A Temporal Anomaly Detection System for Vehicles utilizing Functional Working Groups and Sensor Channels

Subash Neupane*, Ivan A. Fernandez*, Wilson Patterson, Sudip Mittal, Shahram Rahimi Department of Computer Science & Engineering Mississippi State University, Mississippi, USA, (email: {sn922, iaf28, wep104}@msstate.edu, {mittal, rahimi}@cse.msstate.edu)

Abstract-A modern vehicle fitted with sensors, actuators, and Electronic Control Units (ECUs) can be divided into several operational subsystems called Functional Working Groups (FWGs). Examples of these FWGs include the engine system, transmission, fuel system, brakes, etc. Each FWG has associated sensorchannels that gauge vehicular operating conditions. This data rich environment is conducive to the development of Predictive Maintenance (PdM) technologies. Undercutting various PdM technologies is the need for robust anomaly detection models that can identify events or observations which deviate significantly from the majority of the data and do not conform to a well defined notion of normal vehicular operational behavior. In this paper, we introduce the Vehicle Performance, Reliability, and Operations (VePRO) dataset and use it to create a multi-phased approach to anomaly detection. Utilizing Temporal Convolution Networks (TCN), our anomaly detection system can achieve 96% detection accuracy and accurately predicts 91% of true anomalies. The performance of our anomaly detection system improves when sensor channels from multiple FWGs are utilized.

I. INTRODUCTION

Vehicles can be considered a specialized form of Cyber Physical Systems (CPS) equipped with a variety of sensors that generate a volume of operational data. Recent advancements in hardware has made available unprecedented computational power via the use of modern processors and GPUs. Artificial Intelligence / Machine learning technologies running on this hardware, can utilize these large vehicular datasets to build intelligent systems. These systems can help detect abnormal sensor behaviour and identify precursors of various sensor channel faults, monitor and predict the progression of faults, and provide decision-support to develop maintenance schedules. One concept that has risen to prominence both in academia and industry is Predictive Maintenance (PdM) [1]-[4]. However, PdM is complex and expensive to implement [1]. Undercutting the domain of PdM, is the need for robust anomaly detection. AI based anomaly detection systems can identify events or observations that deviate significantly from the majority of the data and do not conform to a well defined notion of normal behavior [5]-[8].

In vehicular systems, several factors are attributed to the deterioration of vehicle components over time. For example,

engine and transmission wear is a direct result of massive heat and vibration produced. This deterioration of components or sub-components can result in abnormal sensor channel behavior signaling impending failure. However, early detection of abnormal patterns and timely scheduling of maintenance events can reduce risk to the underlying processes while increasing lifespan, reliability, and availability, thereby avoiding unplanned downtime and minimizing repair costs [1].

When it comes to a modern vehicle, systems have shifted from being purely mechanical to being comprised of Electronic Control Units (ECU), Sensor, and Actuators. These components work together to ensure normal operation of a vehicle. An ECU waits for a signal from the gas pedal sensor to communicate with a fuel pump, instructing it to deliver more fuel to the internal combustion engine. A modern vehicle is an amalgamation of these individual pieces of machinery working together to complete the desired functionality. We can consider a vehicle to be comprised of various operational subsystems called "Functional Working Groups" (FWGs) [9]. These include the engine system, transmission, fuel system, brakes system, etc. Each FWG is comprised of multiple sensor channels that allow a mechanic to infer the condition of an individual mechanical part and recommend mitigating service. We can say that a vehicle is operating under normal conditions if all individual FWG sensor channel values are gauging normally. Abnormal channel values may be indicative of a future impending failure.

To combat this problem, we present in this paper a temporal anomaly detection system that focuses on operational data of 4 vehicular FWGs and associated sensor channels. Our anomaly detection system is a multi-phased approach utilizing Temporal Convolution Networks (TCN) that learns the underlying relationships between FWG sensor channels. The intuition behind using TCN as a predictor model is that it is suitable for modeling sequential data and the use of flexible receptive fields [10]. In addition, TCN based models are computationally efficient as compared to Long short-term memory (LSTM) methods and possess representational capability to achieve robust and superior prediction performance [11]. We leverage the Vehicle Performance, Reliability, and Operations (VePRO) dataset created by the United States (U.S.) Army Corps of En-

^{*} These two authors contributed equally.

gineers and Mississippi State University (MSU). Our anomaly detection model performs time series forecasting by taking as an input, multi-sensor values for the current time window and outputs a prediction for the subsequent observation. The final decision about an anomaly is made by comparing the prediction to the actual observation.

Major contributions of this paper include:

- We provide a comprehensive description of the Vehicle Performance, Reliability, and Operations (VePRO) dataset. The dataset includes the normal driving data for a fleet of vehicles. We explain and utilize 4 vehicular Functional Working Groups such as engine, transmission, fuel, and brakes, along with the associated sensor channels.
- We develop a TCN model using a sequence-to-sequence learning approach. The model performs time series forecasting by operating on a multi-channel sequence for the current time window and outputs a prediction for the subsequent observation.
- We identify anomalies in FWG sensor channels by comparing the prediction made by the TCN model with the actual observation. We then check robustness of our approach by creating three anomalous test scenarios to investigate the behavior of different sensor channels that affect *FuelRate*.

The rest of the paper is structured as follows: Section II, discusses the model used in the study and provides a brief overview of the current state-of-the-art. Dataset used in the study has been described in Section III. Section IV, gives an overview of our multi-phased architecture. In Section V, we present the evaluation criteria and our results. Finally, Section VI summarizes and concludes the paper.

II. BACKGROUND & RELATED WORK

In this section, we present some background information and some related works on Temporal Convolution Network and Anomaly Detection.

