Advanced Multi-Variate Time Series Analytic Techniques (ATTENDS)

Richard Lau[1*], Anindo Bagchi[1], John Shen[1], Tony Triolo[1],
Kenneth Sanchez[2], Lihan Yao[1], Jacob Kovarskiy[1], Roberto Castro[1]
Peraton Labs[1], Test Resources Management Center[2]
331 Newman Springs Road, Red Bank New Jersey, 07701[1]
clau@peratonlabs.com[1*]

ABSTRACT

We describe an advanced architecture supporting fast decisions by using multi-variate time series analytic techniques on voluminous datasets that were previously inaccessible. The system, Advanced Multi-Variate Time Series Analytic Techniques (ATTENDS) automates data ingestion, knowledge extraction, and Artificial Intelligence/Machine Learning (AI/ML) algorithm configuration for anomaly detection, failure prediction, causal analysis, and diagnosis. To enable reusability, ATTENDS presents a set of Application Programming Interfaces (API) to support user configurability and remote invocation. The APIs implement state-of-the art AI/ML algorithms for predictive maintenance, sensor component correlation for problem diagnosis, and unsupervised learning of sensor measurement anomaly for support of automated testing and evaluation. We will present two use cases including prediction of Remaining Useful Life (RUL) of Turbofan [1] and sensor diagnosis and recommendation for maintenance actions, as well as detection and quantification of target location error in an airborne platform.

1. Introduction

The Test Resources Management Center (TRMC) has determined that advanced analytical processing of very large diverse data sets using high-performance computing are requirements for successful operational testing of future Department of Defense (DoD) Warfighter systems. One of the key challenges associated with test and evaluation (T&E) is that the data sets are large, diverse, and are received at a high rate, making traditional human-in-the-loop analysis and processing error-prone, time consuming and impractical. The goal of the ATTENDS (Advanced Multi-Variate Time Series Analytic Techniques) system is to enable test analysts to make better/faster decisions. This is accomplished by exploitation of previously inaccessible or unusable data, thereby generating new information and gaining new insights to support evaluation tasks. Instead of analyzing individual small chunks of data, the ATTENDS tools will give the analyst a broad view across the system data and provide advanced AI/ML methods for time series analysis, thus allowing the discovery of “unknown unknowns” and to uncover hitherto hidden problems. Specifically, this paper will investigate AI/ML models and explain how different model are suitable for certain test applications. We will illustrate the benefits of ATTENDS by considering two use cases: 1) Prediction of Remaining Useful Life (RUL) in Turbofans [1], and 2) Detection and analysis of Target Location Errors (TLE) in airborne platforms. In both cases, we will also describe how to perform deeper analysis to investigate the correlation of failures/anomalies with sensor measurements, thus providing guidance to further maintenance and repair tasks.

2. What are the Challenges

Building the next generation of operations support infrastructure that support high degree of automation, data driven feature learning, and have predictive capabilities would require 3 communities of stakeholders to come together: user community, data scientists, and software architects. Specifically, they will need to address several challenges. The following list identifies the main challenges and how they are addressed by ATTENDS:

1. ***Processing large volume of data***. The data source is multi-modal and multi-variate time series data collected from flight recorders during execution of test and evaluation missions. Due to the large volumes of data generated by test events, manual performance analysis results in a significant amount of unmined data. The ATTENDS architecture will use a *workflow engine* to automate and orchestrate the data processing flows and to coordinate among data ingest, storage, training, execution of AI/ML algorithm, and presentation of results to user.
2. ***Automation, minimal human intervention***. Since the content and context of the data vary depending on applications, ATTENDS uses a Knowledge Base Data Management (KBDM) system to extract information from the meta data so that minimum human intervention is needed.
3. ***Predictive maintenance***. Modern solution commonly deploys top-notch embedded sensor devices to monitor state of hardware to strive for faultless functionality. However, most traditional operations still rely on reactive maintenance or scheduled maintenance. On the other hand, predictive maintenance aims to uncover potential problems before they occur and thereby significantly cut down equipment failure rate and downtime. AI/ML will be instrumental in achieving goal of predictive maintenance. This is also a major goal of ATTENDS.
4. ***Unsupervised learning*** for anomaly detection. This is an important topic for many test applications as ground truth data is not always available. In AI/ML, there are many techniques to deal with unsupervised learning such as Principal Component Analysis (PCA), K-means clustering or K nearest neighbors (KNN), which are considered by ATTENDES. In this paper we describe how ATTENDS uses AutoEncoder to detect anomalies of multivariate time series due to its capability to discover complex patterns.
5. ***Deep dive Diagnosis*** In the context of predictive maintenance and planning, it is highly desirable if there are associated diagnosis capability that comes with that the tool. Problem diagnosis is historical one of the hardest problems and often requires experienced maintenance personnel which are scarce and of high demand. ATTENDS exploits AI/ML to extract structure and correlation of the input multivariate data. The result is used to guide human operator to focus on the important parts of a data set instead of the entire data set.

