Anomaly Detection for Time Series Data using VAE-LSTM Model

Randall Fowler
Department of ECE
UC Davis
rlfowler@ucdavis.edu

Conor King
Department of ECE
UC Davis
cfking@ucdavis.edu

Ajay Suresh
Department of ECE
UC Davis
ajsuresh@ucdavis.edu

***Abstract*—Here is an abstract...**

***Index Terms*—Anomaly Detection, VAE, LSTM, Time Series Data, IMU, PPG, EV Drive Cycle, Deep Learning**

I. Introduction

This is an introduction...

Anomaly detection in electric vehicles (EV) can be extremely useful for ensuring the reliability, safety, and efficiency of EVs. Identifying deviations from normal patterns, it helps in early detection of potential issues, preventing costly breakdowns and ensuring optimal performance. While there are a wide variety of applications within the EV space, this report choses to focus on the usage of anomaly detection for vehicle speed and battery voltage as examples.

Anomaly detection for vehicle speed has several potential applications, including:

- **Drivetrain Monitoring:** Detecting anomalies in vehicle speed helps in identifying issues with the drivetrain, including motor performance, transmission, and control systems.
- **Driver Behavior Analysis:** Anomalies in speed patterns can also be used to analyze and improve driver behavior, ensuring safer and more efficient driving practices.
- **Autonomous Driving Systems:** In autonomous vehicles, maintaining consistent speed is crucial. Anomaly detection in speed data helps in ensuring that the autonomous driving system is functioning correctly and safely.

Similarly, anomaly detection for battery voltage is also crucial for the following reasons:

- **Battery Health Monitoring:** Detecting anomalies in vehicle speed helps in identifying issues with the drivetrain, including motor performance, transmission, and control systems.
- **Energy Management Systems:** Anomalies in speed patterns can also be used to analyze and improve driver behavior, ensuring safer and more efficient driving practices.
- **Safety and Reliability:** In autonomous vehicles, maintaining consistent speed is crucial. Anomaly detection in speed data helps in ensuring that the autonomous driving system is functioning correctly and safely.

II. Methodology

Anomaly detection is represented by measuring the reconstruction error from a given deep learning model. For this system, a VAE and LSTM hybrid model is used for finding a latent space with a known distribution and capturing temporal information over time series data. This data is presented in a windowed fashion to locate anomalies within a given window. Evaluation of the anomaly detection will be measured using the harmonic mean of precision and recall (F1 score).

A. Preprocessing

The nature of the data used for this detection system does not need to be specific, but it does require being time series data. With a given window size, a sliding window will sample the original signal, and the VAE input and output will be equivalent to the size of the window.

As this data does not need to be labeled, it does need to be clean of noise or anomalies. This is due to the model learning the clean distribution of the data, and after training, the poor reconstruction of a signal will indicate the possibility of an anomaly.

B. Variational Autoencoder

An autoencoder structure establishes a method for compressing information into a dense latent space. This space is not unique as the original signal can be encoded into a large number of different latent spaces. The VAE addresses this non-regularization issue of the latent space by forcing the space to follow a specific distribution [**?**]. For most VAEs, this distribution is a normal distribution, and the output of the encoder will be a prediction of mean and standard deviation. Once the distribution of the latent space is known, a sampling will be performed to create a latent vector for the input of the decoder. When the latent space distribution is well formed, the output of the decoder aims to reconstruct the original signal as best as possible.

As the autoencoder represents encoder and decoder models, these structures can be created in a variety of ways. For this project, both will be presented by 1-dimensional convolutional layers with a leaky ReLU activation function and batch normalization. The number of samples per window will decrease while depth increases for every layer in the encoder structure. At the end of the encoder, the samples and depth are flattened and passed to linear layers to condense to the final latent space mean and standard deviation values. Samples from the normal distribution will be given to a decoder that is identical to the encoder, but with transpose convolutional layers to increase the size and decrease the depth. Padding for each layer is forced to maintain a consistent linear decrease in the number of samples. For this architecture, four convolutional layers are used with a split set of linear layers for the mean and standard deviation prediction.