A. Temporal Convolution Network

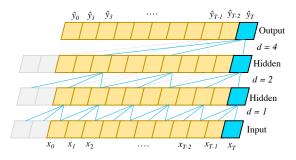


Fig. 1: A causal dilated network with dilated factors d = 1,2, 4 and a kernel size of 3. The first hidden layer has a dilation factor of 2, and the second hidden layer has a dilation factor of 4. The receptive field can accommodate all input sequence values [11].

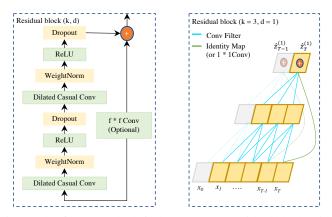


Fig. 2: The figure on the left shows a TCN residual block with a dilation factor of d and a filter of k. When the dimensions of the residual input and output differ, a 1×1 convolution is added. The figure on the right is an example of residual connections in a TCN with d = 1 and k = 3. In this figure, the green lines show identity mappings, while the blue lines show filters in the residual function [11].

TCN [12], is a time series architecture that utilizes onedimensional convolutional layers. It is trained to forecast the next sequence of time series input. Consider a scenario in which we are given a set of time series inputs for a window size t, i.e. $x_0, x_1, x_2,..., x_t$ and we wish to predict some corresponding output $y_{t+1}, y_{t+2}, y_{t+3},..., y_{t+t}$ for the next time window of size t. The key limitation is that when predicting the outputs y values for the next window size t, it can only use the previously observed inputs: $x_0, x_1, x_2,..., x_t$. TCNs adhere to the following two fundamental principles [11]:

- The architecture employs a 1D fully connected convolutional network and transforms any dimensional length of input sequence into a sequence with the same dimensional length [13].
- 2) The convolutions are causal, which means that there is no information leakage from the future to the past [14].

To allow for a long effective history size and a deep network, two important concepts are introduced as part of the TCN architectural elements, which we describe in the following sub sections.

1) Dilated Convolution: When working with time series data, it is often assumed that the model network can memorize long-term information. A simple causal convolution can only examine a history with a size proportional to the network's depth. This makes it difficult to apply causal convolution to sequence tasks, particularly those requiring a longer historical context. To address this issue, dilated convolution, originally proposed by Oored et al. [15] that enables exponentially large receptive fields is employed [16].

For a sequence of input x_0 , x_1 , x_2 ,..., x_t , and k-sized convolution kernel f, the dilated operation F in element s

of the sequence is defined as

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i}$$
(1)

Where d is dilation factor, k is filter size, and $s - d \cdot i$ accounts for the direction of the past. Figure 1 illustrates the dilated-causal convolutions with dilation factors d = 1, 2, and 4.

2) Residual Connections: Figure 1, shows how increasing network depth or extending network width can increase network performance. However, as network depth increases, it becomes more difficult to train the network, and network performance rapidly degrades. To address this issue, residual connections are applied to TCN [17]. This utilizes skip connections throughout the network to accelerate the training process and prevent the vanishing gradient problem. A residual block for baseline TCN is depicted in Figure 2.

B. Anomaly Detection

Liu et al. [19] propose a TCN-Gaussian mixture model (GMM) model for an online anomaly detection time series dataset. First, the time series features are extracted by a dilated TCN. Then their distribution is estimated using a GMM. TCN-GMM measures the similarity between features and clusters using the Mahalanobis distance. The threshold T is compared to the model's measure of similarity to see if the current state is abnormal.

On the other hand, the authors of [10] use a multi-stage TCN (to improve performance) and a multivariate Gaussian distribution to estimate the distribution of prediction errors rather than the features of training data. The anomaly score is then calculated by computing the Mahalanobis distance (MD) between the current prediction error and the pre-estimated error distribution. Three real-world datasets, like electrocardiograms, a dataset of 2-D gestures, and a dataset from the space shuttle, are used to validate the approach.

In another work, using Temporal Convolutional Neural Attention (TCNA) networks, the authors of [20] present a novel anomaly detection framework called TENET for automotive cyber-physical systems. The research introduces a metric called the "divergence score" that evaluates the discrepancy between the expected signal value and the actual signal value. Their method employs a robust quantitative metric and classifier, as well as learned relationships, to detect abnormal patterns.

A comparison of the effectiveness of TCN and LSTM methods in detecting anomalies in time series data has been proposed [21]. Experimental results demonstrated that the TCN method outperformed its LSTM equivalent in terms of recall and F1 score. In addition, it is noted that the TCN model required fewer training iterations. A similar work that evaluates the performance of the TCN approach by comparing its results with the LSTM approach in detecting anomalies of spacecraft telemetry time series data transmitted from the on-orbit satellite is proposed in [22].

Mo et al. [23] present a TCN-based unsupervised anomaly identification approach for time series data that includes seasonality and trend. In contrast to the typical method of setting a global threshold for anomaly detection, the authors developed a novel approach capable of detecting seasonal and trenddriven local anomalies.

To build an anomaly detection system for vehicles, Alizadeh et al. [24] created a hybrid AutoRegressive Integrated Moving Average-Wavelet Neural Networks (ARIMA-WNN) model for predicting the behavior of the operating vehicle and detecting anomalous states based on multiple channel time series data.

In our past work, we have published a Hidden Markov Model (HMM) based survey on machinery fault prediction [25] and have also built a number of anomaly detection systems for vehicles, smart homes, cyber-physical systems, etc. These systems utilize both non-neural network methods and various deep learning approaches [26]–[30]. For example, in [26], we presented an alert system based on HMM, a stochastic model that follows the Markov principle, to detect anomalous states of real data acquired from an operating vehicle. This research looked at the speed, load, engine coolant temperature, and other physical sensor values of vehicles made by different automotive manufacturers.

