Improving the Resiliency of Embedded Networks in Heavy Vehicles - Towards Fault Tolerance

Chandrima Ghatak
Colorado State University
Fort Collins, United States of America
chandrima.ghatak@colostate.edu

Saira Jabeen
Colorado State University
Fort Collins, United States of America
saira.jabeen@colostate.edu

Hossein Shirazi
San Diego State University
San Diego, United States of America
hshirazi@sdsu.edu

Indrakshi Ray
Colorado State University
Fort Collins, United States of America
indrakshi.ray@colostate.edu

ABSTRACT

In both military and civilian sectors, Medium and Heavy Duty (MHD) vehicles form a critical part of infrastructure, logistics, and operations. Modern MHD vehicles are equipped with embedded computers that heavily rely on sensor data for maintaining operations. This data is often exchanged using standard protocols. The Society of Automotive Engineering's (SAE) J1939 Protocol on top of the Controller Area Network (CAN) is the common standard for data exchange in modern MHD vehicles. The resilience of this communication network is pivotal for the operational feasibility of MHD vehicles, especially in hostile environments where sensors may be compromised as a result of cyber attacks or kinetic malfunctions. This paper proposes the use of predictive machine learning algorithms to forecast accurate sensor readings when a sensor system becomes unavailable, whether due to physical or cyber-attacks. Utilizing real-world data from a Class 6 Kenworth T270 truck as a case study, we explore and evaluate the effectiveness of three different machine-learning methods in predicting missing sensor data. Our findings indicate that the machine learning models are capable of nearly accurate predictions, which can prevent the vehicle from entering into engine protection or limp mode. This not only maintains the vehicle's operational status for extended periods but also contributes to enhancing the resilience of networked cyber-physical systems.

KEYWORDS

Resilient systems, Fault Tolerance, Heavy vehicle systems, SAE J1939 networking architectures, Predictive algorithms, Networked cyber-physical system

1 INTRODUCTION

Medium and Heavy Duty (MHD) vehicles are a part of critical modern infrastructure serving as the backbone of various commercial and governmental activities. Modern MHD vehicles are equipped with multiple embedded computers or Electronic Controller Units (ECUs) which control most functions. These ECUs are increasingly reliant on complex sensor systems for maintaining operations through the exchange of sensor data over communication lines. In the US the standard choice of communication among ECUs is the Society of Automotive Engineering's (SAE) J1939 [18] based on the Controller Area Networks (CAN) [3] protocol. While these standard communication protocols offer remarkable fault tolerance, they fall short in risk mitigation strategies, particularly when sensor systems malfunction or when the sensor system faces a cyber-attack. In such scenarios compromised sensor data could render a vehicle non-operational—forcing it into a 'limp mode' or causing a complete shutdown.

Traditional fault-tolerance mechanisms have limitations in adaptability and predictive accuracy. They often require manual updates or intervention and can not adapt to new types of sensor failures or cyber threats in real time. Machine learning offers a dynamic, automated alternative. Specifically, machine learning models can be trained to recognize complex patterns in sensor data, predict future values, and detect anomalies, making them uniquely suited to the problem of ensuring resilience in MHD vehicles.

Against this backdrop, the central research question we explore is: *Can a single, generalized machine learning model deliver real-time, robust predictions for compromised or missing sensor values in MHD vehicles?*

The overarching goal of this study is to develop a machine learning-based solution that serves as a resilient countermeasure to sensor malfunctions in MHD vehicles. The ambition is to construct a unified neural network model that is both robust and adaptable across various real-world operational scenarios. A unique aspect of this approach is the use of a single generalized neural network model that is trained on diverse sensor data. This not only ensures broader applicability but also reduces the system's complexity by eliminating the need to design separate neural networks for each sensor type. To validate our approach, we conduct experiments using three machine learning algorithms chosen for their theoretical merits: Dense Binary Transformer (DBT) [11, 24], Sparse Binary Transformer (SBT) [7], and Long Short-Term Memory (LSTM) [8]. Each algorithm was selected for its unique advantages in dealing with time-series data, computational efficiency, and predictive accuracy. These attributes are subjected to empirical validation through a series of experiments. Initial analysis suggests that the DBT model consistently outperforms its counterparts in a majority of scenarios. However, it is worth noting that LSTM and SBT models also demonstrated commendable efficacy, trailing not far behind, thereby underlining the promise of machine learning approaches in enhancing MHD vehicle robustness. Our experiments yielded promising results, suggesting that machine learning can indeed provide robust, real-time solutions for the complex problem of sensor malfunction in MHD vehicles.

