Proceedings of the IISE Annual Conference & Expo 2023
K. Babski-Reeves, B. Eksioglu, D. Hampton, eds.

Ensemble learning for Fault Condition Prediction and Health Status Monitoring in Military Ground Vehicles

Abstract ID: 1479

Sungkwang Mun[1], Joonsik Hwang[1], Linkan Bian[1], TC Falls[2], and W Glenn Bond[3,*]
[1] Mississippi State University: Center for Advanced Vehicular Systems (CAVS), Mississippi State, MS, 39759, USA
[2] Mississippi State University: Institute for Systems Engineering Research (ISER), Vicksburg, Mississippi, 39180, USA
[3] US Army Engineer Research and Development Center, Vicksburg, MS, 39180, USA
[*] Corresponding author: William.G.Bond@erdc.dren.mil

Abstract

Unexpected technical issues such as engine and transmission failure represent critical Reliability, Availability, and Maintainability (RAM) issues for military ground vehicles. It is essential to keep the vehicles in healthy condition and to provide predictive maintenance that enables efficient diagnosis and repair of vehicle failures, reduces associated operation/sustainment costs and vehicle downtime, and supports predictive logistics for critical components. This paper develops a predictive model for early fault detection using a machine learning (ML) algorithm trained by a real vehicle data set. The ML model is built on various operational time series sensor data and fault codes collected via a Digital Source Collector and a CAN bus device for several vehicles over extensive time intervals. The approach proposed here is an ensemble learning of a multivariate Long Short-Term Memory (LSTM) neural network for operational data forecasting based on select channel data such as coolant temperature, engine oil temperature/pressure, and battery voltage with respect to the driving status. The LSTM model can use recorded parameters to classify vehicle or component reliability and health. For further improvement of prediction accuracy and generalization of the model across various vehicle types, multiple independent LSTM models are trained over training/validation datasets from the randomly subsampled period of a single vehicle or those from the same families of vehicles. Outputs from the individual networks are then linearly combined to produce the output of the ensemble network. The analysis shows better prediction accuracy than the single LSTM approach, providing promising early fault detection performance.

Keywords
Artificial Neural Network, Machine Learning, and Predictive Maintenance

1. Introduction

Monitoring trends and analyzing patterns in data of a complex system such as military ground vehicles can provide insight into the health status of a vehicle by enabling the prediction of component fault or failure in one vehicle or even over a fleet. Anomalies or outliers in operational data that fall outside the normal operational profile of a vehicle can represent performance issues, wear, or symptoms of imminent component failure that require further investigation. It is crucial to detect such anomalies, present in the time series sensor data, to provide decision support for operations, maintenance, and logistics. Recently, various deep learning models have been proposed for detecting anomalies in time series data, including autoencoder (AE), Long Short-Term Memory (LSTM), recurrent neural network (RNN), convolutional neural network (CNN), and hybrid approaches such as CNN-AE, with some challenges including lack of defined pattern of anomaly, noise in the data, and irregular size of the time series[1]. In this paper, we propose a method to detect abnormal periods in multi-channel sensor data of vehicles based on Long Short-Term Memory (LSTM) neural networks by differentiating the predicted time series values by the model trained over a normal period of time and the observed values. Here, the normal period of time is gathered from the time series data associated with the maintenance log data. Maintenance logs are searched for service or repair of critical components such as batteries, engines, or transmissions to establish a time interval surrounding logged faults. The time series data, before and after the maintenance event, is then classified as normal and abnormal operation. The rest of the paper is organized as follows. Section 2 describes how we selected the training/validation data for machine learning algorithms. Section 3 briefly describes the LSTM network and ensemble learning. Section 4 details overall simulation procedures and prediction results, followed by the conclusion section.

2. Problem Description

2.1 Data Selection

The time series data used in this paper is collected from more than 3,000 US Army ground vehicles of various types, each with over 100 sensors, with associated maintenance event data, curated at the US Army Engineer Research and Development Center (ERDC)[2]. The sensor data includes various signals, such as temperature, pressure, and RPM, at operation conditions recorded at 1Hz frequency via Digital Source Collector (DSC) and fault signals recorded via Controller Area Network (CAN) bus. Maintenance data consists of scheduled and unscheduled maintenance logs that include the type of failure, type of correction, labor hours, and costs. It should be noted that all data are associated with date and time information providing hashed vehicle identification number (VIN) with associated metadata. Sensor and maintenance data were independently collected, leading to instances that are not perfectly matched in terms of vehicle or time. Training and validation data were chosen, therefore, based on the availability of the operational/fault sensor data and relevant maintenance event documents, as shown in **Figure 1**.