C. Long Short-Term Memory

With a sampled latent space created from the encoder, the LSTM takes the current latent vector and attempts to predict the next sample. If the VAE is trained well, it should hold the distribution for the dataset, and the LSTM should learn the temporal differences between each latent window. The LSTM tracks previous outcomes by utilizing memory variables or acting as a state machine. Cell memory and hidden state become the long-term and short-term aspects of the model, and those components carry into the model as inputs [**?**]. After each iteration, the memory variables will output to be used by another LSTM layer or dropped before other layers.

A standard LSTM layer contains a tangent activation function, and the PyTorch library does not provide an option to adjust the activation function [**?**]. For this reason and the lack of activation function post latent space sampling, a linear layer is placed after the final LSTM layer to adjust for scaling.

D. VAE-LSTM Model Training

Since the two models hold separate functions for the detection system, the training for the VAE and LSTM are handled separately. Both will use the same training set as the models will be combined after training. Batch size, loss function, and optimizer may vary between the two models.

Loss function can differ greatly as the VAE can be trained with Kullback-Leibler (KL) divergence. This loss relates to the similarity between the latent space distribution and the unit normal distribution [**?**]. Essentially, it enables the latent space to hold this normal distribution shape. Reconstruction loss is another term that focuses on training the weights regardless of the latent space. The sum of the similarity and reconstruction loss promotes a smooth and continuous latent space while improving signal reconstruction.

The LSTM only needs to focus on the error for prediction on known future samples. This error may be the same as reconstruction loss for the VAE. With the memory components, these need to be initialized properly when training. For each given batch, the batch needs to remain in sequential order, and the memory should be reset if batches are not in order. For predictions on an entire dataset, the memory should only be reset at the very beginning. When resetting, the cell memory and hidden state are initialized with a Normal Xavier Initialization [**?**]. This sets the weights using a normal distribution with a mean of 0 and a variance dependent on the number of inputs and outputs of a layer.

E. Anomaly Detection

After training, anomalies are detected using the reconstruction error of the predicted sample and thresholding this value. When an anomaly occurs, the model will predict the known distribution and notice the disruption after decoding. The VAE can detect anomalies without the LSTM, but it will be lacking the temporal aspects and recognize the anomaly on reconstruction error alone.

Since the original signal will be windowed prior to input, each window will have an error associated with predicting an anomaly. To represent the anomaly detection in the original signal rather than the windowed version, each data point in the window will be stored in its respective position. Each data point will be the average of all window errors that data point is utilized in. For the first data point in the original signal and first window, this error will be the error of the first window. The second data point will be the average error between the first window and second window. Once the last data point of the first window is reached, this point will be averaged over the same number of windows as the window size, and all following data points will be averaged the same until the final window is reached.

The purpose of averaging error over windows provides a smoother transition between several points where an anomaly would be. It is unlikely that a signal data point will hold an anomaly as typically, an anomaly will occur over a range of time. This issue is also addressed in the next section to increase the score if a section of an anomaly is detected.

F. Evaluation Metrics

Evaluation of the detection system is handled using an F1 score as this is a measure of precision over sensitivity [**?**]. Precision is the accuracy of positive predictions, and it is divided by the total positive predictions. Sensitivity, or recall, is the rate of true positives where the number of true positives is divided by the sum of true positives and false negatives. While accuracy of predictions may be a good metric, F1 score considers the distribution of predictions as it is the harmonic mean between the precision and recall.

Since anomalies typically occur in sections of a signal rather than pointwise, assessment per anomaly region could be a better method for measuring true positives. One strategy is to assume any anomaly segment in the ground truth that is detected would be considered a correct detection for that segment, and any point outside of that segment would be treated normally [**?**]. This method will be referred to as the augmented F1 scoring, and both the regular and augmented F1 score will be evaluated.

III. Experiments

Due to the restrained time, the exploration of hyperparameters, architecture, and optimization techniques were limited to first implementations.