Through this comprehensive examination, our research demonstrates the feasibility of a robust machine-learning framework that can substantially enhance the resilience and adaptability of sensor systems in MHD vehicles. This is a critical advancement in mitigating the risks associated with mechanical failures and cyber vulnerabilities in the increasingly interconnected and digital landscape of modern vehicular technologies.

To elaborate on these aspects, the remainder of this paper is structured into key sections. Section 2 provides background on MHD vehicle communication protocols and presents related work. Section 3 explains the unified machine learning model we propose, detailing its advantages and unique contributions. This is followed by section 4 which outlines our experimentation design and the resultant observations, thereby highlighting the strengths and weaknesses of each algorithm in the given application context. This is followed by section 5 which summarizes the findings of our experiments. The paper concludes with a discussion of the broader implications, limitations, and avenues for future research.

2 BACKGROUND

In this section, we will provide an overview of the message format employed in the widely used SAE J1939 protocol for data communication among Electronic Control Units (ECUs) in heavy vehicles as well as discuss the research that has been done previously.

SAE J1939 protocol is a high-level protocol that is used for communication among Electronic Control Units (ECUs) in Medium and Heavy Duty (MHD) vehicles. Utilizing the Controller Area Network (CAN) bus [3], it supports the transmission of Protocol Data Units (PDUs) as the primary message packets. Each PDU is composed of a 29-bit extended identifier field and a data field that accommodates 0 to 64 bytes of data. This identifier contains multiple subfields like Priority, Extended Data Page (EDP), Data Page (DP), PDU Format (PF), PDU Specific (PS), Source Address (SA), and Destination Address (DA), which collectively provide the contextual layer for the encapsulated data. A special aspect is the Parameter Group Number (PGN), which is formed by combining the PF and PS fields and serves as a unique identifier for the type of data in the PDU. The data field, in turn, contains Suspect Parameter Numbers (SPNs), which represent various types of sensor readings [18]. For the purpose of this paper, SPN and sensor data will be used interchangeably.

Security challenges in vehicular networks have been well-studied. Initial findings have shown vulnerabilities in CAN protocols that passenger vehicles use [2]. Attacks like message injection and even remote exploitation have been identified [6, 9, 23]. Additionally, attacks on sensor systems have also increased over the years [10, 16, 17]. Similar vulnerabilities have also been found to exist in MHD vehicles [15, 22], accentuated by the fact that J1939 specifications are publicly available, posing an increased risk of targeted attacks [4, 5, 12–14].

Countermeasures proposed include intrusion detection systems and cryptographic solutions [15, 22]. However, cryptographic solutions may be too resource-intensive for practical use in MHD vehicles [1].

Machine learning has emerged as a promising approach for mitigating these vulnerabilities. Shirazi et al. successfully utilized machine learning models to reconstruct compromised sensor data [19, 20]. They primarily employed Long Short-Term Memory (LSTM) autoencoders, suggesting that compromised or missing values can be accurately replaced by using other available sensor readings. However, their methodology requires a distinct neural network for each sensor, which may not be practical in a resource-constrained environment [19].

Previous efforts have shown promise but are limited by their need for multiple neural networks to predict specific sensor values. Our work seeks to surmount this limitation by proposing a unified approach. Employing a single neural network model, we aim to provide a robust solution that can predict any missing or compromised sensor data with high accuracy.

3 PROPOSED APPROACH

In this section, we discuss the different methods we used for our experiments, the reason for using these methods, and the special features of these methods that will be beneficial for our goals.

Our primary goal is to develop a unified neural network model capable of predicting one or multiple missing or compromised sensor values in real-time in MHD vehicles. This centralized approach aims to enhance the system's robustness and adaptability.

