Predicting Imminent Failures in Heavy-Duty Trucks: A Machine Learning Approach to Enhance Vehicle Reliability

1st Abdulkarim Dawalibi

2nd Adnan Altukleh

I. INTRODUCTION

In recent years, the field of predictive maintenance has seen significant advancements, driven by the increasing availability of vehicle operational data and the development of sophisticated analytical techniques. Current research in this field is rapidly evolving, driven by advancements in data analysis techniques and the increasing availability of detailed operational data. However, the challenge remains in accurately predicting maintenance needs in complex systems like heavy-duty trucks, where diverse operational patterns and varying repair histories complicate the prediction models [1]. The domain of this project lies at the intersection of predictive analytics and vehicle maintenance, specifically focusing on predicting maintenance requirements for heavy-duty trucks.

Our project utilizes a rich dataset provided for the 2024 IDA Challenge. This dataset comprises multiple files containing detailed operational readouts and repair records for heavy-duty trucks. A notable aspect of this dataset is its structure, designed to mimic real-world conditions. While the training data includes the complete sequence of readouts until the end of the study, the validation and test data simulate real-world usage, providing only partial information up to a randomly selected point in time. This approach aims to test the robustness of predictive models in scenarios where complete historical data might not be available [2].

Our project seeks to build and evaluate a predictive model that leverages this dataset to accurately forecast maintenance needs for heavy-duty trucks. By doing so, we aim to contribute to the ongoing research in predictive maintenance, addressing the specific challenges posed by the uneven and sporadic nature of operational data in heavy-duty vehicles. The successful development of such a model could lead to more efficient maintenance schedules, reduced downtime, and longer vehicle lifespans.

II. RELATED WORK

Focused on predictive maintenance for heavy-duty vehicles, we draw insights from a suit of research contributions that explore various machine learning (ML) and data handling techniques suitable for predictive maintenance projects.

Silvestrin et al. (2019) provide a comprehensive comparison between traditional ML algorithms and Deep learning (DL) models, emphasizing for a Long Short-Term memory network (LSTM) to perform good in predictive maintenance, it requires large amounts of labeled data to be trained effectively. Their work suggests that while LSTMs exhibit bad performance in data-limited scenarios compared to other DL methods, traditional ML algorithms based on feature engineering outperform DL methods, highlighting the importance of data characteristics in model selection [3].

Sridhar and Sanagavarapu (2021) delve into the challenges posed by data imbalance in predictive maintenance, demonstrating the efficiency of Synthetic Minority Over-sampling Technique (SMOTE)-based oversampling in enhancing model performance. Their findings reveal a significant increase in the Area Under the Curve (AUC) score, underscoring the value of addressing class imbalance through data augmentation [4].

Sharma et al. (2023) focus on the classification aspect of predictive maintenance using ML, exploring a range of algorithms to ascertain which delivers optimal accuracy and performance metrics. Their research, particularly on Random Forest, highlights the algorithm's robustness in achieving high accuracy, stressing the critical role of predictive power in enhancing equipment reliability and reducing associated costs [5].

Drawing upon these studies, our approach to the project leverages the strengths of both traditional ML and advanced DL techniques, particularly LSTMs and Random Forest, to develop predictive maintenance model/s for heavy-duty vehicles.

III. METHODOLOGY

To address the aim of this project we applied CRoss Industry Standard Process for Data Mining (CRISP-DM) process model, see *Appendix1*.

A. *Business Understanding:* Predictive maintenance stands at the forefront of transforming the heavy-duty vehicles industry. By accurately predicting equipment failures before they occur, companies can shift from reactive to proactive maintenance strategies. For Scania, a leader in manufacturing trucks for heavy-duty purposes, this transition represents an opportunity to solidify its market position by offering vehicles that boast higher reliability and lower total cost of ownership.