The window size was set to be 100 samples of the original signal with a stride of 1. Each prediction by the LSTM is dedicated to a single window, but since the stride is one, this is a sample-by-sample prediction. Padding for each layer is forced to maintain a consistent linear decrease in the number of samples. Different window sizes should be explored to relate to the different range of time noticed by the model in each moment, but with the difference between signal size and convolutional layers, the exploration was neglected. Four convolutional layers are used in the encoder with a split set of linear layers for the mean and standard deviation prediction, and the decoder layers are identical to the encoder. In the latent space, the LSTM is also given 4 LSTM layers to capture a temporal distribution in the latent space. Number of parameters for the entire detection model was nearly 1.25 million parameters.

For the loss function of both models, mean square error (MSE) is selected for ease. Binary cross entropy (BCE) was considered for loss, but the input and output would be difficult to constrain between 0 and 1. One challenge related to training with similarity loss, KL divergence, as the calculated values did not seem appropriate. This is expected to be a small bug in the implementation, but for the sake of time, it was not used. Without the KL divergence, the VAE would resemble a standard autoencoder.

Training was conducted with varying batch sizes and number of epochs between different datasets, and an Adam optimizer with a learning rate of 4e-4 and betas as 0.9 and 0.95 was selected.

To explore the robust design of the detection model, three datasets are used for training and testing. IMU dataset consists of motion recordings from multiple sensors to detect movement or changes to the monitoring system. Synthesized PPG data is created from simulating tissue models and aims to detect motion, electromagnetic interference, and other variations of noise. The Electric Vehicle Drive Cycle dataset is synthesized based on referenced data from a Cadillac EV that includes time series data for vehicle speed and battery voltage.

A. Inertial Measurement Unit Dataset

Here is the dataset...

B. Synthesized Photoplethysmography Dataset

Photoplethysmography (PPG) is a non-invasive optical technique used to detect blood volume changes in tissue. Light is beamed into tissue in one area and light intensity is detected in a different area; as blood is the dominant light absorber in tissue, changes in the intensity correlate with changes in blood volume. Different wavelengths of light are absorbed differently by oxygenated and deoxygenated hemoglobin, and different light source-detector separation distances probe different tissue depths. By using multiple wavelengths and detectors in a single run, the resultant time-series signals are rich in information about blood volume and oxygenation in different areas of the tissue.

The data used in this experiment is synthesized using parameters taken from in-vivo data gathered from pregnant sheep. The time-varying frequency trace and the pulse shapes were taken directly from the sheep-data, while the magnitudes of the different components were determined via Montecarlo simulation. Thus, the data strongly resembles in-vivo data but is free from anomalies and is suitable for training the VAE-LSTM model. There are four sets of time-series data, each with 10 channels, corresponding to 5 detectors and two wavelengths.

To test the model, the same data is augmented with artificial anomalies. The anomalies are short intervals with a DC shift. The number of anomalies, their positioning, their duration, and their intensity are all determined probabilistically, and are varied slightly between channels, also probabilistically. The data-points affected by the DC shifts are the true anomalies against which the trained VAE-LSTM model is tested. Figure 1 provides an example of the clean data and the anomalous data, with true anomalies highlighted in red.

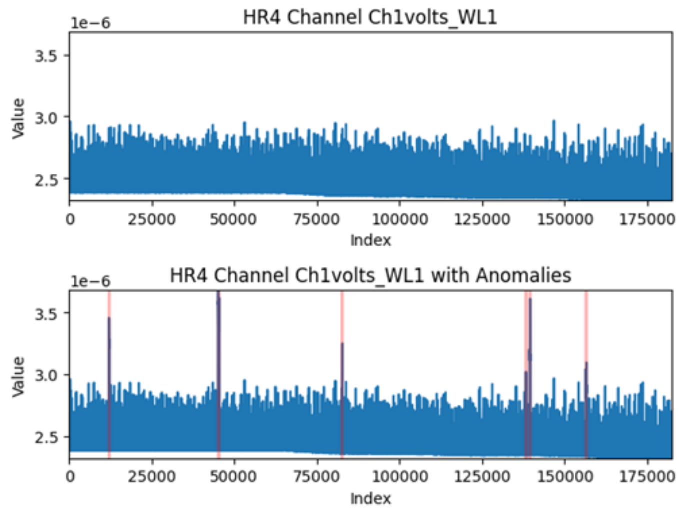

Fig. 1. Example of synthesized PPG data (top) and the same data with added anomalies (bottom)