We are looking for the following factors for our research.

- **Generalization:** By employing a single generalized network trained on a diverse sensor dataset, our approach offers broad applicability. This eliminates the constraints of designing separate networks for each sensor or requiring a fixed sensor configuration.
- **Real-time Prediction:** The generalized model is designed to deliver real-time predictions. This ensures the smooth operation of other sensors and systems that are dependent on the missing or compromised sensor data.
- **Reduced System Complexity:** Using one model for all sensors simplifies the system architecture, reducing points of failure and the computational overhead of managing multiple models.

Dense Binary Transformer (DBT) The Dense Binary Transformer (DBT) serves as an advanced variant of the well-established Transformer model, which is an architecture originally designed for natural language processing tasks. The Transformer architecture consists of two primary components: an encoder and a decoder. These are responsible for transforming the input data into a format that can be utilized for tasks like classification or prediction. A noteworthy feature of the Transformer model is the self-attention mechanism, which allows the model to weigh different parts of the input data based on their relevance to the task at hand. DBT employs a specialized form of the self-attention mechanism known as ProbSparse Self-Attention. Unlike traditional self-attention, which involves each input unit (commonly referred to as a 'key') interacting with all other units (referred to as 'queries'), the ProbSparse mechanism limits these interactions. Specifically, each key is permitted to interact only with a subset of queries, effectively reducing computational time and memory usage. This is particularly advantageous when the model has to handle large and complex datasets.

Window Size	Training with all data			Training with 5% of missing data		
	DBT	SBT	LSTM	DBT	SBT	LSTM
10	**0.2446**	**0.5901**	**0.4810**	**0.2432**	**0.4969**	**0.4952**
50	0.2767	0.6391	0.5710	0.2661	0.5886	0.6721
100	0.3174	0.6784	0.6312	0.2638	0.6453	0.7611
200	0.4850	0.7537	0.7005	0.4209	0.6892	0.9023

Table 1: Mean Squared Error on Test Data

In the context of machine learning, an encoder is a component of a neural network responsible for transforming raw input data into a condensed, machine-interpretable form. DBT optimizes the conventional Transformer encoder for increased efficiency in processing long sequences of data. To achieve this, it integrates Convolutional 1D layers—which are generally employed for spatial feature extraction in image data—and MaxPooling layers that reduce data dimensions while preserving essential features. These layers work in tandem with traditional self-attention blocks to extract vital features from large datasets.DBT is specially engineered to capture intricate patterns in large and complex data structures, making it highly effective for tasks requiring predictive accuracy. In particular, it excels in real-time sensor value prediction. The architecture of DBT is tailored to optimize both computational efficiency and the capability to recognize complex patterns, positioning it as an ideal choice for performance-critical applications.

Sparse Binary Transformer (SBT) The Sparse Binary Transformer (SBT) is another derivative of the Transformer model, which, similar to DBT, addresses specific challenges in computational efficiency and scalability. As discussed under DBT, the Transformer architecture is bifurcated into an encoder and a decoder, with self-attention mechanisms playing a crucial role in data transformation and task-specific learning. SBT distinguishes itself by adopting an even more computationally efficient self-attention mechanism. The mechanism is optimized to reduce the amount of computational resources required for processing, making it suitable for applications that need real-time response, such as sensor value prediction. The encoder in SBT is engineered to minimize memory usage and computational time. It employs a selection of specialized layers designed to reduce the dimensionality of the input data effectively while maintaining essential features. The layers include variants of traditional self-attention blocks but are optimized to be computationally less demanding. The SBT model is ideal for scenarios that require real-time sensor value prediction. Its architecture focuses on achieving high predictive performance while being computationally efficient, a crucial factor for ensuring optimal system performance in real-time applications.

