the data. They also learnt that this data is stationary about the mean with variations and that these variations cannot be modelled using a LSTM network as the changes occurring in the data do not follow a pattern that can be learnt by studying the past events of the time series. The results are indicated in Figure 9.

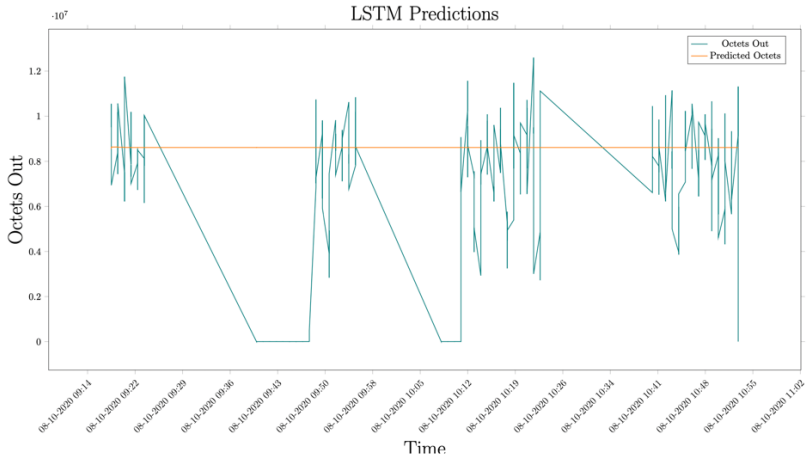

Figure 9: The output of the LSTM network shows the predicted output as an average through the test dataset capturing data stationarity with variance about the mean of the data.

4.2 1D CNN

Several 1D CNN models with windows matching those of the LSTM above for training with data from it $\hat{x}$ were trained and the CNN models also register results similar to those of the LSTM. CNNs are designed to share weights and while this is different from the LSTMs which selectively remember sequences that are deemed important to make predictions, they also draw the same result indicating that the input data is stationary and that the deviations about the mean cannot be learnt in a way that the model does not overfit to the training data. The results of the model are indicated in Figure 10.

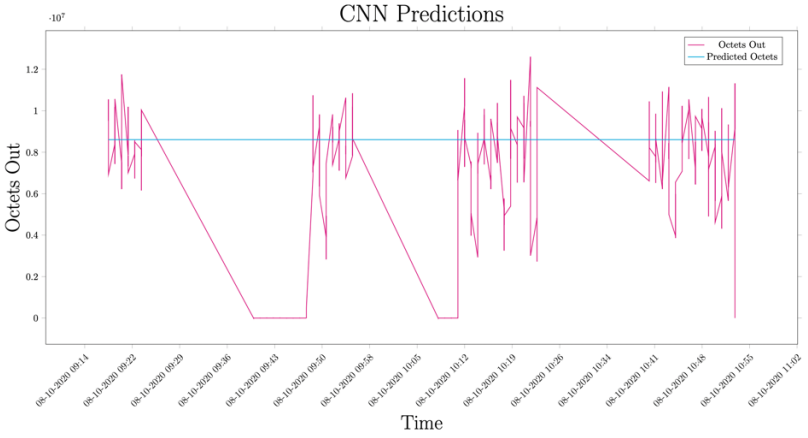

Figure 10: The output of the 1D CNN shows the predicted output as an average through the test dataset also capturing data stationarity with variance about the mean of the data.

4.3 HAVOK

Figure 11 presents the results of decomposing the input signal into its Singular Value components. For a fully developed system we expect to observe pure sinusoidal waves for the modes of V. In our case we observe partial development of the modes of V and this partial decomposition is in line with expectations.