B. *Understanding the data:* The dataset provided by the IDA 2024 Industrial Challenge includes:

1. Operational readouts data is an unevenly sampled time series sensor readouts from each vehicle, the data consists of 107 anonymized features. (train, validation, and test)
2. Time to event data contains the information about the repair records for each vehicle and it consists of 3 features which 2 of them represent the length_of_study_time_step and in_study_repair. (train)
3. Vehicle specifications provide us with the specification for each vehicle, it consists of 9 features. The data is categorical and anonymized. (train, validation, test)
4. Labels data, it provides the class label for vehicles from class 0-4. See *Data preparation (i.1)* for clarification of what each class label indicates. (validation and test)

All datasets contain a feature that is common between all of them which is vehicle_id. The data is designed to mirror the complex nature of real-world truck operations. Offering insights into patterns and indicators which are crucial for preventing costly downtimes. Note that we did not consider the Test dataset provided by the IDA challenge since we did not participate in the challenge and it is not labeled. Instead, we considered the validation data as the test data.

C. *Data preparation:* We have conducted 4 different experiments and in each we have preperaed the data diffrently.

i. Power Spectral Density (PSD) using one class models:

1. Started with labeling the data based, by selecting the last data point for time_step feature for each vehicle in the readout data and selected the length_of_study_time _step feature for each vehicle and computed the difference between them. Then sorted the difference for each vehicle as follows, if the difference is between $0<x<6$ time unites, the vehicle is classified as class 4, if the difference is $6<x<12$ time unites, the vehicle is classified as class 3, if the difference is between $12<x<24$ time unites, the vehicle is classified as class 2, if the difference is between $24<x<48$ time unites, the vehicle is classified as class 1 and if the difference is between $48<x$ time unites, the vehicle is classified as class 0.

2. Balanced the data by selecting equal number of vehicles from each class label.

3. Computed the condition indicator PSD for each vehicle in the readout data

4. Applied feature extraction on each feature in the readout data such as mean, median, std, variance, skewness, spectral entropy, and spectral kurtosis. And transformed from multiple sequences for each vehicle to only one data point.

5. Handled Null values by replacing them with 0.

6. Merged the specification data with the procced data (readout data).

7. Prepared the data for training/evaluation by first combining the vehicles that belongs to classes 1, 2, 3 and 4 and labelled it to -1. Then selected the vehicles that belongs class 0 and labelled it as 1. Then split the data into test and training data.

8. Standardized the splited data, train and test separately. And kept a copy of the data without performing standardization on it.

ii. PSD, applying feature selection before modeling, keeping all classes.

The first 6 steps are similar to (*i.)*

7. Prepared the data for training/evaluation by keeping all 5 classes and splitting the data into train and test.

8. Performed feature selection using Logistic

regression L1. Dropped all features that had absolute value coefficient of 0.

9. Preprocess the evaluation data in the same way as training data.

iii. PSD, using two classes and applying feature selection.

Applied the same first 6 steps as in (*i.).*

7. Selected all classes except class 0 and combined all of them into one class which is 1.

8. Split the data into training and testing data.

9. Similar to step *8* in (*ii.*).

10. Similar to step *9* in (*ii.*).

iiii. LSTM, two classes

1. Ensure data balance by aligning the distribution of non-repaired vehicles with that of repaired vehicles in terms of row counts and vehicle numbers for both training and testing datasets.

2. Implemented window class labeling into the data. considering label 1 as the class-label. This combines all classes 1-4 into one class which is 1.

3. Conduct a feature extraction process for each attribute within the dataset, calculating statistics such as the mean, median, standard deviation, and variance. Convert the data from varying sequence lengths to a uniform sequence length of 10-time steps for each entry.

4. Normalize the dataset to ensure uniformity of scale across all features.

5. Address any missing values by substituting them with zeros.

6. Pad the data to achieve a uniform sequence length of 47 readings per vehicle, using a padding value of -1.

7. Reshape the dataset into a structure of (samples, time steps, features). For the validation data, select an equitable number of vehicles from each category (based on the training dataset) and reshape accordingly.

D. Modelling:

i. Power Spectral Density (PSD) one class model:

With help of sklearn, we implemented two OneClassSVM models, one IsolationForest model and one LocalOutlierFactor. Also test manual hyperparameter tuning for both IsolationForest and LocalOutlierFactor.

ii. PSD, applying feature selection before modeling, keeping all classes.