Long Short-Term Memory (LSTM) Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) architecture, specifically designed for sequence prediction problems. RNNs are neural networks where connections between nodes form a directed graph along a temporal sequence. This allows them to maintain a 'memory' of previous inputs, making them well-suited for tasks involving sequential data, such as time series prediction. LSTM networks include memory cells that allow them to store and recall information over long sequences effectively. Unlike standard RNNs, which often suffer from the 'vanishing gradient' problem, LSTMs are capable of learning long-term dependencies in the data. The LSTM architecture described employs a two-stage process—an encoder that processes the input sequence and captures its information in a 'context vector', and a decoder that generates the output sequence based on this context vector. The architecture includes special layers like 'RepeatVector' for replicating the context vector and 'TimeDistributed' for generating the output sequence. LSTM is particularly advantageous for making inferences from sensor data that exhibit temporal correlations. It is adept at capturing long-range dependencies in time-series data, making it a potentially effective algorithm for applications that require high predictive accuracy over extended periods.

The selection criteria for these models stem from their respective theoretical advantages in both predictive accuracy and computational efficiency, which will be empirically validated through a battery of experiments designed to rigorously evaluate their performance under various real-world conditions.

4 EXPERIMENTS AND RESULTS

In this section, we discuss how we prepared the data for our experiments, our testing methodology, and the results from our different experiments.

4.1 Data Preparation

We collected data from a 2014 Kenworth T270 research truck during a 2018 cross-country trip from Fort Collins to Detroit [21]. The data, initially in 'candump' format, consisted of time-stamped ASCII values with CAN identifiers and data fields. Using the SAE-J1939 standard, we isolated relevant PGNs and decoded them to actual engineering values. Out of all the sensor values on the CANbus, 52 sensors had non-static, measured values that can affect the training and testing process positively, contributing to predicting missing sensor values effectively. We used these 52 SPNs for our experiments. To address different periodicities in CAN messages, we sampled the most recent sensor data at 500 millisecond intervals, creating a time-series dataset. The dataset was then normalized to scale the values between 0 and 1 for training and testing.

Our experiments revolved around optimizing the performance of the algorithms we used. We extensively evaluated these algorithms by training them under varied data conditions, specifically focusing