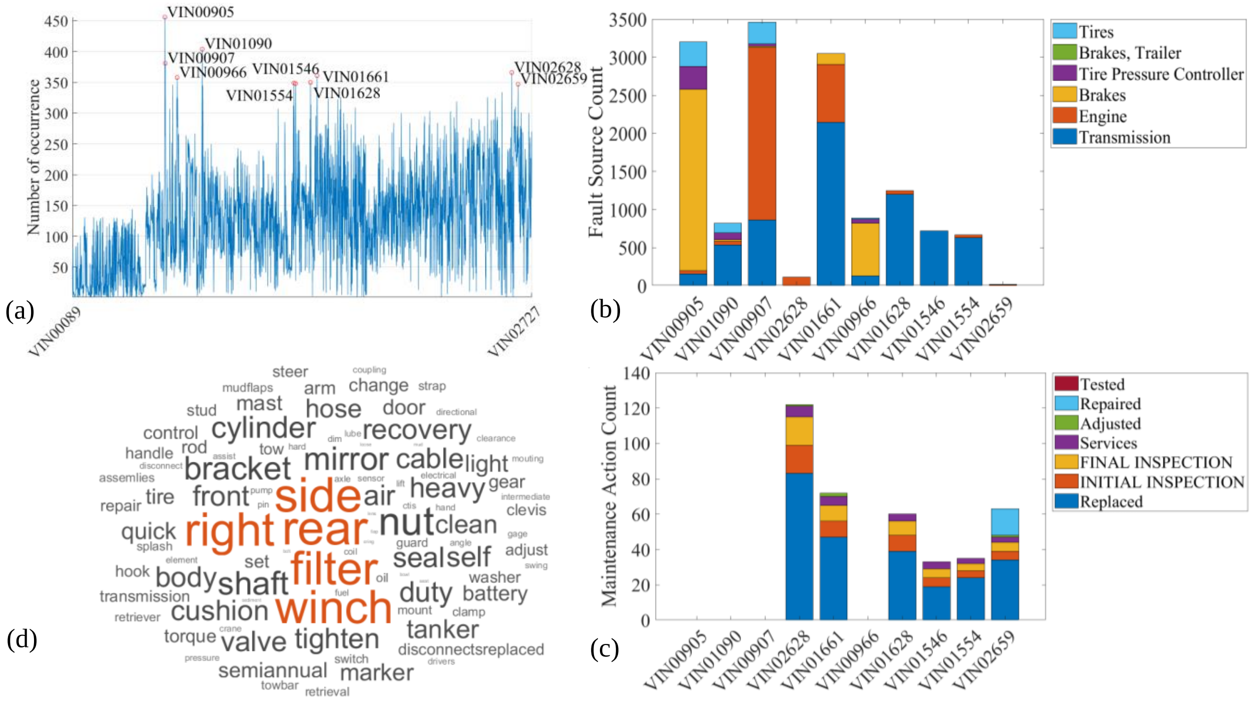

Figure 1. (a) number of available daily logs of individual vehicles with indication (red circle) of the top ten most daily used vehicles (b) stacked bar chart with respect to various fault source types for the top ten vehicles (c) stacked bar chart with respect to various maintenance action type for the top ten vehicles with no available data for VIN00905, VIN01090, VIN00907, and VIN00966 (d) Word cloud for overall maintenance actions in subfigure (c). Note subfigures (c) and (d) are reverse ordered for better visual comparison of subfigures (b) and (c).