With help of sklearn, we implemented a LogisticRegression, Random Forest and Gradient Bossting models. We used the default parameter for all models.

iii. PSD, using two classes and applying feature selection

with help of sklearn, we implemented LogisticRegression, Random Forest, Gradient Bossting and CatBoost models. We applied GridSearchCV in order to search for the most suitable hyper parameter for the data.

iiii. LSTM, two classes

Construct a Long Short-Term Memory (LSTM) network incorporating masking layers along with L1 and L2 regularization layers to prevent overfitting.

E. Evaluation:

*E*valuated the performance of the models in all experiments using metrics such as accuracy, F1 score, pression and recall. Also AUC.

F. Deployment:

In order to consider deploying any model/s from the experiments the we should obtain at least 90% accuracy and 90% recall in all classes.

IV. RESULTS AND ANALYSIS

We will present only the best model from each experiment and the rest of the results will be provided in the appendix.

```
Contamination: 0.5, n_estimators: 100
AUC: 0.5662923854848305
Confusion Matrix:
[[ 67  97]
 [ 41 123]]
Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.41      0.49       164
           1       0.56      0.75      0.64       164

    accuracy                           0.58       328
   macro avg       0.59      0.58      0.57       328
weighted avg       0.59      0.58      0.57       328
```

Image 1: Isolation Forest model had the best result from one-class models experiment.

Gradient Boosting model

	precision	recall	f1-score	support
0	0.98	0.67	0.80	76
1	0.00	0.00	0.00	16
2	0.08	0.14	0.10	14
3	0.22	0.80	0.35	30
4	0.19	0.04	0.07	76
accuracy			0.38	212
macro avg	0.29	0.33	0.26	212
weighted avg	0.46	0.38	0.36	212

Image 2: Gradient Boosting model had the best result from feature extraction experiment, with classes 0-4.

	precision	recall	f1-score	support
0	0.77	0.65	0.70	136
1	0.70	0.81	0.75	136
accuracy			0.73	272
macro avg	0.73	0.73	0.73	272
weighted avg	0.73	0.73	0.73	272

Image 3: Cat Boost model had the best result from two classes experiment, with classes 0-1 which have time window (48-0) time unit to fail.

	precision	recall	f1-score	support
0.0	0.61	0.70	0.65	136
1.0	0.65	0.56	0.60	136
accuracy			0.63	272
macro avg	0.63	0.63	0.63	272
weighted avg	0.63	0.63	0.63	272

AUC: 0.6518706747404843

Image 4: LSTM model result, with classes 0-1 which have time window (48-0) time unit to fail.

V. DISCUSSION

From the results in image 1, we observe that the AUC (Area Under the Curve) value of approximately 0.5 indicates that the model's ability to distinguish between classes is equivalent to random chance. An AUC value of 0.5 implies that the model lacks the discriminative capacity to differentiate between unrepaired (class 0) and repaired (class 1) vehicles. Despite employing hyperparameter tuning to ascertain the optimal settings, which are presented in the image, the model still fails to perform better than a random classifier. This outcome necessitates a thorough investigation into potential issues such as inadequate feature selection, data quality, or the need for more complex modeling techniques. The test results are using the train read-out data.

The results in image 2 indicate that the Gradient Boosting model outperformed the other evaluated models, including random forest and logistic regression. Notably, all three models failed to identify class 1, as evidenced by a score of zero across all performance metrics (precision, recall, and F1-score) for this class. The most accurately predicted classes were 0 and 3. Despite the implementation of feature selection and extraction techniques, the anticipated improvement in model performance was not realized. The results remain suboptimal.

In image 3, the CatBoost model outperformed the other models tested—including random forest, gradient boosting, logistic regression, and a voting ensemble model—in the binary classification task. The model displayed promising performance in distinguishing between repaired (class 1) and unrepaired (class 0) vehicles, with all performance metrics (precision, recall, and F1-score) hovering around 70%. It is noteworthy that this result was achieved by selecting the first 136 unrepaired (class 0) vehicles for the experiment taken from validation data. However, when we modified the selection to include the last 136 vehicles instead, there was a noticeable drop in performance across all metrics. This suggests that some vehicles in the dataset may influence the model's ability to accurately classify them.

In image 4, the LSTM model's performance for binary classification within the 0–48-time unit window before to vehicle failure is documented. The model achieved an AUC value of approximately 0.65, which is considered moderate, indicating some ability to distinguish between the classes, but there is room for improvement.