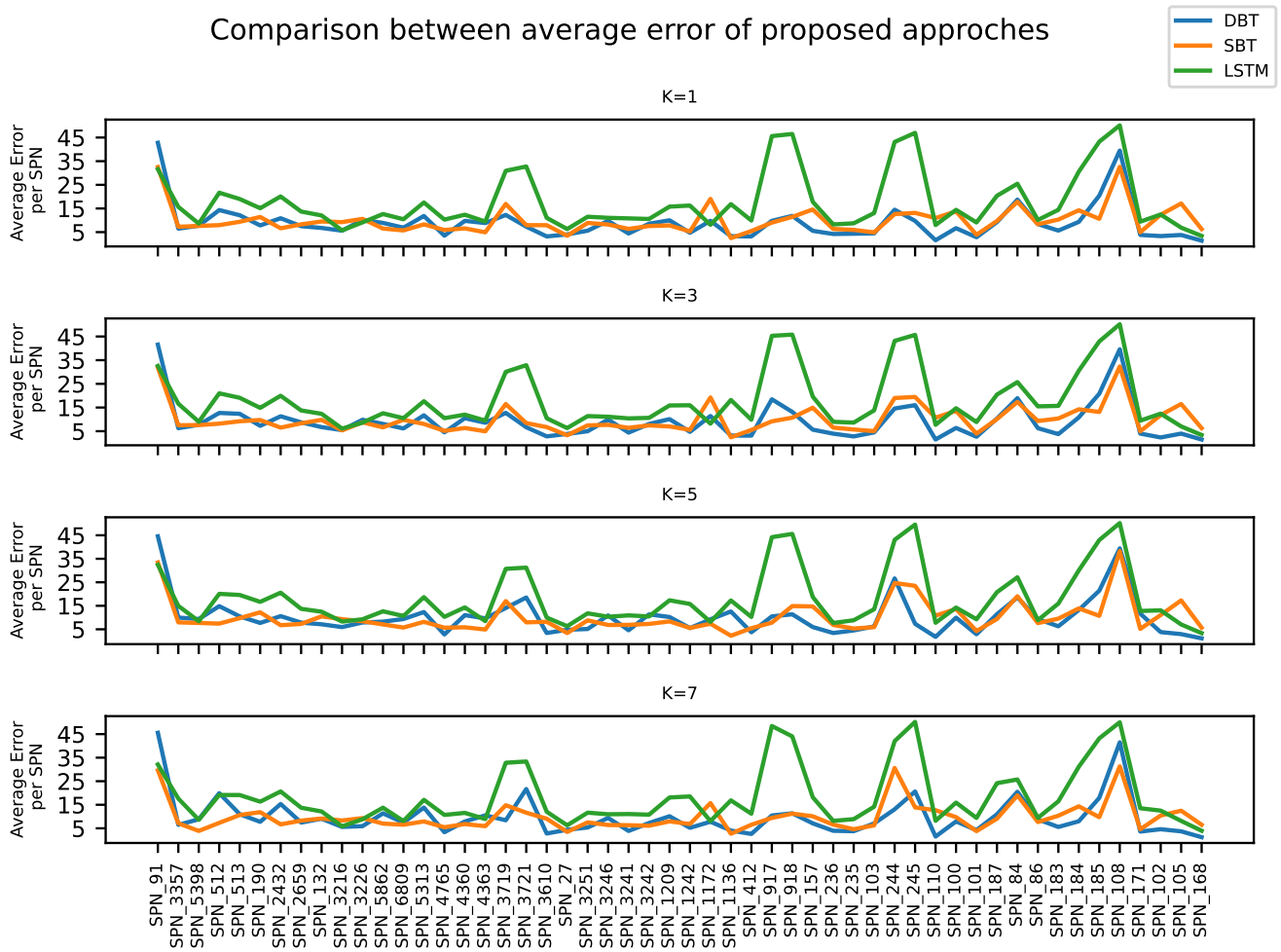

Figure 1: Error comparison at 4 different values of k, where k represents the number of missing SPNs

on the impact of missing values – a practical challenge in real-world scenarios. For quantifying the accuracy of the results during testing, the Average Percentage Error (*err*) served as our metric, computed as:

$$err = \frac{1}{n}\sum_{i=1}^{n}\frac{|e_i|}{z} \times 100, \tag{1}$$

where $|e_i|$ represents the absolute error between actual and predicted values for a data point, z is the SPN's range, and n encapsulates all dataset points for the specific SPN. This error measurement provided a normalized perspective across varied SPNs and their distinct ranges. The experiments were bifurcated into two main training strategies to emulate both ideal and realistic situations. We adopted two distinct training strategies to better understand model performance under different data conditions:

Full-data Training. In this approach, we trained our models using the complete dataset, ensuring that all original values from the sensors were incorporated. This served as our baseline to evaluate the maximum potential performance of the models when given access to all available data.

Training with Missing Values. Recognizing that in real-world scenarios, sensor data might be sporadically unavailable due to outages or malfunctions, we introduced a second training strategy. Here, we artificially induced missing values in the training dataset to simulate the scenario of sensors being unavailable during training. This was done to ascertain how well the models can learn and generalize in the face of incomplete data.

The dataset, comprising 16,000 instances, is divided into training and testing sets with proportions of 75% and 25%, respectively. Both transformer models are trained for 100 epochs using an NVIDIA TITAN V with 12GB of memory.

Table 1 displays the mean squared error (MSE) calculated using two distinct training approaches across various window sizes. The MSE values indicate that all DBT, SBT, and LSTM perform optimally when the window size is set to 10, suggesting that larger window sizes do not significantly affect performance. Additionally,

the results demonstrate that training models on missing data enable them to learn missing data behavior effectively.

A randomized model search algorithm is employed to find the best hyperparameters for the models, DBT, SBT, and LSTM. A number of hyperparameters are searched, including optimizer, learning rate, batch size, and layer/batch normalization. Following are the architectural details of the best-searched models. Interestingly, best-performing architectures were found to be similar in both DBT and SBT:

- **Positional encoding size of 64 units:** This specifies the size of the positional encoding vector, which helps the model understand the order of the data.
- **Four encoder layers:** The encoder processes the input data and converts it into a form that the decoder can understand. Four layers were found to be optimal for this.
- **Four self-attention heads in each encoder layer:** These are used to focus on different parts of the input sequence when processing it, essentially allowing the model to pay 'attention' to different aspects of the data.
- **Hidden layers with the size of 128 units:** This is the number of neurons in the hidden layers of the neural network, crucial for the learning process.