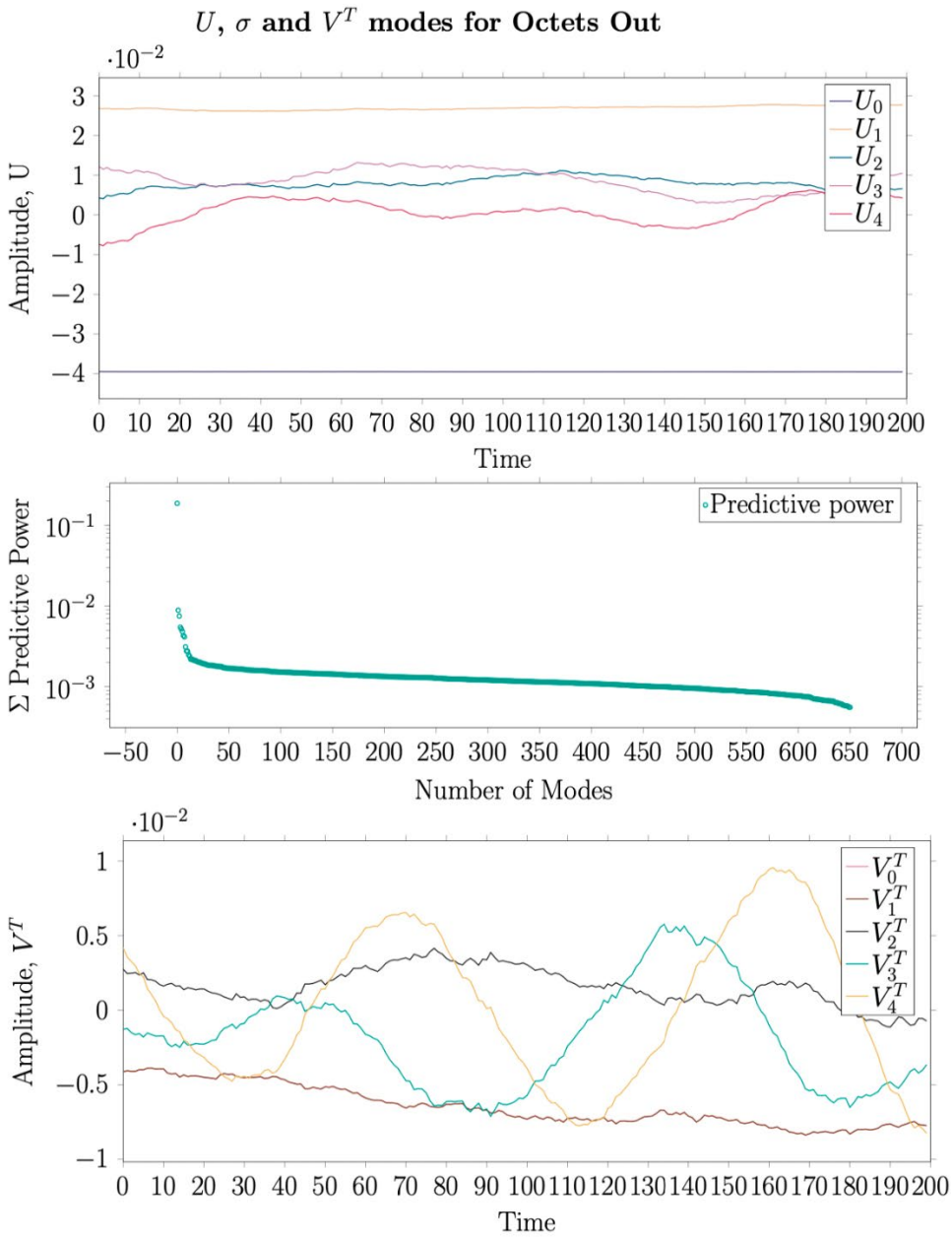

Figure 1: Singular Value Decomposition of Octets Out time delay co-ordinates

Figure 11: Overview of the Hankel Matrix decomposition for Koopman. The modes of U and V are not fully developed but indicate what we expect to observe when a time delay system is decomposed.

4.4 HTM

Figure 12 indicates the results from the HTM model. The HTM model learns on the fly and as such it does not need multiple datasets. The results from this algorithm indicate the presence of anomalies at various points along the data timesteps.