Despite efforts to fine-tune hyperparameters, we did not significantly enhance the model's predictive accuracy. It's important to note that the training data for non-repaired (class 0) vehicles was distributed to match the repaired (class 1) vehicles in terms of the number of readout rows and vehicle count.

Further investigation to better understand the behavioral distinctions between non-repaired and repaired vehicles and identify factors that can accurately differentiate between them was done. Our observations indicate that the models can detect differences in certain subsets of non-repaired vehicles, yet they fail to maintain this performance across all segments. This inconsistency suggests that the discriminative features may vary within the non-repaired (class 0) vehicle category, or that our current modeling approach may not capture the complexity of the underlying patterns effectively.

We must acknowledge that the test results “in appendix” from the read-out data (training data) showed outstanding performance compared to the validation data results, due to data leakage. This occurred because the data was preprocessed prior to being split into training and test sets, which unintentionally introduced knowledge of the test data into the training process.

We also implemented SMOTE to address data imbalance, rather than down sampling; however, this approach did not yield positive results. In fact, it led to a decrease in performance, the tests showed that the models were overfitted when we used SMOTE.

To address this issue with imbalance between the 5 classes. We combined the minority classes (1-4) into one class to avoid model bias and as well as simplify the training process for the models.

VI. LIMITATION

While conducting this project of developing a predictive model for forecasting imminent failures of heavy-duty trucks we faced several limitations. The limitation that affected our analysis the most is that the data was anonymized, this had an impact on our analysis, in a way that made it hard to understand the aim of each feature. Despite conducting visual and statistical tests to determine the importance of each feature. The data contained missing values, and without knowledge of the features that contained the missing values, we suffered to find if the missing values were due to an error, or it might be normal. The sensors readout sequences for each vehicle were variable in the data. This was a limitation, because in order to use a model such as LSTM we had either use padding or down/over sampling the data. Another limitation that is worth to mention as well is the imbalance in data, in class 0 for instance we had more than 1000 vehicles sensor readouts while in class 1 and 2 we had less than 60 vehicles.

VII. CONCLUSION

In conclusion, our project aimed to train a predictive maintenance model for heavy-duty trucks using the 2024 IDA Challenge dataset, exploring a blend of machine learning and deep learning techniques. Despite the complexity of the operational data and challenges such as data imbalance and anonymization, we applied CRISP-DM framework methodologies to develop predictive models.

Key findings from our study emphasize the critical role of data quality and preparation in predictive maintenance, notably how techniques like SMOTE can influence model outcomes by mitigating class imbalance. Our exploration revealed that while traditional machine learning models like Random Forest showed promised performance, the integration of deep learning models, particularly LSTMs, faced challenges due to data limitations.

VIII. REFERENCES

[1] (2023, October 24). Automotive Predictive Maintenance: How Does it Work? Encora. https://www.encora.com/insights/what-is-automotive-predictive-maintenance

[2] (2024, January 1). Developing an Effective Predictive Model for Imminent Component X Failures in Heavy-Duty Scania Trucks. IDA 2024. https://ida2024.blogs.dsv.su.se/files/2024/01/2024_IDA_challenge_v2.pdf

[3] L. P. Silvestrin, M. Hoogendoorn and G. Koole, "A Comparative Study of State-of-the-Art Machine Learning Algorithms for Predictive Maintenance," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 760-767, doi: 10.1109/SSCI44817.2019.9003044.

[4] S. Sridhar and S. Sanagavarapu, "Handling Data Imbalance in Predictive Maintenance for Machines using SMOTE-based Oversampling," 2021 13th International Conference on Computational Intelligence and Communication Networks (CICN), Lima, Peru, 2021, pp. 44-49, doi: 10.1109/CICN51697.2021.9574668.

[5] D. B. Sharma, Sripradha, Nikita, A. Kodipalli, T. Rao and R. B R, "Machine Predictive Maintenance Classification Using Machine Learning," 2023 International Conference on Computational Intelligence for Information, Security and Communication Applications (CIISCA), Bengaluru, India, 2023, pp. 308-313, doi: 10.1109/CIISCA59740.2023.00066.

IX. APPENDIX

Result for one-class models, using the train read-out data.