To address these challenges, the ATTENDS system is built with the following building blocks: Fast data ingest, knowledge management, workflow for automation, AI/ML algorithm training, and execution, API for support of reusability, User Interface for support of maintenance, test and evaluation. The following sections describe how these building blocks are implemented in ATTENDS and how they are orchestrated in two use cases.

3. ATTENDS Approach

3.1 Architecture

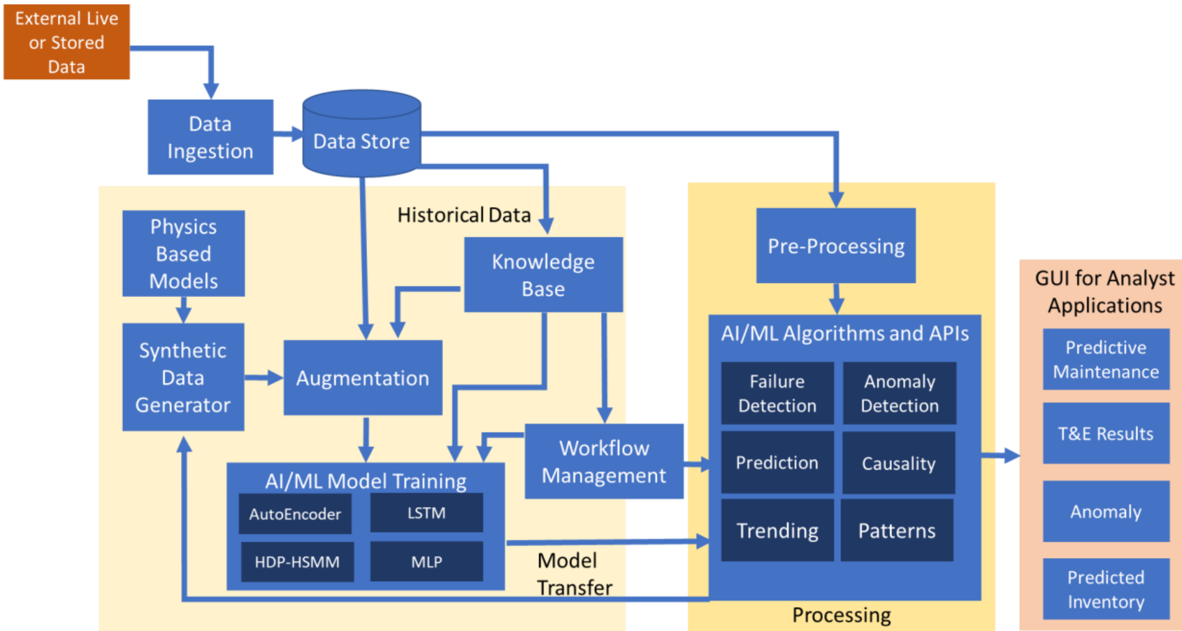

Figure 1 ATTENDS Architecture

ATTENDS supports two principal modes of operations: (1) *batch* mode to enable fully automated operations with minimal human intervention, as well as (2) *query* mode to allow users capabilities for deep dive diagnosis. To allow the system to process large volumes of data, the ATTENDS architecture provides a streamlined Workflow Manager, which schedules rapid Data Ingestion from an external source, Pre-processing of data, the training of AI/ML models and the use of the trained model by invoking the AI/ML API. The system supports workflows in *batch* mode, which are fully automated and triggered when a file becomes available from some external source and is ingested to the Data Store, and workflows in *query* mode, which are invoked by a user through the GUI for deep dive diagnosis.