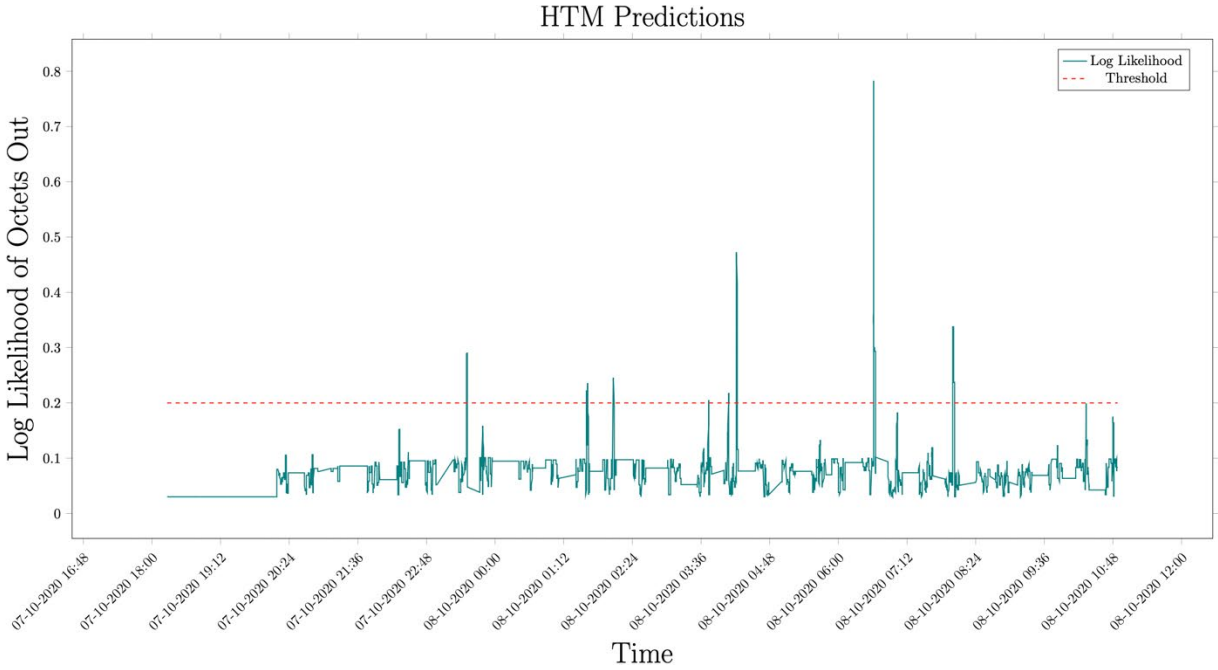

Figure 12: Overview of the HTM algorithm. Several points exceed the anomaly likelihood score of 0.5 indicating the presence of anomalies at those points.

5. Discussion

Figure 9 and Figure 10 illustrate the results from training a 128-unit LSTM model and a 128-unit 1D CNN model respectively. The results show that both models are able to capture the stationarity of the data about the mean and that there is no seasonality. Both models yield the average as a result because the input data does not possess learnable patterns beyond the average that can successfully be learnt to predict the future positions of the input Figure **8** shows the latent space of transforming the input data into a two-dimensional space. No correlation exists between the data points therefore reinforcing the absence of a learnable pattern even when the data is transformed to higher dimensions. While both the LSTM model and the CNN model are able to learn the mean and stationarity of the data, for the purposes of anomaly detection by first forecasting into the future and then classifying the models do not yield the required pattern.

Figure 11 illustrates the result of decomposing the dataset (Singular Value Decomposition of a Hankel Matrix of time delay co-ordinates) into the U, S and V components. The modes of V (top image) are not fully sinusoidal and plotting the first 10 modes of V (truncated to the first 200 time-delay co-ordinates for visibility) shows that the solution is underdeveloped implying the need for more data points in the dataset. However, though not fully developed the results are still consistent with what is reported in the literature. For any system, if a large number (typical orders of 105) of diffeomorphic time delay co-ordinates exist, the system can be decomposed into its U, S and V matrices. The modes of U, S and V are such that the earlier modes are the most identical to the system that was measured (therefore also the most important) allowing for later modes of these matrices to be truncated yielding a linear approximation of the system with close to round off machine accuracy. Because anomaly detection has normal and anomalous components, as shown in (Brunton and Kutz, 2019), it is possible to visualize when anomalies will occur based on the intermittent forcing displayed by the $r'th$ mode of the decomposition where r is the number of modes required to accurately approximate the system. For this use case we note that finding a Koopman invariant subspace for the system being investigated would yield a very robust anomaly detection system.

Figure 12 illustrates the log-likelihood scores for each of the data points in the dataset. The algorithm is tuned such that when there is an anomaly in the system the log likelihood outputs a value higher than the normally expected log-likelihood Fscore (greater than 0.3). Four anomaly regions are apparent in the data stream which tells us that the algorithm notes four different instances for which the system behaves in a way that was not expected. By design, HTM

based architectures are able to benefit from the time horizon for each of the data points and as such can learn underlying patterns which cannot easily be observed even by a human expert. In the literature [10], it is observed that HTM models continuously learn over time and that a presence of false positives early on levels out as more data is seen by the model. For the use being investigated HTM models meet the requirements because they are able to learn sparse data representations as well as being able to recall the patterns they have learnt for the data.