Kang et al. [31] also proposed a method to detect anomalies in the brake operating unit (BOU) of subway trains using a type of LSTM auto-encoder. The authors of [32] propose a pattern recognition technique for detecting anomalies in transit bus braking systems. This technique employs many sensor data points, such as vehicle speed, transmission gear ratios, acceleration pedal position, and the brake switch, to detect irregularities in brake systems. Bussey et al. [33] present a case study on the detection of anomalies involving low oil pressure in commercial vehicle data. The authors studied a number of data-driven models that were based on heuristics to describe the relationship between three main engine channels: oil pressure, oil temperature, and engine speed.

III. DATASET DESCRIPTION

The Vehicle Performance, Reliability, and Operations (Ve-PRO) program is a collaboration between the United States (U.S.) Army Corps of Engineers and Mississippi State University (MSU). The VePRO dataset contains operational data from a fleet of vehicles, including time series sensor measurements, detected faults, and maintenance reports. Each vehicle in the dataset possesses a minimum of 106 sensor channels. The multi-channel sensor data was collected daily over a period of 21 months between 2012 and 2014. The dataset is scheduled to be released publicly after detailed anonymization.

A subset of data with 15 sensor channels and one additional time channel UTC_1Hz is selected for our anomaly detection problem as per the recommendation of a group of mechanical engineers working on the same dataset. The selected 15 sensor channels belonging to the four main FWGs of a vehicle are the basis for the data analysis, model training, and studies performed in this paper. UTC_1Hz sensor channel records the date and timestamp for every observation of data. It is

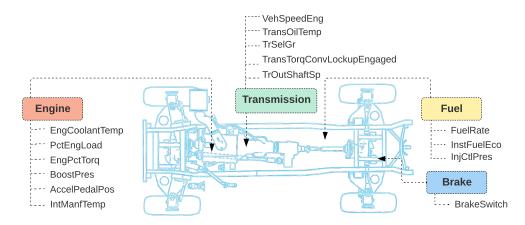


Fig. 3: A graphical representation of the various FWG in a vehicle, including the *engine, transmission, fuel*, and *brake*, as well as the sensor channels for each of these FWG [18].

collected at a frequency of 1 sample per second, starting at the timestamp value when the engine is switched on.

The selected vehicular sensor channels can be visualized in Figure 3 and a summary of these channels is provided in Table I. These channels specifically deal with certain ECUs installed in different vehicular FWGs, such as *engine, transmission, fuel,* and *brake*. Figure 4 illustrates raw observations for a window of 25 minutes on 2014-Jan-30. Next, we describe each of the selected FWGs and associated sensor channels in detail.

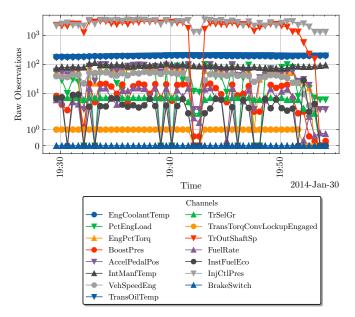


Fig. 4: Snapshot of VePro Dataset raw observations for a period of 25 minutes on 2014-Jan-30.

A. Engine FWG and Sensor Channels

The engine FWG converts energy from combustion to mechanical work. It includes the following sensor channels:

1) EngCoolantTemp (Engine Coolant Temperature): The EngCoolantTemp sensor channel measures the temperature of the engine's liquid coolant. The sensor operates by sensing the temperature emitted by the thermostat or the coolant itself. The

vehicle's computer will use this information to keep the engine at the optimal temperature.

2) PctEngLoad (Percent Engine Load): The engine load determines the demand placed on a motor's ability to generate power. When the engine load increases, the engine speed decreases. To compensate for this decrease in engine speed, additional fuel is injected into the engine. PctEngLoad sensor channel records fractional power of an engine compared to its maximum manufacturer's design capacity at various engine conditions, such as a vehicle in motion or a vehicle at idle.

3) EngPctTorq (Engine Percent Torque): Torque is commonly referred to as 'twisting or turning force'. In the context of automobiles, it measures the rotational force applied by the piston to the crankshaft of the engine. The EngPctTorq sensor channel is the calculated output torque of the engine. The data is recorded as a percentage of net engine torque [34].

4) BoostPres (Turbo Boost Pressure): The BoostPres sensor channel indicates the air pressure information at any given moment. It provides information regarding air pressure and air-to-fuel ratios for regulating engine performance. The ECU then uses boost and air density information in the vehicle's intake manifold to determine how much fuel is needed in the car's combustion chamber so that the air-fuel mixture is at its best.

5) AccelPedalPos (Accelerator Pedal Position): The AccelPedalPos sensor channel captures information about the acceleration, deceleration, and steady-state condition of a vehicle. It monitors the accelerator pedal's position and relays that information to the ECU. The computer opens the throttle body to increase the flow of fuel to the engine when the pedal is pushed forward. On the other hand, when the pedal is released, the throttle body is closed, and the flow of fuel is reduced.

6) IntManfTemp (Intake Manifold Temperature): The Int-ManfTemp sensor channel indicates the temperature of the air inside the intake manifold. A low sensor reading indicates that incoming air is highly dense, while a high sensor reading indicates that the incoming air is extremely thin. To balance the proper air-to-fuel ratio, the Powertrain Control Modules