Figure 1(a) depicts the number of daily records that contain the operational 1Hz stream of data from various sensors in individual vehicles, where the red circle indicates the top ten most used vehicles. Specifically, VIN00905 (model 21, Family 3), VIN01090(Model 28, Family 3), VIN00907 (Model 21, Family 3), VIN02628 (Model 84, Family 8), VIN01661 (Model 62, Family 4), VIN00966 (Model 20, Family 3), VIN01628 (Model 58, Family 4), VIN01546 (Model 58, Family 4), VIN01554 (Model 58, Family 4), and VIN02659 (Model 87, Family 7) were operated 456, 404, 381, 366, 361, 358, 350, 349, 348, and 347 times, respectively, during the data collection period. Note that the 1st-3rd,4th and 7-9th fall under the same families, Family 3 and Family 4, respectively, while 1st,3rd and 7-9th vehicles are the same model, Model 21 and Model 58, respectively. The overall fault counts with respect to different fault types, such as transmission, engine, brakes, and tires, for the select top ten most daily used vehicles are shown in **Figure 1**(b). Likewise, the overall maintenance counts concerning the different actions performed during maintenance events for the identical vehicles are shown in **Figure 1**(c). As aforementioned, collections of sensor data and maintenance logs were performed independently, so maintenance data for some vehicles are unavailable. Such examples include VIN00905, VIN01090, VIN00907, and VIN00966. To give an example of maintenance action, a word cloud of overall maintenance actions of the ten vehicles is plotted in **Figure 1**(d), where frequent words are gradually enlarged and emphasized with orange color. Several observations can be made here. First, the most fault signals are from transmission (45%, 6377 out of the total faults 14194), followed by engine (24 percent). The primary fault source for the select vehicles is the transmission, while the number of engine faults was greater in VIN00907 than in other units. Second, 'Replaced' takes the majority in the maintenance actions as 51 percent of the total 385 actions. Finally, a low fault count does not necessarily mean that the vehicle has continuously operated in normal conditions. For example, VIN02628 and VIN02659 that show low fault source counts had significant maintenance events. These analyses tell us that the vehicles of choice are sufficiently representative of the entire vehicle collection, as it provides a sufficient about of various fault types and maintenance events for ML training of the model.

2.2 Identifying training and test data using maintenance logs

For supervised learning, it is critical to have quality training data. It holds true for supervised anomaly detection in time series data like sensor signal processing. Even though the operational time series data are not explicitly tagged with the health status of the components or vehicle as a system, fault sensor data provide such information. Moreover, information in the maintenance logs also helps identification of normal operations by correlating replacement or repair actions with the relevant sensor signals. For example, the battery voltage signal may appear different before and after the replacement of the component due to its failure, and engine-related service or repair can be correlated with the signals such as engine load and coolant/oil temperature. Therefore, the operational sensor data can be effectively divided into two groups of data periods before and after a component is repaired or replaced with an assumption that the well-trained model over a normal operation period can identify an abnormal period by differentiating the prediction of the supposedly normal operation from the actual signal profile. As an example, **Figure 2** shows the time when battery replacement occurred over the exact timeline of maintenance and the measured voltage signals.