In essence anomaly detection is the comparison of a forecasted future value vs the true value for that time step. The predictive power and robustness of the model influences the forecasted value and as such if the model is a poor predictor of a future value there is high likelihood that the model will cause the triggering of false positives and false negatives resulting in the system alerting for anomalies when there are no anomalies or not alerting for anomalies when there are actual anomalies. The models presented above attempt this anomaly detection task in different ways and from each of them we draw some deductions. From the artificial neural network models (LSTM and 1D CNN) we gather that increased model complexity does not offer you any better solutions if the system cannot be accurately represented. From the spectral methods (Koopman) we gather that the prior is very important for the location of Koopman invariant subspaces. From the biologically inspired models (HTM) we gather that sparsely drawing data from all measurement horizons assists in producing more robust models.

6. Conclusions

The aims of this paper were to present our findings on the development of anomaly detection algorithms for the ATLAS TDAQ high throughput network. The number of dimensions requiring resolution implied that an approach capable of tackling a multivariate input be chosen. In addition, the data possessed an asynchronous time variable intended to aid the solution in better understanding the behaviour of a particular node on the basis of when a device had been polled and to also allow for seasonal patterns to be drawn. The volume of data requiring processing per second implied that methods capable of handling streaming data be considered. Based on all the above factors, data-driven techniques were employed for the solution. In testing these methods only one metric was retained for simplicity (the solutions scale across metrics). We observed that neural network architectures though capable of learning underlying representations of data could not find learnable patterns in the data beyond learning the average of the dataset. Biologically inspired models yielded the best results of the developed methods due to that they were able to not only capture the seasonal trends of the data, but also predict closely what future values would be. Linearised measurement space models yielded partial results due to requiring increased orders of data points to fully define the underlying patterns. Similar to neural network-based architectures, both models assumed that the data recorded fully modelled the system's behaviour and based on the linearised measurement space results we observe why the neural architectures were not successful at finding the approximate path for future values of the measurements.

The models presented above are all class agnostic and as such can be applied to different domains once the data has been pre-processed to follow the formats required by each of the algorithms. One useful application would be in the energy sector where power plants are always expected to operate at optimal efficiency. Two avenues exist whereby there is a real-time anomaly detection system or the case whereby failure has occurred and there is a need to analyse the data from the failure. Robust anomaly detection algorithms would allow for early detection of system failures and also offer an avenue for root cause analysis in the cases when failure has already occurred.

We conclude by noting that the algorithms developed in this paper are generic and thus are applicable to different domains.

References

Aad, G. *et al.* (2008) 'The ATLAS Experiment at the CERN Large Hadron Collider', *JINST*, 3, p. S08003. 437 p. doi: 10.1088/1748-0221/3/08/S08003.
Ahmad, S. *et al.* (2017) 'Unsupervised real-time anomaly detection for streaming data', *Neurocomputing*, 262, pp. 134–147. doi: https://doi.org/10.1016/j.neucom.2017.04.070.
Ahmad, S. and Scheinkman, L. (2019) 'How Can We Be So Dense? The Benefits of Using Highly Sparse Representations'.
Bahdanau, D., Cho, K. and Bengio, Y. (2014) 'Neural Machine Translation by Jointly Learning to Align and Translate'.
Brown, T. B. *et al.* (2020) 'Language Models are Few-Shot Learners'.