For LSTM, the best-performing architecture consisted of:

- **Input layer with shape of $(n_{\text{past}}, n_{\text{features}})$:** The input layer is where the network starts, it takes in an input shape corresponding to the number of past observations and features in each observation.
- **First encoder LSTM layer with node units, returning sequences and states:** The first LSTM layer processes the input data and returns not just the final output, but also an internal state for each step in the input sequence.
- **Second encoder LSTM layer with node units, returning states:** This second LSTM layer processes the sequence further and returns the internal state.
- **A `RepeatVector` layer:** This layer duplicates the final output of the encoder, effectively focusing the decoder's attention on the most important parts of the input sequence.
- **First and Second decoder LSTM layers:** These layers are responsible for generating the output sequence, using the internal states returned by the encoder layers for initialization.
- **`TimeDistributed` wrapper around a `Dense` layer with n_{features} units for output:** This distributes the dense layer for each time step in the output sequence, essentially creating one dense layer per time step.

4.2 Single model to predict *all* SPNs with one or more missing SPNs at a time

In this subsection, we evaluate the performance of different predictive models including LSTM, SBT, and DBT in predicting sensor values under various test cases.

4.2.1 Test Case 1: Missing one or more SPNs. The performance of the prediction models varied based on the chosen algorithm, the nature of the training data (whether it was original or had simulated missing values), and the specific SPN being predicted. The disparities in results across different SPNs underscore the importance of algorithm selection and understanding the nuances of training data.

In this case, all of these data points are missing simultaneously, along with all their previous values within a given window. Figure 1 demonstrates our first test case where k SPNs are missing. For every k value, $k-1$ SPNs are randomly chosen from a set of 51 SPNs, and the kth SPN is the one being predicted. The plot reveals that the prediction error for LSTM is consistently the highest among the three methods for all examined values of k. Conversely, DBT demonstrates the lowest error overall, suggesting that LSTM requires the most SPN values to make accurate predictions, whereas transformer models can learn from key features. It is also noteworthy that increasing the value of k does not notably affect the average error for most SPNs.

4.2.2 Test Case 2: Missing SPN as well as its correlated SPNs. Correlated SPNs are identified using Pearson Correlation having a correlation value greater than 0.5. Figure 2 illustrates the comparison of performances among LSTM, SBT, and DBT, all with a window size of 10. Error is highest when SPNs correlated to target SPN are missing which indicates that the transformer model is able to learn the underlying dependence between practically related SPNs. However, the LSTM model exhibits the lowest error among all methods. That rationale is understandable, as LSTM models tend to focus on learning past temporal patterns rather than inter-sensor dependencies. In conclusion, the outcomes indicate that both transformer and LSTM models possess their own distinct advantages.

Figure 3 shows the average prediction error of DBT for each SPN on both test cases: one with 1, 3, 5, and 7 randomly selected missing SPNs, and the other with missing correlated SPNs from the test data. Figure 3 validates our previous findings that as the number of missing SPNs increases, average prediction error also increases, but not significantly. Error is highest when SPNs correlated to target SPN are missing which indicates that the transformer model is able to learn the underlying dependence between practically related SPNs.

4.3 Single model to predict multiple future steps for one SPN missing at a time

To continue our experiments, we tried to observe how the algorithms performed when trying to predict more than one missing value in the future. For this purpose, we performed several more sets of experiments with each of the algorithms and compared their results.

In evaluating the efficacy of the three machine learning algorithms for predicting missing SPN values for more than one step in the future, we conducted two types of experiments on each algorithm for n steps in future prediction.

4.3.1 Test Case 1: One-shot Method. The 'one-shot' prediction method aims to forecast the value of the missing sensor at a specific future time step n directly. In this method, the model is trained to take the values of the remaining 50 sensors as input and output the value of the missing sensor exactly at the time step n.