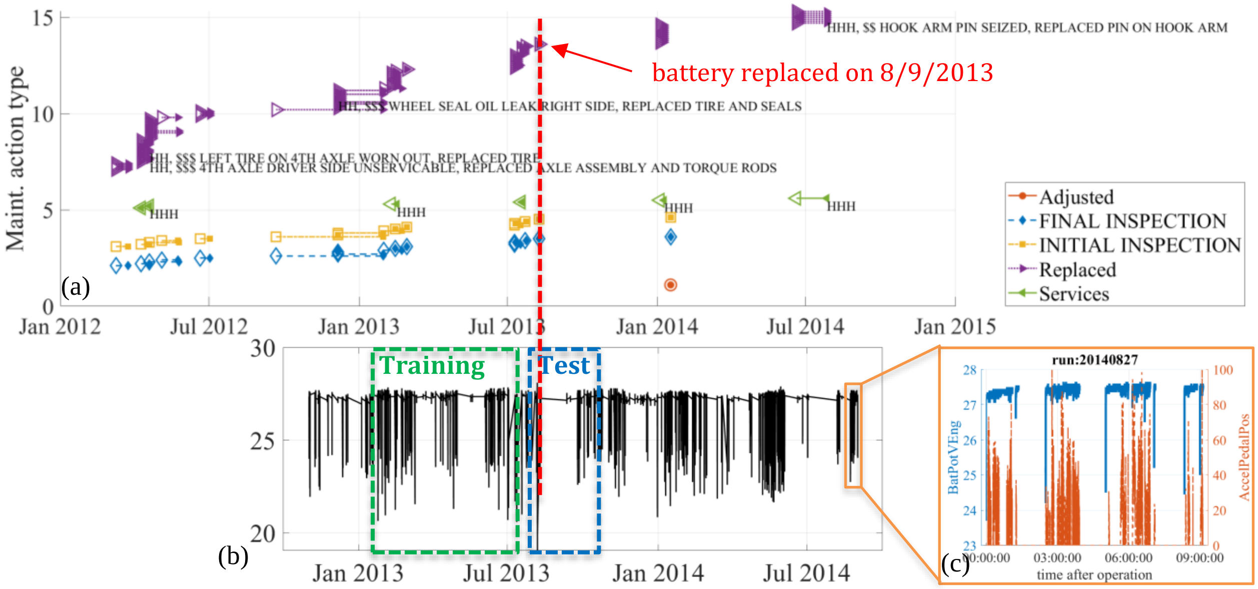

Figure 2. (a) Maintenance timeline of VIN02628. Each color marker indicates the maintenance action type. Unfilled and filled markers indicate the diagnosed date and repair date, respectively. Descriptions with the repeat of 'H' or '$' indicates labor and expense, respectively, followed by fault description and correction narrative. (b) Battery voltage level whose DateTime is aligned with the maintenance timeline. The green/blue box indicates the data used for training and testing, respectively. Test data include those collected before and after the battery was replaced. (c) Zoom-in view of the daily signal after the first operation (left: battery voltage, right: accelerator pedal position)

Specifically, **Figure 2**(a) visualizes the maintenance timeline for different maintenance action types. As the distance between filled and unfilled markers indicates, some maintenance tasks, such as inspections and replacements, require a longer time to complete the correction actions than the services (green markers). The red dash line denotes that the battery was replaced on 8/9/2013, and it can be seen that the battery was replaced during an unscheduled maintenance period, as it is done after the service period. The red line extends to **Figure 2**(b) that is the actual battery voltage signal profile arranged to match the maintenance timeline. As a green and blue box indicates, we separated the data into training data six months before the month when the battery was replaced and test data of the month the battery was replaced and a month before (two months for the test). It should be noted that six months period for the training is arbitrarily set, the different duration can be used. Also, the data of other vehicles in the same family can also be used for training. One important consideration for machine learning training is timestamp information of the associated signal. As shown in **Figure 2**(c), daily operations may include multiple short-time operations instead of a single, long period, so training awareness of such information that represents the potentially arbitrary time intervals between the operations will improve prediction accuracy.

3. Ensemble Learning using the Long Short-Term Memory Model (LSTM)

Long Short-Term Memory (LSTM) neural networks are deep learning networks that use information from the past to address time dependency in the signal. LSTM overcomes the limiting tendency of recurrent neural networks (RNNs): to learn long-term dependencies due to the vanishing of the gradient values during the back-propagation process. Unlike RNNs that use only the immediately previous state obtained through the feedback loop, LSTMs utilize memory cells and three types of gates known as 'forget,' 'input,' and 'output' to control the flow of information into and out of the cell, allowing it to remember values over long-time intervals. We refer the interested reader to [3] or [4] for detailed definitions of how LSTM networks are used in anomaly detection. Like many other neural networks, LSTM can be used for classification and regression. We used the LSTM for regression here. Often, the trained network does not correctly respond to novel, unobserved inputs outside of the training dataset, which are called "overfitting" issues. These issues can be addressed by appropriately sizing the network and tuning hyperparameters. Furthermore, the use of the validation set, a portion of the training set, also helps avoid these issues by keeping the parameter values at the iteration with the best validation error during the training. To further reduce overfitting and achieve generalization, a dropout layer can be added to the LSTM layer as a stack. Dropout is a type of regularization method widely used in convolutional neural networks, especially for image classification applications, but dropout is also applicable as an LSTM layer. It randomly sets the weight parameter values in the hidden neurons to zero with a given probability value between 0 and 1 with a proper rescaling of the remaining neurons' values at each training iteration[5]. Higher number results in more elements being dropped during the training. As the dropout is only used during training, it is treated as a linear layer, simple direct mapping inputs to outputs, at prediction. Ensemble learning is a method that generates several models that are combined to make a prediction used in either classification or regression problems [6], which is typically composed of three steps; model generation (construct redundant models), model pruning (eliminating some models based on the performance) and model integration (combine the remained models). In this work, we linearly combined the outputs of several LSTM models through simple average.