Brunton, S. L. *et al.* (2016) 'Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control', *PLOS ONE*. Edited by H. Ae. Kestler, 11(2), p. e0150171. doi: 10.1371/journal.pone.0150171.
Brunton, S. L. *et al.* (2017) 'Chaos as an intermittently forced linear system', *Nature Communications*, 8(1). doi: 10.1038/s41467-017-00030-8.
Brunton, S. L. and Kutz, J. N. (2019) *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press. doi: 10.1017/9781108380690.
Chalapathy, R., Toth, E. and Chawla, S. (2018) 'Group Anomaly Detection using Deep Generative Models'.
Champion, K. P., Brunton, S. L. and Kutz, J. N. (2019) 'Discovery of Nonlinear Multiscale Systems: Sampling Strategies and Embeddings', *SIAM Journal on Applied Dynamical Systems*, 18(1), pp. 312–333. doi: 10.1137/18m1188227.
Chollet, F. (2018) *Deep learning with Python*. Shelter Island, NY: Manning Publications Co.
Collaboration, A. (2017) *Technical Design Report for the Phase-II Upgrade of theATLAS TDAQ System*. Geneva. Available at: https://cds.cern.ch/record/2285584.
Cui, Y., Ahmad, S. and Hawkins, J. (2016) 'Continuous Online Sequence Learning with an Unsupervised Neural Network Model', *Neural Computation*, 28(11), pp. 2474–2504. doi: 10.1162/NECO_a_00893.
Devlin, J. *et al.* (2018) 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'.
Dhariwal, P. *et al.* (2020) 'Jukebox: A Generative Model for Music'.
Giannakis, D. (2015) 'Data-driven spectral decomposition and forecasting of ergodic dynamical systems'.
Gin, C. *et al.* (2019) 'Deep Learning Models for Global Coordinate Transformations that Linearize PDEs'.
Godfried, I. (2019) 'Attention for time series classification and forecasting', *Medium*. Towards Data Science. Available at: https://towardsdatascience.com/attention-for-time-series-classification-and-forecasting-261723e0006d.
Gong, D. *et al.* (2019) 'Memorizing Normality to Detect Anomaly: Memory-augmented Deep Autoencoder for Unsupervised Anomaly Detection', in *IEEE International Conference on Computer Vision (ICCV)*.
Hawkins, J. *et al.* (2016) *Biological and Machine Intelligence (BAMI)*. Available at: https://numenta.com/resources/biological-and-machine-intelligence/.
Hawkins, J. and Ahmad, S. (2016) 'Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex', *Frontiers in Neural Circuits*, 10, p. 23. doi: 10.3389/fncir.2016.00023.
Huang, C.-Z. A. *et al.* (2019) 'Music Transformer: Generating Music with Long-Term Structure', in. Available at: https://arxiv.org/abs/1809.04281.
Kaiser, E., Kutz, J. N. and Brunton, S. L. (2017) 'Data-driven discovery of Koopman eigenfunctions for control'.
Karpathy, A. (2015) 'The Unreasonable Effectiveness of Recurrent Neural Networks'.
Kazemi, S. M. *et al.* (2019) 'Time2Vec: Learning a Vector Representation of Time'.
Koopman, B. O. (1931) 'Hamiltonian Systems and Transformation in Hilbert Space', *Proceedings of the National Academy of Sciences*, 17(5), pp. 315–318. doi: 10.1073/pnas.17.5.315.
Lange, H., Brunton, S. L. and Kutz, N. (2020) 'From Fourier to Koopman: Spectral Methods for Long-term Time Series Prediction'.
Leahu, L. (2013) *Analysis and predictive modeling of the performance ofthe ATLAS TDAQ network*. Available at: https://cds.cern.ch/record/1504817.
Li, S. *et al.* (2019) 'Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting'.
Lin, S. *et al.* (2020) 'Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model', in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
Lusch, B., Kutz, J. N. and Brunton, S. L. (2018) 'Deep learning for universal linear embeddings of nonlinear dynamics', *Nature Communications*, 9(1). doi: 10.1038/s41467-018-07210-0.
Malhotra, P. *et al.* (2017) 'TimeNet: Pre-trained deep recurrent neural network for time series classification'.
Mezić, I. (2005) 'Spectral Properties of Dynamical Systems, Model Reduction and Decompositions', *Nonlinear Dynamics*, 41(1–3), pp. 309–325. doi: 10.1007/s11071-005-2824-x.
Numenta (no date) 'Hierarchical Temporal Memory (HTM)', *Numenta.org • Hierarchical Temporal Memory (HTM)*. Available at: https://numenta.org/hierarchical-temporal-memory/.
Parmar, N. *et al.* (2018) 'Image Transformer'.
Pozo Astigarraga, M. E. (2014) 'Evolution of the ATLAS Trigger and Data Acquisition System'. Available at: http://cds.cern.ch/record/1751941.
Pozo Astigarraga, M. E. (2017) *THE ATLAS DATA ACQUISITION SYSTEM IN LHC RUN 2*. Geneva. Available at: https://cds.cern.ch/record/2292434.
Pozo Astigarraga, M. E. (2018) 'ATLAS Trigger and Data Acquisition Upgrades for the HighLuminosity LHC'. Available at: https://cds.cern.ch/record/2645546.