| FWG | Sensor Channel | Full name | Unit | Description | | | | |
|--------------|--------------------------------|--|----------------------------------|--|--|--|--|--|
| Time | UTC_1HZ | 1 Hz Data | Seconds | Time series data from the 1 Hz channels. | | | | |
| | EngCoolantTemp | Engine Coolant Temperature | Degree Fahrenheit (°F) | Captures how heavy the engine is working and correlates well with the fuel rate. | | | | |
| | PctEngLoad | Percentage Engine Load | Percent (%) | Amount of load required for the engine to perform driving. | | | | |
| | EngPctTorq | Engine Percentage Torque | Percent (%) | The calculated output torque of the engine. | | | | |
| Engine | BoostPres | Turbo Boost Pres- sure | Pound per square inch (psi) | Indicates the air pressure information at any given moment. | | | | |
| | AccelPedalPos | Acclerator Pedal Position | Percent (%) | Captures acceleration, deceleration, and steady state condition. | | | | |
| | IntManfTemp | Intake Manifold Temperature | Degree Fahrenheit (°F) | Indicates the temperature of the air inside the intake manifold. | | | | |
| | VehSpeedEng | Vehicle Speed | Miles Per Hour (m/s) | Speed of vehicle - major contributor to the fuel consumption. | | | | |
| | TransOilTemp | Transmission Oil Temperature | Degree Fahrenheit (°F) | Captures temperature in all operating condition. | | | | |
| | TrSelGr | Transmission Selected Gear | Unitless | Represents which transmission gear the vehicle is currently in. | | | | |
| Transmission | TransTorqConv LockupEngaged | Transmission Torque Converter Lockup Engaged | Base 10 Integer Num- ber | Captures whether or not the torque converter lockup is engaged. | | | | |
| | TrOutShaftSp | Transmission Out- put Shaft Speed | Revolutions Per Minute (rpm) | Calculates the transmission gear ratio when in use. | | | | |
| | FuelRate | Fuel Rate | Gallons (U.S.) Per Hour (gph) | The fuel rate is essentially how quickly the vehicle is burning fuel. | | | | |
| Fuel | InstFuelEco | Instantaneous Fuel Economy | Gallons (U.S.) Per Hour (gph) | Captures current fuel economy at current vehicle velocity. | | | | |
| | InjCtlPres | Injector Control Pressure | Pound per square inch (psi) | Can be used as a health indicator of the injectors. | | | | |
| Brake | Brake BrakeSwitch Brake Switch | | Base 10 Integer Num- ber | Captures the position of break pedal. | | | | |

TABLE I: A brief overview of the various vehicular FWGs included in the study, along with a description of the associated sensor channels.

(PCM) increase the amount of fuel to the engine in the event of a low-temperature reading, whereas the PCM reduces the amount of fuel to the engine in the event of a high-temperature reading.

B. Transmission FWG and Channels

The transmission FWG controls and transfers power from the engine to the driveshaft. The following sensor channels are associated with this FWG:

1) VehSpeedEng (Vehicle Speed): The VehSpeedEng sensor channel represents the speed of a vehicle at a given time. The PCM manipulates numerous electrical subsystems in a vehicle using the VehSpeedEng frequency signal, including fuel injection, ignition, cruise control operation, torque, and clutch lock-up. On the other hand, transmissions rely on vehicle speed data to optimize shift strategy, and it is a significant determinant of fuel usage.

2) TransOilTemp (Transmission Oil Temperature): The TransOilTemp sensor channel captures the temperature of the transmission fluid under all operating conditions. It measures transmission fluid temperature by transmitting a voltage signal proportional to fluid temperature to the ECU; i.e., the lower the voltage, the higher the temperature.

3) TrSelGr (Transmission Selected Gear): The TrSelGr sensor channel represents what transmission gear the vehicle

is currently in. The gear ratio values set a torque limit on the engine output for a certain range of transmission gears.

4) TransTorqConvLockupEngaged (Transmission Torque Converter Lockup Engaged): The TransTorqConvLockupEngaged sensor channel indicates whether or not the torque converter lockup is engaged. It is represented in binary form, i.e., 0 for disengaged and 1 for engaged. The transmission lockup torque converter has a clutch. The engagement of this clutch locks the engine to the input shaft of the transmission, resulting in a direct 1:1 drive ratio. It is used to enhance fuel consumption efficiency.

5) TrOutShaftSp (Transmission Output Shaft Speed): TrOutShaftSp sensor channel is used to calculate the transmission gear ratio when in use. It detects the rotational speed of the output shaft, which is derived from the output shaft tone wheel.The Transmission Electronic Control Unit (TECU) looks at the speeds of the input shaft, the output shaft, and the main shaft to figure out the gear ratios of the main case, the auxiliary case, and the whole transmission.

C. Fuel FWG and Channels

The fuel FWG is responsible for the fuel material needed to produce heat when ignited with oxygen. This FWG includes the following sensor channels:

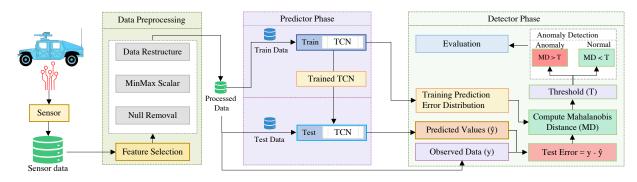


Fig. 5: An overview of the proposed three-phased system architecture. In the initial phase, raw sensor data is cleansed, transformed, and scaled. The second phase predicts the sequence of outputs. In the third phase, anomalies are identified by comparing error values to a predefined threshold.

1) FuelRate (*Fuel Rate*): The *FuelRate* sensor channel essentially captures how quickly the vehicle is burning fuel. It measures the amount of fuel a vehicle consumes to go a specific distance [35].

2) InstFuelEco (Instantaneous Fuel Economy): The Inst-FuelEco sensor channel represents current fuel economy at current vehicle velocity.

3) InjCtlPres (Injector Control Pressure): The InjCtlPres sensor channel is the health indicator of the injectors. It detects the pressure of the fuel entering the injectors. It checks the pressure of the fuel going into the injectors and sends that information to the computer so that it can make the changes needed for the best performance and efficiency.

D. Brake FWG and Channel

The brake FWG is responsible with inhibiting motion by absorbing energy. The position of the brake pedal is captured by the *BrakeSwitch* sensor channel. The brake switch's sensor measures this position. Its principal purpose is to release the converter clutch during deceleration.