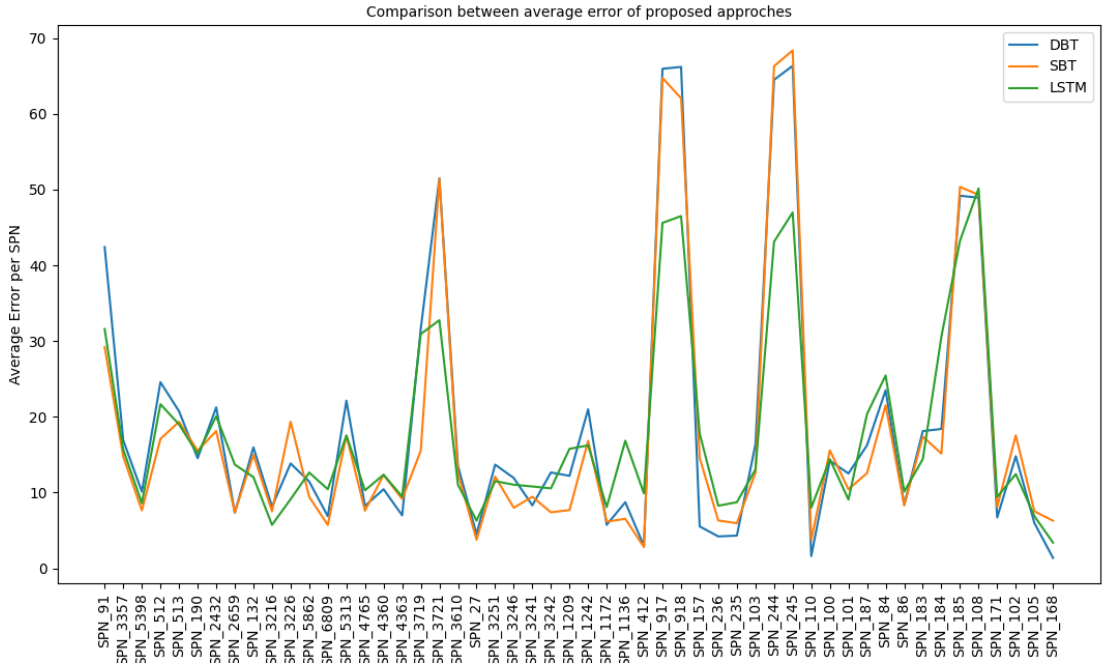

Figure 2: Error comparison between proposed approaches when correlated SPNs are missing