4. Results

4.1 Data access and Preprocessing

Timestamped multi-channel sensor data were fetched from MongoDB in the Data Analytics and Visualization System (DAVS) framework, an outcome of a joint research program sponsored by the US Army ERDC, between Mississippi State University, and Hottinger, Brüel and Kjaer Solutions, LLC (HBK)[2]. Data accessing, preprocessing, training, and predictions were conducted using MATLAB software (version 2022a). Due to CAN bus architecture and the digital source collector interface, the data collected from the sensors infrequently contains measurement errors, timestamp mismatches, or unsynchronized timestamps among different sensor channels collected during the same time interval. Therefore, cleaning and imputation are required for machine learning training. First, after the database query return, the raw data are cleaned by removing 'not-a-number' or missing values and corresponding timestamp data. Similarly, out-of-sync data/time stamp pairs are simply removed. In the case of multi-channel data, only the subset of sensor channels that are perfectly matched in time is selected. Second, using auxiliary speed-related data, such as vehicle speed or acceleration pedal position data, we obtain the time point when the vehicle is first operated during a day and exclude the data from that point until 300 seconds after this initial time point, to account for engine warm-up. Third, smoothing is performed on the data to further reduce potential sensor errors and uncertainty. The straightforward choice is moving average using a fixed window length where the sliding window over the given data outputs the average over the elements within each window. Finally, all preprocessed operational sensor data are normalized to have zero mean and a unity standard deviation in order to process multidimensional sensor data with different ranges of values.

4.2 Parametric study

We performed a series of simulations to find an optimal network structure and setup that give minimum training and test error between the prediction and ground truth data while seeking maximum deviation over the abnormal period by varying the input type or changing the network setup. The training/test dataset used for this study is the battery voltage signal and corresponding timestamps for VIN02628 collected between 01/01/2013-06/30/2013 and 07/01/2013-08/31/2013, which are 516,080 data points from 83 daily logs and 41,837 from 18 logs, respectively. The detailed hyperparameters of interest are listed in **Table 1**. As to the neural network structure, the hyperbolic tangent is used as an activation function across all hidden units in LSTM and a dense layer, a linear activation function for the final output layer. To correctly measure the performance improvement, all simulations except the ensemble learning results were performed with a random seed, which accounts for initial parameter values and validation set selection, held fixed. For the same reason, the number of epochs was held fixed at 400, the number of neurons for the LSTM layer in this work is 32, and the batch size of 128. Increasing the number of neurons or epochs exhibits marginal improvement in the prediction while increasing the training time. The final LSTM stack comprises five layers; input layer, LSTM layer, dropout layer, fully connected layer, and output layer. Regarding the performance metric, there are a few choices, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) for a degree of dissimilarity between the prediction and the ground truth, while R^2 value for the degree of similarity. Here, we chose MAPE, $MAPE = \frac{1}{N}\sum\left|\frac{\hat{y}_{\text{pred}}-y_{\text{true}}}{y_{\text{true}}}\right| \cdot 100$, for throughout the comparison because of its intuitive interpretation in terms of relative error. We confirmed that other metrics follow a similar trend. **Table 1** compares prediction performances of various configurations using the entire training set, test set, and daily maximum in terms of MAPE obtained on the signal each day. As seen in the first and second cases, increasing the default initial learning rate improves the test MAPE by a factor of two, even though the max error decreased. The second case is preferred because minimizing low overall training/test errors is more important than maximizing the max daily error to capture anomalies more accurately. Likewise, employing the dropout layer with a lower probability decreases the overall error. Next, as LSTM can handle multiple input channels, the associated time information is combined with the sensor signal data as inputs to the input layer of the LSTM for training. This approach achieves better performance for all three criteria (low training/test and high daily max) over the single channel results. The window size of the moving average during preprocessing also affects the final prediction accuracy as it simplifies the problem itself by suppressing sensor errors and uncertainty. As window size increases from 11 to 51 to 101, the predictions get closer to the preprocessed data quite rapidly, especially for the training data. Reserving a portion of training data as a validation set further improves overall prediction accuracy as well. Finally, ensembling multiple LSTMs based on the same dataset but different compositions of validation data further enhances the accuracy. Here, we choose the three best models out of ten different models by random selection of validation set and combine the final outputs of all networks by averaging. As the last three results indicate, the more validation data is shuffled for ensemble learning, the better generalized the combined model is.