OneClassSVM Classification Report:

	precision	recall	f1-score	support
-1	0.46	0.45	0.45	164
1	0.46	0.48	0.47	164
accuracy			0.46	328
macro avg	0.46	0.46	0.46	328
weighted avg	0.46	0.46	0.46	328

OneClassSVM Classification Report std:

	precision	recall	f1-score	support
-1	0.33	0.05	0.09	164
1	0.49	0.89	0.63	164
accuracy			0.47	328
macro avg	0.41	0.47	0.36	328
weighted avg	0.41	0.47	0.36	328

Image 1: OneClassSVM result from one-class models experiment standardize data and non-standardize.

Contamination: 0.5, n_estimators: 100
AUC: 0.44631171921475316
Confusion Matrix:
[[74 90]
 [92 72]]
Classification Report:

	precision	recall	f1-score	support
0	0.45	0.45	0.45	164
1	0.44	0.44	0.44	164
accuracy			0.45	328
macro avg	0.45	0.45	0.45	328
weighted avg	0.45	0.45	0.45	328

Image 2: Depicts the best result achieved by the Local Outlier Factor (LOF) model, following hyperparameter tuning, along with the calculation of the AUC (Area Under the Curve).

Contamination: 0.5, n_estimators: 100
AUC: 0.5662923854848305
Confusion Matrix:
[[67 97]
 [41 123]]
Classification Report:

	precision	recall	f1-score	support
0	0.62	0.41	0.49	164
1	0.56	0.75	0.64	164
accuracy			0.58	328
macro avg	0.59	0.58	0.57	328
weighted avg	0.59	0.58	0.57	328

Image 3: Depicts the best result achieved by the Isolation Forest model, following hyperparameter tuning, along with the calculation of the AUC (Area Under the Curve).

The results of the feature extraction experiment test results are using the train read-out data.

	precision	recall	f1-score	support
0	0.50	0.69	0.58	32
1	0.17	0.12	0.14	8
2	0.09	0.07	0.08	14
3	0.36	0.25	0.30	32
4	0.47	0.52	0.49	33
accuracy			0.41	119
macro avg	0.32	0.33	0.32	119
weighted avg	0.39	0.41	0.39	119

Image 4: logistic regression model result

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.79	0.81	0.80	32
1	0.27	0.38	0.32	8
2	0.40	0.29	0.33	14
3	0.52	0.50	0.51	32
4	0.71	0.73	0.72	33
accuracy			0.61	119
macro avg	0.54	0.54	0.53	119
weighted avg	0.61	0.61	0.61	119

Image 5: Random Forest model result

Gradient Boosting Classification Report:

	precision	recall	f1-score	support
0	0.89	0.97	0.93	32
1	0.33	0.25	0.29	8
2	0.18	0.14	0.16	14
3	0.65	0.62	0.63	32
4	0.81	0.88	0.84	33
accuracy			0.71	119
macro avg	0.57	0.57	0.57	119
weighted avg	0.68	0.71	0.69	119

Image 6: Gradient Boosting model result

The results of the feature extraction experiment test results are using the evaluation read-out data.

Logistic Regression model

	precision	recall	f1-score	support
0	0.50	0.17	0.25	76
1	0.00	0.00	0.00	16
2	0.07	0.21	0.11	14
3	0.18	0.33	0.24	30
4	0.33	0.34	0.33	76
accuracy			0.25	212
macro avg	0.22	0.21	0.19	212
weighted avg	0.33	0.25	0.25	212

Image 7: logistic regression model result

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.79	0.59	0.68	76
1	0.00	0.00	0.00	16
2	0.04	0.07	0.05	14
3	0.15	0.50	0.24	30
4	0.50	0.16	0.24	76
accuracy			0.34	212
macro avg	0.30	0.26	0.24	212
weighted avg	0.49	0.34	0.37	212

Image 8: Random Forest model result

Gradient Boosting model

	precision	recall	f1-score	support
0	0.98	0.67	0.80	76
1	0.00	0.00	0.00	16
2	0.08	0.14	0.10	14
3	0.22	0.80	0.35	30
4	0.19	0.04	0.07	76
accuracy			0.38	212
macro avg	0.29	0.33	0.26	212
weighted avg	0.46	0.38	0.36	212