C. Electric Vehicle Drive Cycle Dataset

The dataset consists of time-series data representing the speed of an EV over a drive cycle. The speed data is generated to simulate a realistic driving cycle with a sine wave pattern and added noise to showcase both overall and subtle realistic changes to speed between 0 and 60 mph. The dataset was designed and generated in comparison with actual city and highway drive cycle data from a Cadillac EV. Generating a synthetic dataset here allows for experimentation with further variability of speed and allows for wider coverage while maintaining the frequency of expected changes that a real dataset would see. A visualization of the synthetic speed dataset without anomalies is included in Fig 2.

This dataset represents the voltage levels of an EV battery over a drive cycle. The data is generated to simulate typical voltage patterns for a 400V battery pack, with a sine wave pattern and noise - similarly generated in comparison with actual data obtained from a Cadillac EV. A visualization of the synthetic battery voltage dataset is included in Figure 3.

IV. Results

Here are the results...

A. Inertial Measurement Unit

Here is the dataset...

B. Synthesized Photoplethysmography

The performance of the VAE-LSTM model on the synthesized PPG data with artificial anomalies was somewhat varied. As expected, it depends enormously on the chosen threshold value.

When the best threshold without anomaly augmentation is used, the performance is generally good, as exhibited by Figure 4. The small anomaly to the far right of the data is not detected, but its magnitude is so small that that seems reasonable.