IV. System Architecture & Methodology

For a vehicle to work normally, it is important that all of its Functional Working Groups (FWG) work properly. When one of the vehicle's FWG fails, the vehicle's performance deteriorates. Examples of FWGs include engine, fuel, cooling, air compressor, electrical, transmission, axle, suspension, brake, wheel, and tire related channels. For this study, four FWGs: engine, transmission, fuel, and brakes, have been included for modeling and evaluation (Figure 3).

The architecture of our system, shown in Figure 5, is divided into three phases: data preprocessing, the prediction phase, and the anomaly detection phase. We developed our anomaly detection methodology utilizing this phased approach. In the rest of this section, we describe each phase and expand on our methods.

A. Data Preprocessing

In the first phase of our architecture, the multi-channel data stream is preprocessed before the predictor phase. The dataset used in this study has 106 sensor channels with 955,856 observations. We selected 15 suitable sensor channels based on the advice of mechanical engineers working on the same data. These have been described in Section III. Figure 4 shows raw data observations for these sensor channels during a window of 25 minutes collected on 30-Jan-2014. To process the data, we removed null values and redundant observations from the dataset. Next, we normalized the data using the min-max scaling. The dataset is further processed using a sliding window approach and conditioned to meet the input requirements for the next phase (See Figure 6 as an example). At the end of this phase, for a given time window, the processed data will be formatted as a sequence containing a certain number of observations per channel.

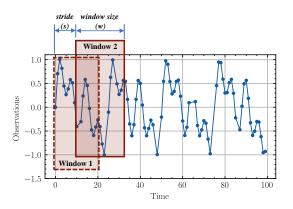


Fig. 6: Sliding window example. A window is comprised of w = 20 observations. Consecutive windows can be separated by utilizing a stride value.

B. Predictor Phase

In the second phase, the predictor performs time series forecasting. TCN is utilized as the predictor model due to its ability to abstract time series data and its use of flexible receptive fields [10]. In addition, these models are computationally efficient and possess representational capability to achieve robust and superior prediction performance [11]. In our model, time series forecasting is defined as predicting the next value from an input sequence of observations.

A sequence-to-sequence (Seq2Seq) learning approach is used due to the temporal nature of the dataset. The processed data is restructured as sequences, i.e., series of contiguous observations. In order to generate sequences, a sliding window is used on the processed data. Figure 6, illustrates an example of the sliding window approach. The size of the window (w) is the number of observations to consider as a sequence. In order to avoid interpolation and to ensure that sequences are composed of contiguous observations, the *time elapsed* (et) between the first and last observation in a window (endpoints) is calculated. The *expected data rate* (dr) for this dataset is 1 Hertz (Hz). Due to dropped packets in streaming data, an additional delta *time* (dt) is added when considering sequences. For example, consider a sequence of 10 observations with a 1 Hz data rate that has an elapsed time of 12 seconds. This sequence would be considered valid if dt is set to 2. In other words, a sequence is valid if: $et \leq (w/dr + dt)$ seconds. All other sequences are removed since they contain non-contiguous observations. Non-contiguous observations can be due to gaps in the data when the vehicle is idle or not in use. A fixed stride is also used to control the amount of overlap between windows. The final processed dataset is a three-dimensional array of the form $N \times M \times P$, where N is the number of instances, M is the length of the sequence, and P is the number of channels. For our experiments, the restructured dataset has 15 sensor channels, each with w = 20 observations.

The restructured data is split into two parts: training and testing. For this study, we used 70%-30% random splitting criteria for training and testing. Due to the random split, training and testing datasets contain sequences throughout the 21 month period between 2012 and 2014. Each sequence is broken into two sub-sequences. The first M-1 observations are used as the input sequence, and the last one is used as the target output. The predictor is a sequential model with an temporal convolutional layer and a dense layer. The temporal convolutional layer has the following hyperparameters: 64 filters; kernel size of 3; causal padding; single residual block stack; ReLU activation function; and dilations of 1, 2, 4, 8, 16, and 32. The size of the receptive field is 505 (See Section II-A for details). The dense layer has $1 \times P$ units and uses a linear activation function. A 10% validation split is used on the training data to ensure that the model does not overfit. All the abovementioned hyperparameters were determined experimentally during optimization.

C. Detector Phase

In the final phase, we perform anomaly detection. The objective of this phase is to determine whether or not a FWG is functioning normally. The predicted observation, calculated in the previous phase, is compared to the actual observation. In general, if the prediction error exceeds a threshold, then the system estimates an abnormal observation.

The anomaly detection technique used in this paper is outlined in the Algorithm 1. Once the predictor has been trained, Algorithm 1 Training and Testing a TCN based Anomaly Detection System

Inputs:

data multivariate time series observations

 tcn_{hp} TCN hyperparameters

Outputs:

Model trained temporal convolutional network model μ_{pred} training prediction error mean Σ_{pred} training prediction error covariance T anomaly threshold

CM confusion matrix

Variables:

X model input, y model target, \hat{y} model prediction, fa actual anomaly label, fp predicted anomaly label, trn training data, tst test data, ζ prediction error, MD Mahalanobis distance