Table 1. Parametric study results using VIN02628 battery voltage level data. (1-channel: sensor signal, 2-channel: sensor signal with relative time information, MA(•): moving average with window size, 3 models: Ensemble of 3 best models, #iter: the number of iterations, lr: initial learning rate, and P_{do}: dropout probability)

Input data type	**Network setup**	**Training MAPE Avg. error (↓)**	**Test MAPE Avg. error (↓)**	**Test MAPE Max error (↑)**
1-channel, MA (11)	lr=0.001, no dropout layer	0.099639	0.87649	13.2455
1-channel, MA (11)	lr=0.01, no dropout layer	0.098302	0.46027	4.8844
1-channel, MA (11)	lr=0.01, P_{do}=0.5	0.099839	0.46051	5.5682
1-channel, MA (11)	lr=0.01, P_{do}=0.2	0.098505	0.43063	4.6116
2-channel, MA (11)	lr=0.01, P_{do}=0.2	0.098720	0.42507	5.0491
2-channel, MA (51)	lr=0.01, P_{do}=0.2	0.029032	0.25875	4.6764
2-channel: MA (101)	lr=0.01, P_{do}=0.2	0.016665	0.37185	7.4298
2-channel: MA (101)	lr=0.01, P_{do}=0.2, 10% valid.	0.015974	0.37073	7.4213
2-channel: MA (101), 3 models	lr=0.01, P_{do}=0.2, 10% valid.	0.015616	0.35968	7.2456
2-channel: MA (101), 3 models	lr=0.01, P_{do}=0.2, 20% valid.	0.015656	0.37219	7.4675
2-channel: MA (101), 3 models	lr=0.01, P_{do}=0.2, 30% valid.	0.015450	0.33801	6.8575

The computational time for training mainly depends on the number of iterations and the number of neurons in the LSTM layer. Most cases in this work were completed within 5 minutes when training on NVIDIA GTX 1660 TI GPU, and the prediction time for the test time series was less than milliseconds.

4.3 Prediction results

The prediction results on the test dataset are compared with the measured signals for two vehicles, VIN02628 and VIN01628, without the timestamp information in **Figure 3**(a) and (b), when the last trained model in **Table 1** is used. Note that the prediction on the plot for the training data is omitted because the MAPE for the training data set is as low as less than 0.02 percent on average. Overall prediction for the test set matches well with the measured data until the battery levels drop significantly and stay low. Below, the per-day MAPE values are plotted piecewise to match the axis of the prediction plot. A simple threshold would detect the onset of the abnormal behavior of the battery. With a one percent error threshold for “Level 1” and five percent for “Level 2”, the algorithm detected the early symptom as soon as seven and two operation days before the replacement. As final validation, the overall prediction results of a few vehicles at the full datetime range are turned into the severity level daily profile using the two threshold levels, as shown in **Figure 3**(c). The first two cases, as expected, well detect prognostic symptoms before the replacement. No abnormal signal is captured for VIN01090, while VIN00907 does have a few anomalies during the period. As shown in **Figure 1**(c), no maintenance log is available for the vehicles, but the zoom-in view of the battery voltage signal on 01/14/2014 at the “Level 2” point confirmed that the voltage dropped significantly during the operations of the day. As described in this section, our ensemble LSTM model trained only on single vehicle data (VIN02628) successfully detected the abnormal operational behaviors of multiple vehicles due to the battery problem.