A key element of the architecture includes a Knowledge Base Data Management (KBDM) system that provides capabilities for Knowledge Management, implemented in the form of Directed Graphs. The Knowledge Base includes domain knowledge about the AI/ML algorithms, the datasets and the analysis supported. The Workflow Manager, when needed, will retrieve knowledge from the Knowledge Base for information about the AI/ML algorithms to be used, or how a dataset needs to be pre-processed, as well as information from meta data to support training the underlying Hierarchical Dirichlet Process Hidden Semi-Markov Model (HDP-HSMM) or neural network models, such as Multi-Layer Perceptron (MLP) and Long-Short Time Memory (LSTM) before they can be used for AI/ML applications.

The architecture recognizes the need for Data Augmentation since there is often a lack of data for training models. ATTENDS provides capabilities to generate synthetic data using AI/ML algorithms that enable the generation of synthetic data with the same characteristics as the underlying sparse data. In addition, ATTENDS includes capabilities to use Physics based models, which are also stored in the Knowledge Base, to augment data.

The architecture supports AI/ML Algorithms/API for proactive maintenance by providing capabilities for failure detection and prediction of RUL as well as capabilities for discovering causality which will be used to assist analysts for problem diagnosis. The architecture includes a variety of algorithms including anomaly detection with focus on unsupervised learning capabilities, trending, and the recognition of patterns in a dataset. Each of the AI/ML algorithms is trained and is invoked for use by the Workflows through the AI-ML API.

When an AI/ML algorithm completes the analysis, the results are provided to the User through the Graphical User Interface (GUI). The GUI provides users a rich capability to visualize the results as well as to perform further deep dive diagnosis on completed analysis. The output of the system lets analysts visualize information at different levels of abstraction, and obtain results for T&E in the form of prioritized lists or numerous graphical visualization forms including time series plots, histograms, and heat-map displays, as well as singling out anomalies in the data.

3.2 AI/ML – New Capabilities and Applications

The versatility and popularity of AI/ML algorithms have attracted many industry practitioners to develop intelligent applications in diverse fields including robotics, self-driving automobiles, medical diagnosis, weather forecast, recommender systems, and video/photo labeling, etc. The collection of AI/ML applications in the last decade penetrates such diverse areas that innovation in applications is only limited by the imagination of the researchers and application practitioners.

While there are many new capabilities users can exploit from AI/ML innovations, we will focus on 3 main domains of application in this paper: 1) Predictive capability; 2) Diagnosis; 3) Automated anomaly detection. Given enough historical data on the equipment failure and corresponding sensor measurements, a trained AI/ML algorithm is effective in prediction of failure time. Such capability will be extremely important in predictive maintenance, where machines with high probability of failure are isolated for maintenance and repair effort. To further assist the maintenance effort, sophisticated AI/ML algorithms can be used to perform deep analysis by learning the internal states of the machines via the sensor data. Such information will be valuable for diagnosis and prioritization of maintenance schedule or ordering of repair parts.

In some applications, it is difficult to acquire ground truth failure data. Here is where AI/ML unsupervised learning would be useful. This branch of AI/ML algorithm aims to learn the most important components of the measurement data (e.g. sensor data of a flight mission) and remove other unnecessary components such as noise and anomalies. As the term unsupervised learning suggests, the data does not need to be labelled as anomalous or normal. Such approach exploits the low dimensionality of the sensor data and automatically identifies whether any of the sensor data are anomalous.

In subsequent sections, we will describe each of the 3 techniques and give two example use cases to illustrate how AI/ML is used to address them.

3.3 Re-usability

AI/ML Algorithms are implemented in ATTENDS with a corresponding API that can be used for the invocation of any algorithm. The API for any algorithm specifies the input variables that will be needed for the algorithm, as well as the output from the algorithm when applied for any of the applications. The API also defines appropriate alerts that can be generated, when needed, as part of the output from the algorithm. The use of the API enables re-usability of the algorithms since the API can be exposed over an external interface to enable remote invocation.

Figure 2 shows an example of the API for ATTENDS TLE Analysis. In this example, the API provides access to the Auto-Encoder for TLE analysis. Two levels of users, the Casual User, and the Advanced User are provided. Both users have capabilities for users coming in through the GUI to prepare the data, train the Model, or invoke the algorithm to analyze the TLE dataset, but advanced users can access additional capabilities for configuration and customization, as shown on the right side of Figure 2.