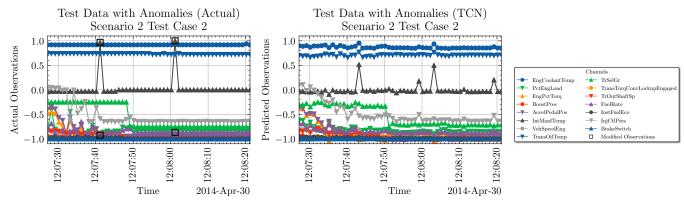
1: $X_{trn}, y_{trn}, X_{tst}, y_{tst}, fa_{tst} = \text{split}_\text{dataset}(data)$ 2: $Model \leftarrow build_model(tcn_{hp}, X_trn, y_trn)$ 3: $\hat{y}_{trn} = \text{predict}(Model, X_{trn})$ 4: $\zeta_{trn} = y_{trn_i} - \hat{y}_{trn_i}$ 5: $\mu_{pred} = \text{mean}(\zeta_{trn})$ 6: $\Sigma_{pred} = \operatorname{cov}(\zeta_{trn})$ 7: $\hat{y}_{tst} = \text{predict}(Model, X_{tst})$ 8: $\zeta_{tst} = y_{tst} - \hat{y}_{tst}$ 9: Select T10: $MD = \sqrt{(\zeta_{tst} - \mu_{pred})^T \Sigma_{pred}^{-1} (\zeta_{tst} - \mu_{pred})}$ 11: if MD > T then 12: $f p_{tst} = Anomaly$ 13: **else** 14: $f p_{tst} = Normal$ 15: end if 16: $CM = confusion_matrix(fp_{tst}, fa_{tst})$

the training dataset is used to calculate the estimated prediction error. The estimated (training) prediction error is assumed to follow a multivariate Gaussian distribution. It is important to note that the training data may contain a very small number (α) of abnormal instances due to sensor measurement noise. Another assumption is made that the vehicle operates under normal conditions for a large number (β) of training instances (i.e., $\beta >> \alpha$). The Mahalanobis distance is used to determine how closely the test prediction error matches the estimated distribution.

Anomalies are inserted in 20% of the test dataset. Then, the predictor is used to make predictions on the test dataset. As stated before, the Mahalanobis distance is used as a metric to detect anomalies. Finally, the performance of the anomaly detection system is evaluated using receiver operating characteristic (ROC) curve and area under the curve (AUC) 8. The MD threshold is derived from the ROC curve by using the geometric mean of sensitivity and specificity. The Confusion Matrix III and detection accuracy can be formulated using a MD threshold.

| TC | FuelRate | AccelPedalPos | TrSelGr | Result | TC | FuelRate | IntManfTemp | Result | TC | FuelRate | InjCtlPres | Result |
|----------------|-------------------|-------------------|-------------------|--------|----------------|-------------------|-------------------|--------|----|-------------------|-------------------|--------|
| 1 | \Leftrightarrow | \Leftrightarrow | \Leftrightarrow | FALSE | 1 | \Leftrightarrow | \Leftrightarrow | FALSE | 1 | \Leftrightarrow | \Leftrightarrow | FALSE |
| 2 | \Leftrightarrow | ₩ | ↓↓ | TRUE | 2 | \Leftrightarrow | ↑ | TRUE | 2 | ⇔ | ↑ | TRUE |
| 3 | \Leftrightarrow | ↑ | ↓ | TRUE | 3 | \Leftrightarrow | \downarrow | TRUE | 3 | \Leftrightarrow | \Downarrow | TRUE |
| 4 | ↑ | ↑ | ↑ | FALSE | 4 | ↑ | \Downarrow | FALSE | 4 | ↑ | \Downarrow | TRUE |
| 5 | ↑ | ₩ | ↓ | TRUE | 5 | \Downarrow | ↑ | TRUE | 5 | ↓ | ↑ | TRUE |
| 6 | \downarrow | ↑ | ↑ | TRUE | 6 | ↑ | ↑ | TRUE | 6 | ↑ | ↑ | FALSE |
| 7 | ₩ | \Downarrow | ₩ | FALSE | 7 | \Downarrow | \Downarrow | TRUE | 7 | ₩ | \Downarrow | FALSE |
| (a) Scenario 1 | | (b) Scenario 2 | | | (c) Scenario 3 | | | | | | | |

TABLE II: \Leftrightarrow represent normal condition, \Uparrow represent rise and \Downarrow represent fall. TC refers to test cases for a scenario. Scenario 1 illustrates the normal and abnormal relationships between the *FuelRate, AccelPedalPos*, and *TrSelGr* channels. The second scenario presents the relationship between *FuelRate* and *IntManfTemp* in normal and abnormal conditions. Scenario 3 illustrates the normal and abnormal relationship between *FuelRate* and *IntCtrlPres*.



(a) Synthetic anomalies inserted into test data.

(b) TCN predictions on test data.

Fig. 7: Plot (a) illustrates a snapshot of synthetic anomalies mixed into the test data for a time period of 54 seconds on 2014-Apr-30. Scenario 2, Test Case 2 involves an increase in *IntManfTemp* and an unchanged *FuelRate*. Modifications are bounded within -1.0 and 1.0. Black square markers denote modified observations. Plot (b) shows TCN predictions on the test data.

V. EVALUATION AND RESULTS

To evaluate our system, we must ensure that the anomaly detection model is robust enough to predict both normal and abnormal states for FWGs. This means that when the sensor channels have normal readings, the model must not raise a flag. Conversely, if there is any abnormal data collected from the sensor channels, then the model should raise an alert. To test the system's ability to find sensor channels in states that are not normal, we mix in some synthetic data with real data to create a number of test scenarios. Next, we describe these anomalous text scenarios. These were also created in consultation with mechanical engineers working on the same dataset.

A. Anomalous Test Scenarios

To evaluate the vehicle's fuel system, we must consider three types of information: (1) performance sensor channels such as *FuelRate*; (2) sensor channels that affects fuel rate such as *AccelPedalPos, VehSpeedEng, PctEngLoad*, and *Eng-PctTorq*; and (3) fuel rate associated sensor channels such as *InjCtlPres, IntManfTemp*, and *TrSelGr*. The detailed description of these channels can be found in Section III and Table I. On the basis of this information, we develop three scenarios containing 21 test conditions, as shown in Table II. Our model should identify these anomalies accurately. For example, the model should not predict the first test case in Table IIa as an anomaly, whereas it should accurately predict the second case as an anomaly.

1) Scenario 1: When the AccelPedalPos is pushed, the ECU opens the throttle body to increase the flow of fuel to the engine and closes it when the pedal is released. Ideally, with rapid acceleration, the *TrSelGr* should be higher. Using this scenario, we construct seven test cases as depicted in Table IIa.