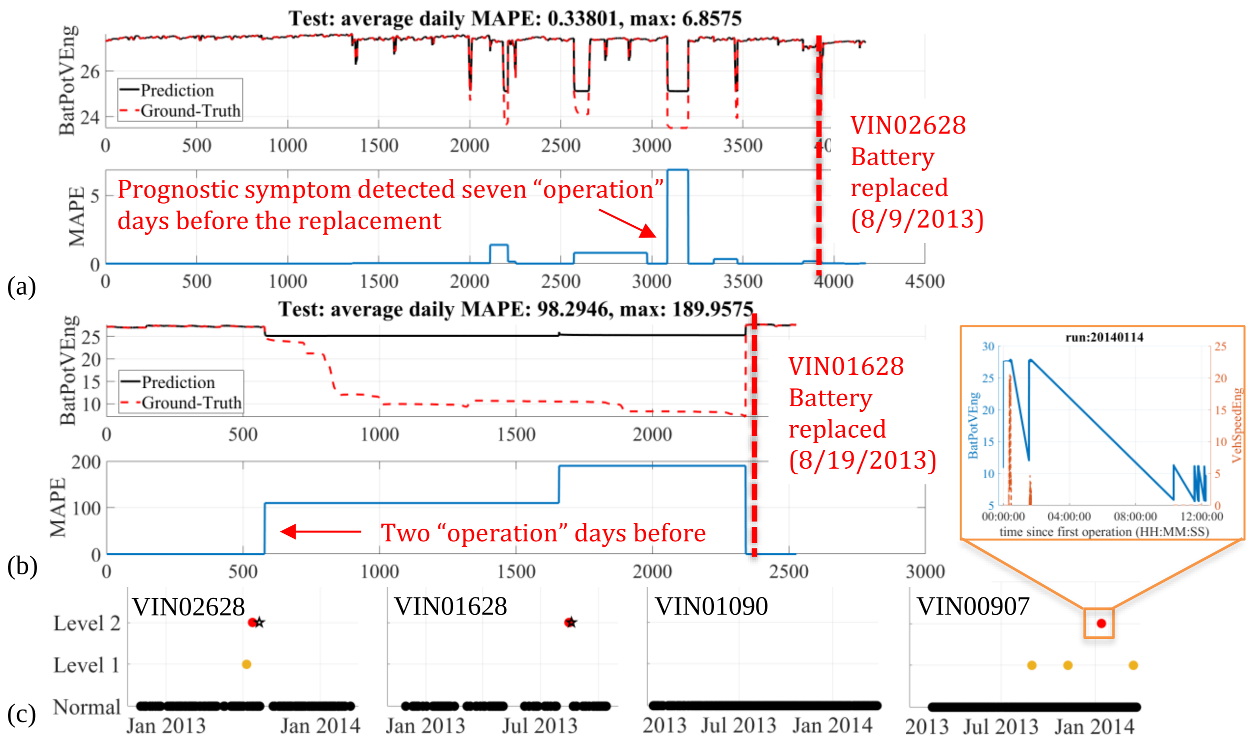

Figure 3. (a) Prediction on VIN02628 using the model trained using the data of the same vehicle (b) Prediction on VIN01628 using the model trained using the data of a different vehicle (VIN02628). (c) Daily severity level profile for select vehicles. Black star: battery replacement. For VIN00907, the battery signal on 1/14/2014 is shown in zoom-in view.

5. Conclusions

The primary objective of this research was to provide an efficient predictive approach for the early detection of fault and failure in army ground vehicles. This study addressed the issues caused by no maintenance annotations and heterogeneous data sources. To resolve the issue in an efficient way, this paper utilized maintenance event information to identify the normal conditions on operational time series data as a training database for supervised learning using the LSTM algorithm. The prediction results showed training a model does not necessarily require the data for the same vehicle. A pre-trained model using a single vehicle's training database successfully discriminates the predicted signals from abnormal periods of other vehicles that are not in the training database. This is particularly useful for the scenario when training data is unavailable; a pre-trained model can be imported for prediction or to retrain the networks with a new dataset. As for future work, the proposed approach will be generalized by extending it to other components, such as engine and transmission, and validated not only on select vehicles but also on many other vehicles in the database to demonstrate its capacity and potential for better supporting predictive maintenance of army ground vehicles.

Acknowledgements

This research was conducted under contract to the U.S. DoD High-Performance Computing Modernization Program through the US Army ERDC Contract #W912HZ21C0014.

References

[1] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey." arXiv, Jan. 23, 2019. Accessed: Jan. 23, 2023. [Online]. Available: http://arxiv.org/abs/1901.03407

[2] R. Carley *et al.*, "Data Analytics and Visualization Application for Asset Health Monitoring," *Annual Conference of the PHM Society*, vol. 14, no. 1, Art. no. 1, Oct. 2022, doi: 10.36001/phmconf.2022.v14i1.3214.

[3] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[4] P. Malhotra, L. Vig, G. Shroff, P. Agarwal, and others, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, 2015, vol. 89, pp. 89–94.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[6] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *ACM Comput. Surv.*, vol. 45, no. 1, pp. 1–40, Nov. 2012, doi: 10.1145/2379776.2379786.