Raina, R. *et al.* (2007) 'Self-Taught Learning: Transfer Learning from Unlabeled Data', in *Proceedings of the 24th International Conference on Machine Learning*. New York, NY, USA: Association for Computing Machinery (ICML '07), pp. 759–766. doi: 10.1145/1273496.1273592.
ROWLEY, C. W. *et al.* (2009) 'Spectral analysis of nonlinear flows', *Journal of Fluid Mechanics*, 641, pp. 115–127. doi: 10.1017/S0022112009992059.
SCHMID, P. J. (2010) 'Dynamic mode decomposition of numerical and experimental data', *Journal of Fluid Mechanics*, 656, pp. 5–28. doi: 10.1017/S0022112010001217.
Shaw, P., Uszkoreit, J. and Vaswani, A. (2018) 'Self-Attention with Relative Position Representations'.
Shih, S.-Y., Sun, F.-K. and Lee, H. (2018) 'Temporal Pattern Attention for Multivariate Time Series Forecasting'.
Song, H. *et al.* (2017) 'Attend and Diagnose: Clinical Time Series Analysis using Attention Models'.
The ATLAS Collaboration (2019) *ATLAS: a 25-year insider story of the LHC experiment*. Singapore: World Scientific (Advanced series on directions in high energy physics). doi: 10.1142/11030.
Vandelli, W. and Collaboration, A. (2019) *ATLAS Trigger and Data Acquisition Upgrades for the HighLuminosity LHC*. Geneva. Available at: https://cds.cern.ch/record/2688793.
Vaswani, A. *et al.* (2017) 'Attention Is All You Need'.
Wang, H. *et al.* (2020) 'Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation'.
Wielgosz, M. *et al.* (2018) 'The model of an anomaly detector for HiLumi LHC magnets based on Recurrent Neural Networks and adaptive quantization', *Engineering Applications of Artificial Intelligence*, 74, pp. 166–185. doi: https://doi.org/10.1016/j.engappai.2018.06.012.
Williams, M. O., Kevrekidis, I. G. and Rowley, C. W. (2015) 'A Data–Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition', *Journal of Nonlinear Science*, 25(6), pp. 1307–1346. doi: 10.1007/s00332-015-9258-5.
Wyszynski, O. J. and Pozo Astigarraga, M. E. (2019) 'Machine Learning Techniques in the ATLAS TDAQ NetworkMonitoring System'. Available at: https://cds.cern.ch/record/2667381.

Biographies

Mitchell Phiri is a Master's candidate in the Department of Mechanical Engineering Science at the University of Johannesburg. He obtained his BEng in Mechanical Engineering from the University of Johannesburg in 2018. His current research interests lie in advancing current methods employed in Engineering systems via Artificially Intelligent methods.

Simon Connell is a Professor of Physics at the University of Johannesburg within the Faculty of Engineering and the Built Environment in the Department of Mechanical Engineering Science. Research interests in Particle Physics, Nuclear Physics, Nuclear Energy, Materials Science, Quantum Physics, High Performance Computing and Applied (innovation) Physics. Rating by the SA Research Funding Agency (NRF) acknowledges "considerable international recognition". He is a past president of the South African Institute of Physics. He is the founding member of the South African participation in High Energy Physics at the ATLAS Experiment at CERN.

Pathmanathan Naidoo is a Professor of Research in the Faculty of Engineering and the Built Environment, University of Johannesburg. He is a Fellow of the South African Academy of Engineers, a Fellow of the South African Institute of Electrical Engineers, a senior member of IEEE and a member of IET and Cigre. He is a registered professional engineer and a specialist consultant in electrical energy and power systems. His current research interests are in Sustainable Development as driven by the Green Economy and Industrial Revolution 4.0. Dr. Naidoo's four decade industrial career was with the Electricity Supply Commission of South Africa; from Engineer in Training to Non-Executive Director.

Dave Nicholls is the Chair of the SAIEE Nuclear Chapter. He started work as a nuclear engineer officer in the Royal Navy submarine service before joining Eskom where he worked for 35 years before retiring in 2018 as the Chief Nuclear Officer. He is currently the Chairman of the South African Nuclear Engineering Corporation (NECSA). Mr. Nicholls was the Chairman of the IAEA Technical Working Group on Light Water Reactors from 2010 to 2016 and is currently the Co-Chair of the IAEA Technical Working Group on Nuclear Power Plant Operations. He was a member of WANO's Post-Fukushima Design Review Team from 2012 to 2018.