2) Scenario 2: In the normal operating conditions, if the *IntManfTemp* sensor channel experiences a low reading, then it is considered that the air coming to the intake manifold is highly dense, and if the reading is high, the air is considered to be thin. To balance things, more fuel is injected into the engine when the internal manifold temperature is low, and fuel flow is reduced when the manifold temperature is high. Using this scenario, we generate the seven test cases as shown in Table IIb.

3) Scenario 3: In the case of the *InjCtlPres* sensor channel, when the pressure is increased in the injectors, it causes the valve to open wide, leading to an increase in fuel flow into the engine. Based on this assumption, the test conditions shown

in Table IIc are formulated.

B. Insertion of Anomalous Observations to Test Dataset

Using the abovementioned test scenarios, anomalous observations are inserted into the testing data. Figure 7 illustrates an example of inserting synthetic anomalies into the test data and predictions made by the TCN model. This is done by assigning 20% of testing instances an anomaly scenario (type) at random. The assigned scenario dictates the particular sensor channels and the direction of modification. If an instance is chosen to be anomalous, then the target observation is modified based of the assigned scenario. The magnitude of the modification is also chosen at random from the set (1, 1.5). Note that modified values are bounded between -1 and 1.

C. Results

Before discussing the results, we define performance metrics that we used in the context of our anomaly detection system. We classify a sensor channel reading as a true positive (TP) only if the model detects an abnormal sensor channel reading as an anomaly, and a true negative (TN) when a normal channel sensor reading is detected as normal. When the model detects a normal sensor channel reading as an anomaly, it is defined as a false positive (FP), whereas an actual abnormal sensor channel reading which is not detected is a false negative (FN). Using these definitions, we evaluate the proposed method based on two performance metrics:

1) Detection Accuracy: Quantifies the detection system's ability to correctly identify an anomaly, as defined below:

Detection accuracy
$$= \frac{TP + TN}{TP + FP + TN + FN}$$
 (2)

 Receiver Operating Characteristic (ROC) curve with area under the curve (AUC): Measures the performance of the proposed model as the area under the curve in a plot between the true positive rate (TPR) and false positive rate (FPR):

$$TPR = \frac{TP}{TP + FN}$$
, $FPR = \frac{FP}{FP + TN}$ (3)

The predictor TCN model was trained using validationbased early stopping. The overall result of the anomaly detection system is shown using the ROC curve (see Figure 8). The area under the curve (AUC) of the system is 0.982. Using a MD threshold 9.61, 91% of true anomalies were detected. The proposed system has an detection accuracy of 0.96 (See Table III).

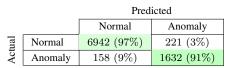


TABLE III: Confusion matrix. A Mahalanobis distance threshold of 9.61 is used to detect anomalies.

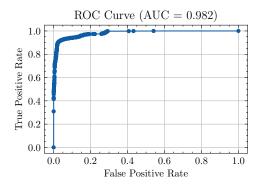


Fig. 8: ROC Curve on test dataset.

The composition of anomalous instances per scenario was: 518 from Scenario 1, 707 from Scenario 2, and 565 from Scenario 3. In total, there were 1790 anomalous instances. The following is the breakdown of missed anomalies by the system: 49 missed in Scenario 1, 3 missed in Scenario 2, and 106 missed in Scenario 3. This suggests that the system performs better when testing with channels from multiple functional working groups. Additionally, by using the TCN model prediction, the system can further isolate the anomaly to a specific channel and FWG (see Figure 7b). Overall, the system successfully identified vehicular anomalies and states well outside normal operating conditions (e.g., potentially unsafe states).

VI. CONCLUSION & FUTURE WORK

In this paper, we introduce a comprehensive anomaly detection system for vehicles utilizing multiple Functional Working Groups (FWGs). FWGs provide information to the anomaly detection system in the form of multisensor time series data. The anomaly detection model was developed using the Vehicle Performance, Reliability and Operations (VePRO) dataset, which contains usage data from a fleet of U.S. Department of Defense vehicles.

To detect anomalies, a multi-phased approach is employed with a Temporal Convolution Network (TCN) used as a kernel to learn the underlying relationships between sensor channels. The TCN model was trained using the VePRO data, which contains normal operating conditions. The model performs time series forecasting by operating on a multi-channel sequence for the current time window and outputs a prediction for the next observation. The final anomaly determination is made by comparing the prediction with the actual observation. Through a series of targeted scenarios and tests, the system is able to isolate abnormal behavior to a specific FWG. The results of this study show that our system successfully identifies vehicular anomalies and unsafe states.

In the future, a logical next step is to expand on the forecasting model to consider long sequences using RNNbased LSTM methods. Given the architectural differences between LSTM and TCN methods, it would be interesting to compare the predictive and anomaly detection capabilities of these two algorithms. In addition, we would like to investigate the performance of hybrid approaches, such as ConvLSTM [36], for anomaly detection in vehicular operational datasets.

ACKNOWLEDGEMENT

This Work was supported by PATENT Lab (Predictive Analytics and TEchnology iNTegration Laboratory) at the Department of Computer Science and Engineering, Mississippi State University; U.S. Department of Defense grant through U.S. Army Engineer Research and Development Center (ERDC); National Science Foundation (#1565484). The views and conclusions are those of the authors.