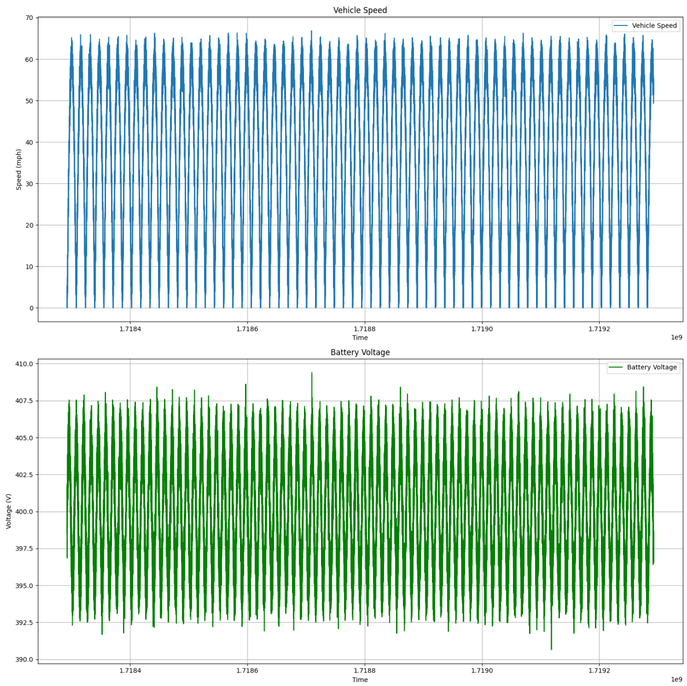

Fig. 2. Synthetic Data showcasing Vehicle Speed

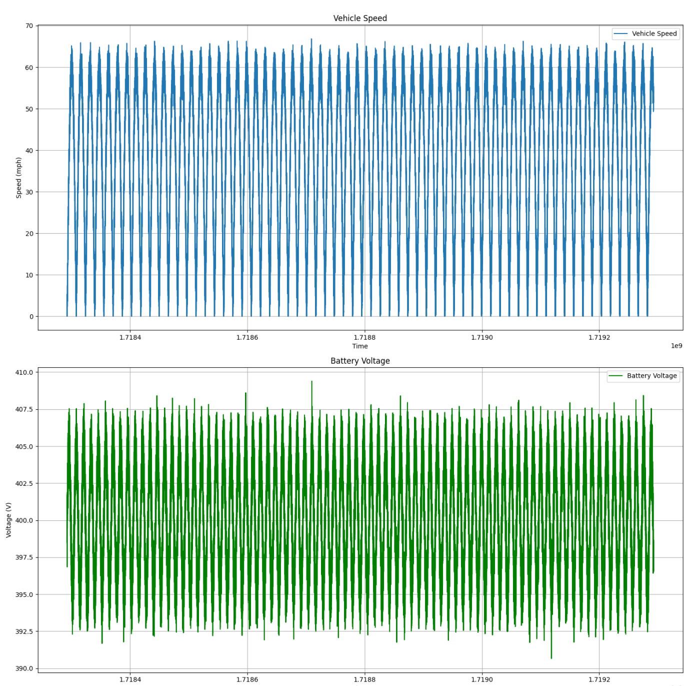

Fig. 3. Synthetic Data showcasing Battery Pack Voltage

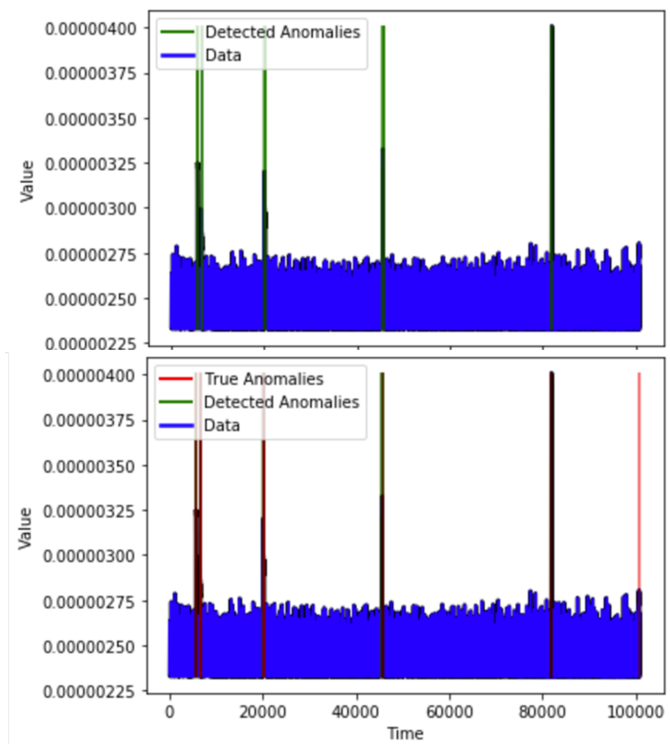

Fig. 4. Example of good performance of the model. Slight false negative on the far right

However, sometimes that thresholding level causes the model to suffer from extreme overprediction, and the best threshold found during testing with anomaly augmentation works better, as in Figure 5.

Often, a value between the two "best" values actually produced the best results. In the case of Figure 6, the best threshold when augmentation was not used as 0.1 (top), and the best threshold when augmentation was used was 5.5 (bottom). The top plot shows extreme overprediction of anomalies while the bottom misses the leftmost anomaly and a smaller anomaly in the middle. The middle plot, using a hand-selected threshold of 0.5, successfully finds the anomalies without overpredicting.

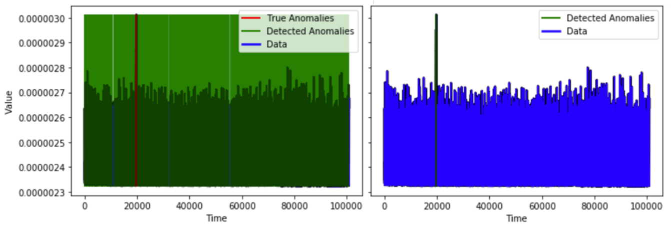

Fig. 5. Left: Best threshold w/o augmentation. Right: Best threshold w/ augmentation

Overall, these results suggest that the model learned the structure of the synthesized PPG data well and is able to reliably predict anomalies when a suitable threshold is chosen. However, finding a systematic way of selecting a good threshold proves more challenging.