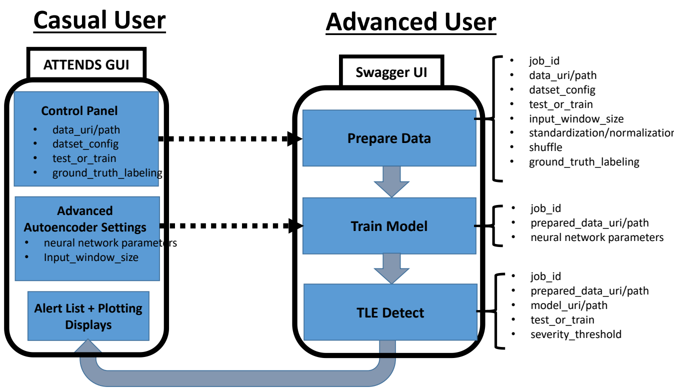

Figure 2 AutoEncoder API

Implementation of the AI/ML API is provided by a server built with a FastAPI Python package. It hides the specificities of the details of the AI/ML algorithm, standardizes the interactions with them, and allows other parts of the system to access the algorithm remotely. It provides uniform access via HTTP/REST protocol, allowing the workflow manager to coordinate the work of various modules, such as data preprocessing, model training, analysis, and storage of results. When a request for a service (such as “build a Neural Network with certain parameters”) is received from a Workflow Manager, the API, stored in an interface server invokes the proper AI/ML application to fulfill that request, and returns the results to the caller. The results are persisted in a common storage area and can be used by other services without the need to transfer a huge amount of data.

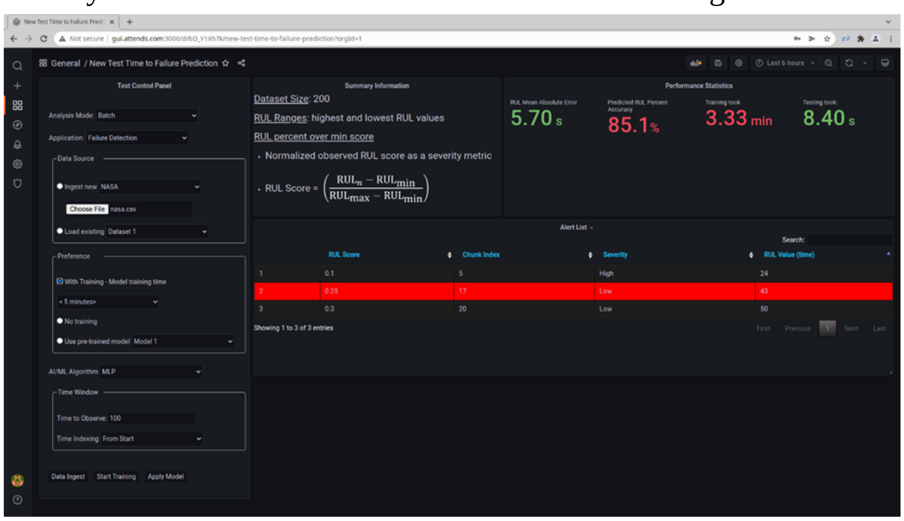

Figure 3 Example GUI screen for RUL Analysis

3.4 User Interface

ATTENDS includes a rich Graphical User Interface that provides a flexible display of results of AI/ML analysis. The same interface can also be used to provide status of the workflow and initiate new tasks. The display is an interactive tool for users to navigate through different levels of the analysis and results, providing results and alerts to users who do not have much of a data science background, while offering advanced capabilities to analysts with a data science background to further explore the results.

Figure 3 provides an example of the GUI screen for the RUL Analysis. It includes a panel on the left that can be used by users to initiate new analysis or to modify the parameters to use for an analysis. The panel on the right provides the key results from RUL prediction using MLP.

The GUI is implemented using the open-source dashboard and visualization tool Grafana. The tool provides an effective way to view results of an AI/ML analysis, to visualize the results, to perform queries and to obtain the results for the analysis. It includes a dashboard structure with multiple panels that can be arranged as a grid for various elements to be viewed in a summary form. The solution includes capabilities for users to customize the dashboard so that it can be set up in an effective way to display results.