REFERENCES

- Yongyi Ran, Xin Zhou, Pengfeng Lin, Yonggang Wen, and Ruilong Deng. A survey of predictive maintenance: Systems, purposes and approaches. arXiv preprint arXiv:1912.07383, 2019.
- [2] Weiting Zhang, Dong Yang, and Hongchao Wang. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, 13(3):2213–2227, 2019.
- [3] Tiago Zonta, Cristiano André Da Costa, Rodrigo da Rosa Righi, Miromar Jose de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150:106889, 2020.
- [4] Thyago P Carvalho, Fabrízzio AAMN Soares, Roberto Vita, Roberto da P Francisco, João P Basto, and Symone GS Alcalá. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering*, 137:106024, 2019.
- [5] Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. arXiv preprint arXiv:1901.03407, 2019.
- [6] Donghwoon Kwon, Hyunjoo Kim, Jinoh Kim, Sang C Suh, Ikkyun Kim, and Kuinam J Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, 22(1):949–961, 2019.
- [7] Xiaoxiao Ma, Jia Wu, Shan Xue, Jian Yang, Chuan Zhou, Quan Z Sheng, Hui Xiong, and Leman Akoglu. A comprehensive survey on graph anomaly detection with deep learning. *IEEE Transactions on Knowledge* and Data Engineering, 2021.
- [8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):1–58, 2009.
- [9] Morteza Alizadeh, Michael Hamilton, Parker Jones, Junfeng Ma, and Raed Jaradat. Vehicle operating state anomaly detection and results virtual reality interpretation. *Expert Systems with Applications*, 177:114928, 2021.
- [10] Yangdong He and Jiabao Zhao. Temporal convolutional networks for anomaly detection in time series. In *Journal of Physics: Conference Series*, volume 1213, page 042050. IOP Publishing, 2019.
- [11] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271, 2018.
- [12] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 156–165, 2017.
- [13] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431– 3440, 2015.
- [14] Sridhar Alla and Suman Kalyan Adari. Beginning anomaly detection using python-based deep learning. Springer, 2019.
- [15] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.
- [16] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122, 2015.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

- [18] OPERATOR'S MANUAL us army ground vehicles. https://www. hmmwvinscale.com/HMMWV%20TM.pdf, 1993. Accessed: 06-20-2022.
- [19] Jianwei Liu, Hongwei Zhu, Yongxia Liu, Haobo Wu, Yunsheng Lan, and Xinyu Zhang. Anomaly detection for time series using temporal convolutional networks and gaussian mixture model. In *Journal of Physics: Conference Series*, volume 1187, page 042111. IOP Publishing, 2019.
- [20] Sooryaa Vignesh Thiruloga, Vipin Kumar Kukkala, and Sudeep Pasricha. Tenet: Temporal cnn with attention for anomaly detection in automotive cyber-physical systems. In 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC), pages 326–331. IEEE, 2022.
- [21] Saroj Gopali, Faranak Abri, Sima Siami-Namini, and Akbar Siami Namin. A comparison of tcn and lstm models in detecting anomalies in time series data. In 2021 IEEE International Conference on Big Data (Big Data), pages 2415–2420. IEEE, 2021.
- [22] Yuan Wang, Yan Wu, Qiong Yang, and Jun Zhang. Anomaly detection of spacecraft telemetry data using temporal convolution network. In 2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pages 1–5. IEEE, 2021.
- [23] Ronghong Mo, Yiyang Pei, Neelakantam Venkatarayalu, Pereira Nathaniel, AB Premkumar, and Sumei Sun. An unsupervised tcn-based outlier detection for time series with seasonality and trend. In 2021 IEEE VTS 17th Asia Pacific Wireless Communications Symposium (APWCS), pages 1–5. IEEE, 2021.
- [24] Morteza Alizadeh, Shahram Rahimi, and Junfeng Ma. A hybrid arimawnn approach to model vehicle operating behavior and detect unhealthy states. *Expert Systems with Applications*, 194:116515, 2022.
- [25] Somayeh Bakhtiari Ramezani, Brad Killen, Logan Cummins, Shahram Rahimi, Amin Amirlatifi, and Maria Seale. A survey of hmm-based algorithms in machinery fault prediction. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–9, 2021.
- [26] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi. Obd_securealert: An anomaly detection system for vehicles. In 2016 IEEE International Conference on Smart Computing (SMARTCOMP), pages 1–6. IEEE, 2016.
- [27] Sai Sree Laya Chukkapalli, Priyanka Ranade, Sudip Mittal, and Anupam Joshi. A privacy preserving anomaly detection framework for cooperative smart farming ecosystem. In 2021 Third IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), pages 340–347. IEEE, 2021.
- [28] Sowmya Ramapatruni, Sandeep Nair Narayanan, Sudip Mittal, Anupam Joshi, and Karuna Joshi. Anomaly detection models for smart home security. In 2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), pages 19–24. IEEE, 2019.
- [29] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi. Using semantic technologies to mine vehicular context for security. In 2016 IEEE 37th Sarnoff Symposium, pages 124–129. IEEE, 2016.
- [30] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi. Using data analytics to detect anomalous states in vehicles. *arXiv e-prints*, pages arXiv–1512, 2015.
- [31] Jaeyong Kang, Chul-Su Kim, Jeong Won Kang, and Jeonghwan Gwak. Anomaly detection of the brake operating unit on metro vehicles using a one-class lstm autoencoder. *Applied Sciences*, 11(19):9290, 2021.
- [32] Nicolas Cheifetz, Allou Samé, Patrice Aknin, and Emmanuel De Verdalle. A pattern recognition approach for anomaly detection on buses brake system. In 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pages 266–271. IEEE, 2011.
- [33] Howard E Bussey, Nenad G Nenadic, Paul A Ardis, and Michael G Thurston. Case study: Models for detecting low oil pressure anomalies on commercial vehicles. In *Annual Conference of the PHM Society*, volume 6, 2014.
- [34] Rodney A Rohrer, Joe D Luck, Santosh K Pitla, and Roger Hoy. Evaluation of the accuracy of machine reported can data for engine torque and speed. *Transactions of the ASABE*, 61(5):1547–1557, 2018.
- [35] National Research Council et al. Assessment of fuel economy technologies for light-duty vehicles. National Academies Press, 2011.
- [36] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.