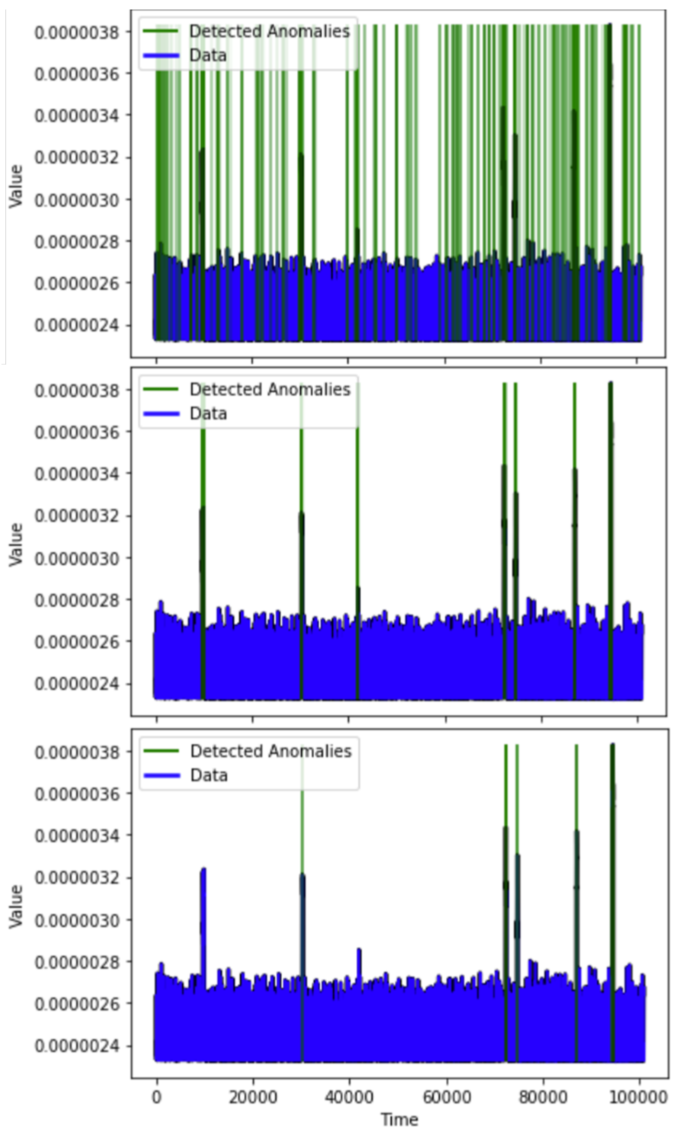

Fig. 6. A threshold value between the two ostensible best values produces better results than either

C. Electric Vehicle Drive Cycle

1) Vehicle Speed: Figure 7 shows the synthesized and normalized test dataset to evaluate the VAE + LSTM model to detect anomalies within vehicle speed. Random data points were selected and a speed spike of 20 to 40 mph were introduced to simulate an actual spike in vehicle speed that wouldn't normally occur during a regular drive cycle. The goal was to fine tune the model to be able to detect just the 5 red dots (speed spikes) that are visible in the figure.

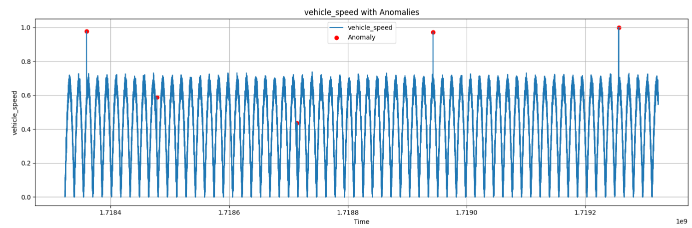

Fig. 7. Vehicle Speed with introduced anomalies (spikes in speed)

The results of Figure (8) showcase an overdetection of anomalies within the dataset as showcased by the green lines in the figure. Although this has been a common issue amongst all tested datasets, in this case, the variations in speed demonstrate realistic changes that are fairly noisy - which can be difficult to separate from a real anomaly unless further tuning is performed.