4. ATTENDS Use Cases

We describe two use cases which illustrate the benefits of AI/ML techniques in predictive maintenance and unsupervised anomaly detection. The first case utilizes an open-source data set from NASA on maintenance of Turbofan equipment and the second one applies to a simulated dataset related to sensors measurements of airborne platforms during a test event. In the following we describe how ATTENDS addresses these two use cases.

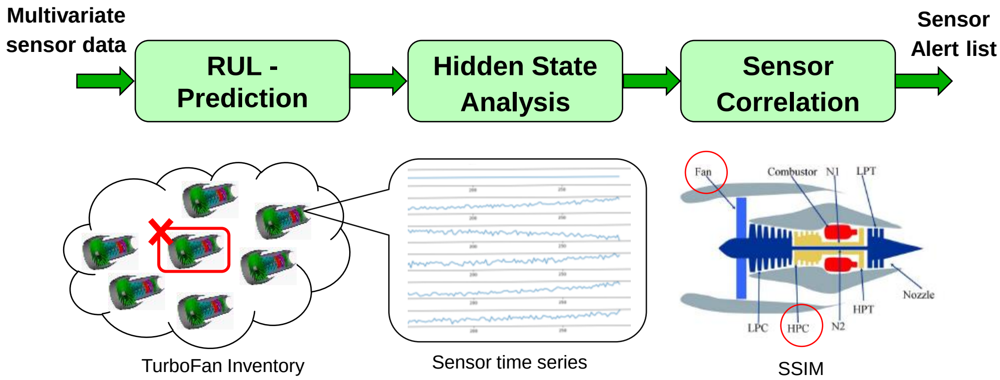

Figure 4 Predictive Maintenance

4.1. Predictive Maintenance

4.1.1. Description of NASA Turbofan problem

In the first case, we will use an example application that is based on simulated engine failure data provided by NASA. Descriptions of the data can be found in [1]. The NASA Prognostics data can be found here [2]. This data consists of sensor measurements from an engine’s lifecycle in the form of 25 sensor values as multivariate time series. Each engine lifecycle time series is characterized by nominal or healthy engine operation at the start until gradual failure at the end of each lifecycle. As the engine approaches failure, deviations and anomalies begin to occur in the data. In traditional reactive maintenance, equipment failure is dealt with after it happens, which not only have high repair cost, but also increase down time of the service. Predictive maintenance, on the other hand, predicts the onset of catastrophic failures, finds the root cause of the problem, and initiates maintenance before equipment fails. ATTENDS’s solution to the predictive maintenance problem is based on AI/ML technologies to predict and diagnose potential root cause and allocate proper resource to fix the problem. Figure 4 shows the overall approach. It involves 3 main AI/ML algorithms, namely Multi-Layer Perceptron for predicting time to failure; Hidden Semi-Markov Model (HSMM) for learning key hidden states of the system, and Sensor-State Interaction Matrix (SSIM) to find problematic sensors. The ATTENDS solution automates data ingestion; feed data for training the AI/ML algorithms; and orchestrates the execution of the algorithms to present results to the maintenance personnel via an interactive GUI. Details of the AI/ML procedure are described in the following.

4.1.2. Prediction of time-to-failure

Multilayer Perceptrons (MLPs) have been widely applied to prediction problems and serve as baseline ML models to compare to state-of-the-art neural network structures. In particular, we apply the MLP to RUL prediction for ATTENDS. This approach takes the set of input sensor values over a local time window to predict the RUL of the turbofan engine. Simple neural network models act as universal function approximators and can learn non-linear relationships for both classification and prediction problems. This property of neural networks makes the MLP a suitable baseline model for RUL prediction.

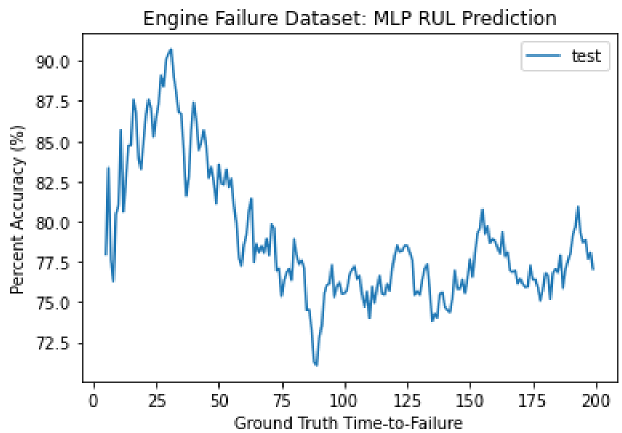

Figure 5 Prediction Accuracy

For the NASA Turbofan Failure Dataset, ground truth RUL values are provided for each engine lifecycle. The NASA dataset splits the data into a 50/50 training and testing split. In the training data, the time series ends at the point of failure while in the testing data, the time series is cut off before failure occurs. Even though test data is cut off before failure occurs, ground truth RUL values are still provided, which is used to train the MLP. After training the MLP model predicts RUL with a percent accuracy greater than 70% with increasing accuracy as the ground truth RUL is smaller (Figure 5).