Image 9: Gradient Boosting model result

The results of the Two class model experiment test results are using the train read-out data.

	precision	recall	f1-score	support
0	0.69	0.76	0.72	33
1	0.73	0.67	0.70	33
accuracy			0.71	66
macro avg	0.71	0.71	0.71	66
weighted avg	0.71	0.71	0.71	66

Image 10: logistic regression model result

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.78	0.76	0.77	33
1	0.76	0.79	0.78	33
accuracy			0.77	66
macro avg	0.77	0.77	0.77	66
weighted avg	0.77	0.77	0.77	66

Image 11: Random Forest model result

Gradient Boosting Classification Report:

	precision	recall	f1-score	support
0	0.80	0.73	0.76	33
1	0.75	0.82	0.78	33
accuracy			0.77	66
macro avg	0.78	0.77	0.77	66
weighted avg	0.78	0.77	0.77	66

Image 12: Gradient Boosting model result

CatBoost Classification Report:

	precision	recall	f1-score	support
0	0.81	0.79	0.80	33
1	0.79	0.82	0.81	33
accuracy			0.80	66
macro avg	0.80	0.80	0.80	66
weighted avg	0.80	0.80	0.80	66

Image 13: Cat Boost model result

The results of Two class model experiment test results are using the evaluation read-out data.

logistic model

	precision	recall	f1-score	support
0	0.64	0.26	0.37	136
1	0.54	0.85	0.66	136
accuracy			0.56	272
macro avg	0.59	0.56	0.52	272
weighted avg	0.59	0.56	0.52	272

Image 14: logistic regression model result

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.74	0.66	0.70	136
1	0.69	0.76	0.73	136
accuracy			0.71	272
macro avg	0.72	0.71	0.71	272
weighted avg	0.72	0.71	0.71	272

Image 15: Random Forest model result

Gradient Boosting model

	precision	recall	f1-score	support
0	0.74	0.66	0.70	136
1	0.69	0.76	0.73	136
accuracy			0.71	272
macro avg	0.72	0.71	0.71	272
weighted avg	0.72	0.71	0.71	272

Image 16: Gradient Boosting model result

Cat Boost model

	precision	recall	f1-score	support
0	0.73	0.63	0.68	136
1	0.68	0.76	0.72	136
accuracy			0.70	272
macro avg	0.70	0.70	0.70	272
weighted avg	0.70	0.70	0.70	272

Image 17: Cat Boost model result

Voting Classifier Classification Report:

	precision	recall	f1-score	support
0	0.73	0.65	0.68	136
1	0.68	0.76	0.72	136
accuracy			0.70	272
macro avg	0.70	0.70	0.70	272
weighted avg	0.70	0.70	0.70	272

Image 18: Voting model result.

	precision	recall	f1-score	support
0.0	0.59	0.74	0.65	1137
1.0	0.64	0.48	0.55	1134
accuracy			0.61	2271
macro avg	0.61	0.61	0.60	2271
weighted avg	0.61	0.61	0.60	2271

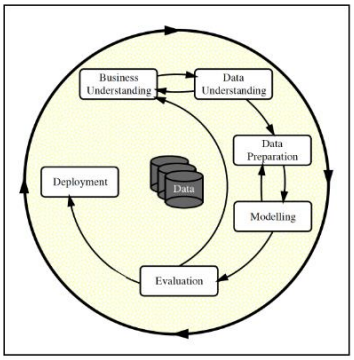
AUC: 0.6503601016940213

Image 19: LSTM model result using validation data that the model used in the fitting step.

	precision	recall	f1-score	support
Class 0	0.60	0.53	0.56	136
Class 1	0.58	0.65	0.61	136
accuracy			0.59	272
macro avg	0.59	0.59	0.59	272
weighted avg	0.59	0.59	0.59	272

AUC: 0.6023464532871973

Image 20: LSTM model result using validation read-out data and hyperparameter tuning.



Appendix1: Phases of the CRISP-DM reference model. Reprinted from "CRISP-DM: Towards a standard process model for data mining" by R. Wirth and J. Hipp, 2000, In Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining (Vol. 1, pp. 29-39).