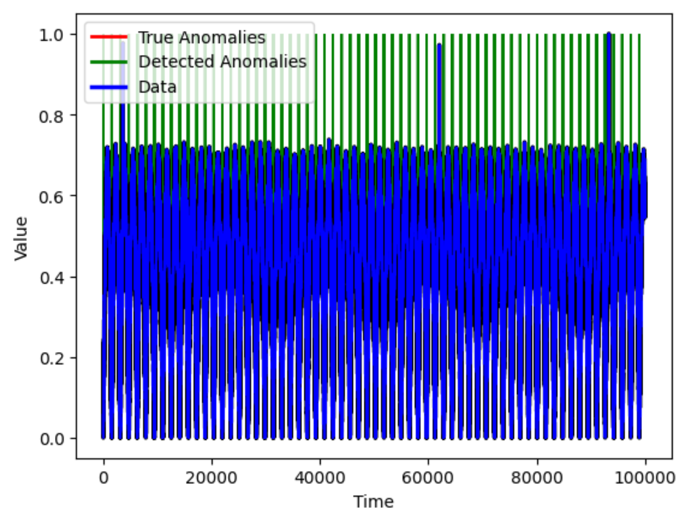

Fig. 8. Results of VAE + LSTM Anomaly Detection for Vehicle speed

2) Battery Voltage: Figure 9 shows the synthesized and normalized test dataset to evaluate the VAE + LSTM model to detect anomalies for battery pack voltage. In the context of this experiment, anomalies in battery voltage are simulated by introducing sudden drops in the voltage levels. The drop in voltage is simulated by subtracting a random value (e.g., between 5V to 10V) from the current voltage at the selected anomaly points - which are above the realistic thresholds of a voltage drop during regular operation.

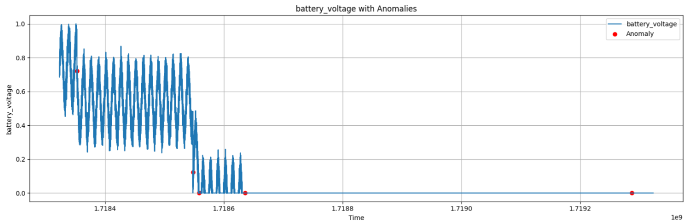

Fig. 9. Battery Pack Voltage with introduced anomalies (sudden drops in voltage)

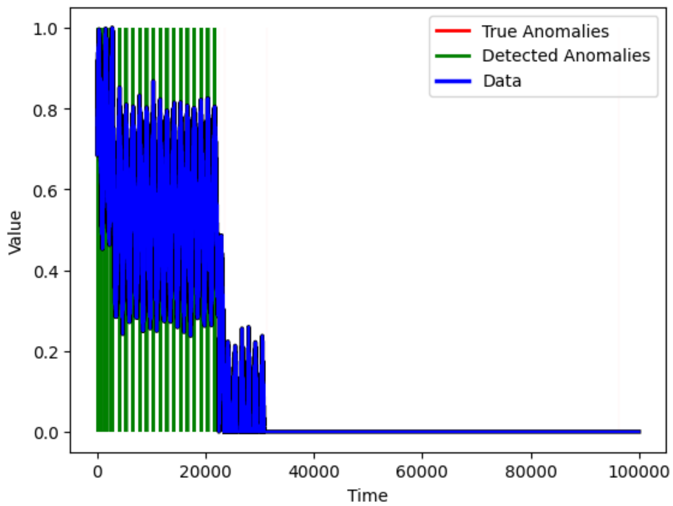

Fig. 10. Results of VAE + LSTM Anomaly Detection for Battery Pack Voltage

As with the previous results, this dataset also showcases an overdetection of anomalies with some anomalies on the lower range going undetected (lack of green lines after time = 20,000, where at least 2 are expected).

V. Conclusion

Here is a conclusion...

A. Future Work

Here is some future work...

Acknowledgment

The authors would like to thank Dr. Yubei Chen for his guidance and support on this course project.

Contributions

Randall Fowler developed the models, evaluation methods, and experimented with the IMU dataset. Conor King experimented with the synthesized PPG dataset. Ajay Suresh experimented with the EV drive cycle dataset. All authors contributed to the writing of the paper.

References

[1] Citation 1
[2] Citation 2
[3] Citation 3
[4] Citation 4
[5] Citation 5