4.1.3. Diagnosis function after failure prediction

The RUL prediction identifies those Turbofans that may fail soon. From a maintenance perspective, it is helpful if the AI/ML system is able to provide indications to about sensor(s) are more likely correlated to the imminent failure. Failure Detection is a diagnostic functionality in the preventative maintenance application, where asset failures are detected in retrospect. The HDP-HSMM is demonstrated on the NASA Turbofan Failure dataset, where 25 sensor readings from engine lifecycles have terminated in a failure event or remained operational.

The HDP-HSMM belongs to the Hidden Markov family of models (HMM models). Model states of HMM models transition due to time series dynamics exhibited throughout the operational cycle. For the preventative maintenance application, they encode the probability of observing a failure event through time. In Figure 6, 5 sensor readings are plotted. Background colors of the plot correspond to one of three model states. The red plot underneath indicates the model's failure decision.

In Figure 6, sensor readings begin initially in a stable condition, colored as blue (periodic, instantaneous lime-green states). When sensor readings enter a new phase, such as trending together, the model transitions to a new, purple state. Throughout the engine dataset, engine operations may terminate in this phase (purple), or operate into worse conditions where readings

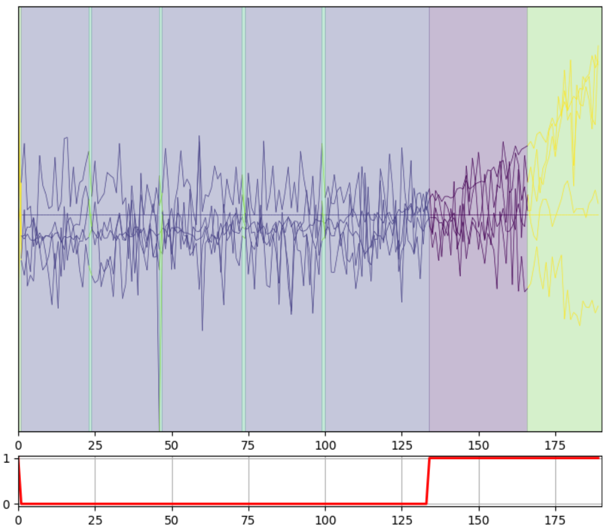

Figure 6 Example HDP-HSMM Output for Engine Lifecycles

diverge (lime-green state). By analyzing failure statistics attributed to each state, an analyst may interpret different states in a sequence that corresponds to worsened engine health.

Sensor-State Interaction Matrix (SSIM) provides correlation relations between sensors of the dataset and model states. Because engine failure states are mapped to model states, the SSIM identifies sensors most relevant to the occurrence of engine failure. An analyst using the preventative maintenance application can thus focus the maintenance effort after analyzing only the relevant sensors.

In the SSIM example, (i, j)-th entry of the matrix (right) denotes the i-th sensor's relevance to the j-th model state (corresponding to an event type). For the column indexed 2 in Figure 7, sensor scores for the impulse event at time 75 to 115 are listed. The analyst observes sensors indexed 13 and 12 have a higher interaction score with the impulse than 16 and 1, which can be visually confirmed as the sensors containing the impulse structure. For the failure event at time 180-210, the column indexed 1 lists sensor interactions. The interaction scores reveal sensor 13 is most relevant to the failure event and suggests to the analyst where to focus the maintenance effort.

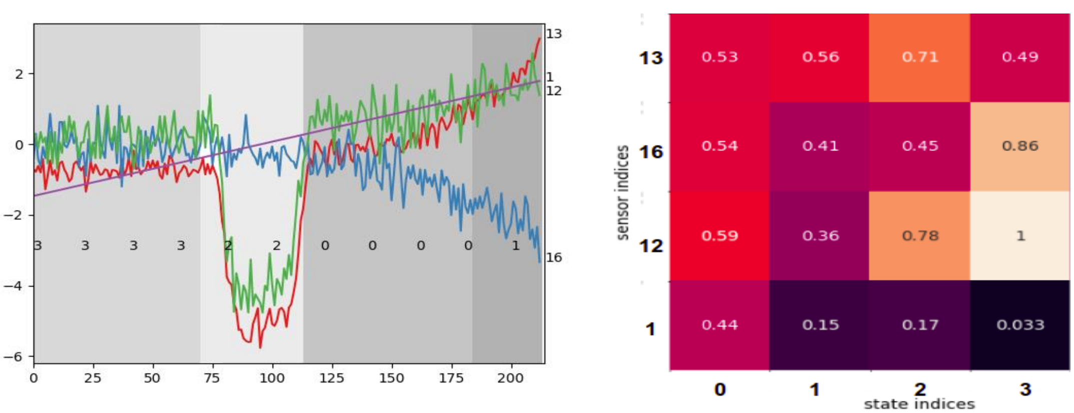

Figure 7 Left: Example sensor readings with the model state annotation displayed horizontally. Right: the SSIM where column indexed 2 correspond to the impulse event, and column indexed 1 correspond to the failure event.

4.2. Anomaly Detection

In the second use case, referred here as the Target Location Error (TLE) use case, we analyze sensor data from multiple airplane platforms which measures locations for targets of interest. The sensor data is expected to be correlated according to certain Physics laws, which are assumed to be unknown for this analysis. Sensor data may contain noise as well as artificially injected errors. Our objective is to detect the anomalies and identify the corresponding erroneous sensors. Although ground truth data has been obtained for checking, we build an AI/ML system capable of identifying erroneous sensors without the need to train with ground truth labels, which are likely to be unavailable in practical operations.

4.2.1. AutoEncoder and Unsupervised learning

A common usage of AutoEncoder is for data compression via dimensional reduction, which is related to the Principal Component Analysis (PCA) concept. In this use case, we propose to use an AutoEncoder neural network to detect anomalies in the input time series, which has unlike a PCA system which is linear, AutoEncoder is capable of modeling non-linear systems.

An anomaly or outlier is one of more data points which is significantly different from the remaining data. Anomaly does not necessarily imply a faulty state but would raise suspicions that it was generated by a different mechanism. In the TLE use case, the error is indeed generated by a different process from that of the normal data, thus fits well with this model. There are a few variations of methods for detection of anomaly. In this paper we describe use of AutoEncoder that belongs to the category of *deviation-based* anomaly detection. The basic principle is that the reconstruction error of the AutoEncoder is used as an anomaly score. As shown Figure 8, the first set is to train an AutoEncoder so that the output best matches the input data.

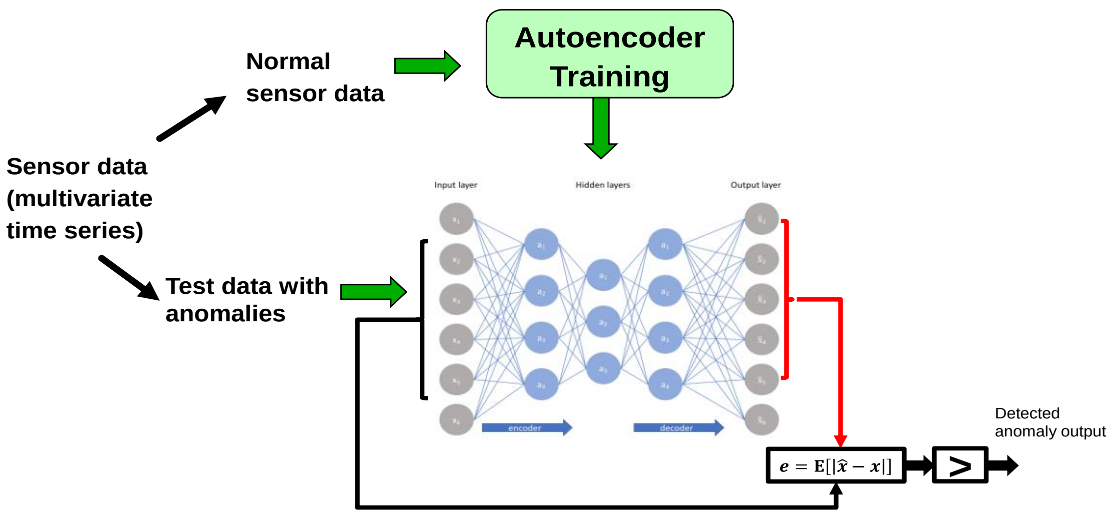

Figure 8 AutoEncoder for Anomaly Detection.

4.2.2. AutoEncoder Algorithm

An AutoEncoder is a neural network consisting of an encoder and a decoder placed in back-to-back configuration. The goal of the AutoEncoder is to compress the input signal into lower dimensional representation ignoring noise or unnecessary information, which is then decoded by the mirrored decoder to generate the original signal as much as possible. As illustrated in Figure 8, the AutoEncoder is first trained with normal data extracted from the input sensor data. The training is *unsupervised* as no extra label (other than the output is the same as the input) is needed. To detect anomaly, the difference between input and output of the AutoEncoder is noted and thresholded to identify outliers.

4.2.2. TLE Detection Result

Figure 9 shows the reconstruction error spread from the position z-axis measurements of the entire dataset. Erroneous data is shown in orange and normal data is shown in blue. The horizontal axis shows the absolute reconstruction error as an MAE while the vertical axis is the number of occurrences of the reconstruction error magnitude. The results show that the AutoEncoder reconstruction error contains a cluster with low reconstruction error while the erroneous data is spread out with comparatively higher reconstruction error values. The threshold selection process places the error detection threshold around 0.01 for the position z-axis sensor. The selected threshold results in a false positive rate (FPR) of 0.027 and false negative rate (FPN) 0.12 shown in Figure 9. These detection characteristics highlight the AutoEncoder's ability to discriminate between erroneous and normal data.

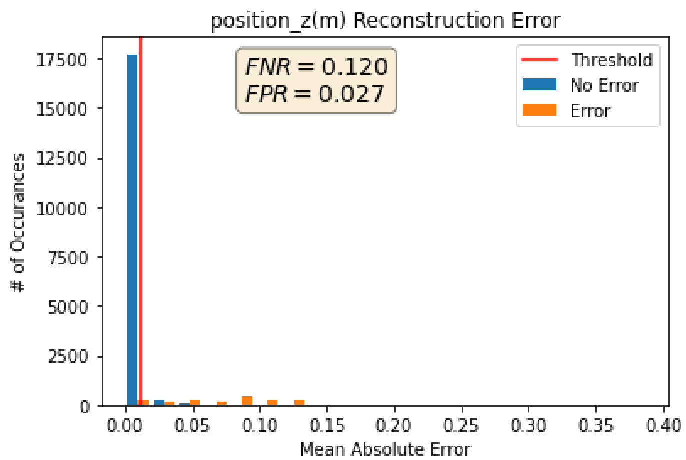

Figure 9 Autoencoder reconstruction error spread for erroneous and normal data

5. Conclusion

This paper describes the Advanced Multi-Variate Time Series Analytic Techniques (ATTENDS) architecture as a solution for many foreseen problems in the areas of predictive maintenance, proactive diagnosis, and automated anomaly detection. The ATTENDS architecture adopts a workflow that facilitates fast ingestion of data for flow-through analytics. The goal is to support fast decisions by using multi-variate time series analytic techniques on voluminous datasets that were previously inaccessible. The system is designed and built in such a way that new AI/ML algorithms can be incorporated with incremental effort. This is achieved by building a set of AI/ML Application Programming Interfaces (API) to support user configurability and remote invocation. This paper also points out why AI/ML is suitable for predictive maintenance and solving diagnosis problems and emphasizes the importance of unsupervised learning in anomaly detection. Two important use cases were used to illustrate the effectiveness of the ATTENDS approach. Future work includes finding new applications and investigation of integration of knowledge information in AI/ML algorithms.

6. Acknowledgement

This paper describes work done by Peraton Labs, which was supported in part by the U.S. Department of Defense, Test Resource Management Center, Test and Evaluation/Science and Technology (T&E/S&T) Program, under their C5 OTA Prime Contract number W15QKN-17-9-5555.

REFERENCES

[1] Saxena et al., "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation" –2008 Intl Conf on Prognostics and Health Management
[2] NASA Turbofan data. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/#turbofan.