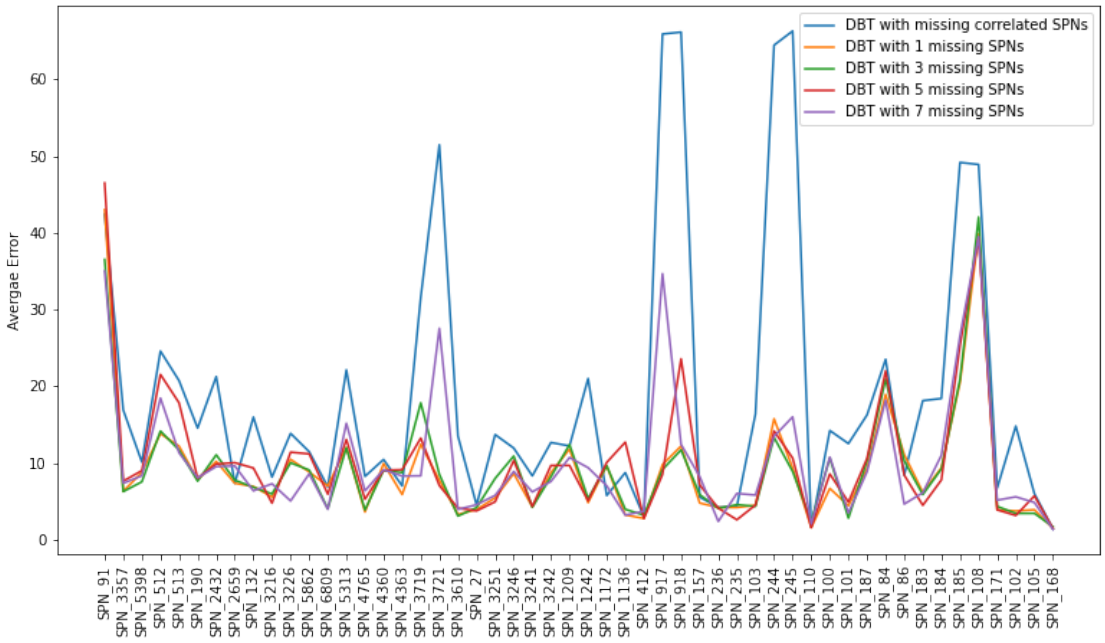

Figure 3: Average prediction error of DBT per SPN

Model	Average Errors for n-steps in future prediction (in %)								
	1	3	5	10	15	20	25	50	100
DBT	4.90	4.45	4.86	5.20	5.41	5.36	7.35	9.43	11.88
SBT	6.90	6.78	6.81	7.31	7.19	7.67	8.62	10.93	14.22
LSTM	5.66	5.45	5.56	5.86	5.93	7.62	9.32	12.93	15.86

Table 2: Average Errors for n-steps in future prediction (in %) for One-shot Method

Model	Average Errors for n-steps in future prediction (in %)								
	1	3	5	10	15	20	25	50	100
DBT	4.90	3.73	3.55	4.01	4.66	4.70	6.04	8.18	10.33
SBT	6.90	5.97	5.40	6.05	5.91	6.27	7.04	9.55	13.17
LSTM	5.66	4.21	4.79	4.34	4.57	6.11	8.13	11.50	14.31

Table 3: Average Errors for n-steps in future prediction (in %) for Recursive Feeding Method

One of its key advantages is computational efficiency. The model directly predicts the missing sensor value at the n-th future time step, thereby bypassing the need to calculate any intermediate values. This often results in faster prediction times. Additionally, this method minimizes the risk of error propagation that can occur when forecasts are built upon previous forecasts, offering potentially more accurate results for the specific future time step of interest.

In the One-shot Method, we evaluated the prediction errors of DBT, SBT, and LSTM for various future time steps ranging from 1 to 100. As observed in Table 2, DBT consistently demonstrates the lowest prediction errors across most future steps, highlighting its efficiency and accuracy in this approach. On the other hand, LSTM and SBT errors are relatively higher, particularly as the future time step increases. Specifically, LSTM exhibits the highest errors for long-term predictions at steps 50 and 100. These findings further solidify the One-shot Method's advantage in computational efficiency, while also emphasizing the model-specific trade-offs in prediction accuracy.

4.3.2 Test Case 2: Recursive Feeding Method. In the recursive feeding method, the model initially predicts the missing sensor value one step ahead (i.e., n=1). This predicted value is then fed back into the model along with the other 50 sensor values to predict the value for the next step. This process is recursively repeated to generate a prediction for the missing sensor at the desired n-th step in the future.

The Recursive Feeding Method brings its own set of merits, notably its simplicity and flexibility. A single model can be used to generate predictions for multiple future steps, making it easier to manage and potentially less computationally expensive to train. Moreover, this approach provides intermediate forecasts for all steps leading up to the target n-th step. This can be particularly valuable when the intermediate states are of interest or when one wishes to understand the progression of sensor values over time.

In the Recursive Feeding Method, we assess the errors associated with DBT, SBT, and LSTM across a range of future time steps, from 1 to 100. As shown in Table 3, DBT again fares the best, with comparatively lower prediction errors at almost all future steps. LSTM, however, displays a notable improvement over its One-shot counterpart, especially at short-to-medium future steps. SBT remains consistently higher in error but shows some improvement in long-term predictions. The advantage of the Recursive Feeding Method lies in its flexibility to predict multiple future steps using a single model, along with the added benefit of providing intermediate states, which could be valuable for certain applications.

5 DISCUSSION

Two distinct types of experiments were executed to address the complexities involved in predicting sensor data in MHD vehicles comprehensively. The first focused on the model's ability to handle missing SPNs under various conditions, aiming to test resilience to real-world uncertainties. The second evaluated the models' predictive accuracy for multiple future time steps, emphasizing their utility in long-term forecasting.

- **DBT** performed well in both experiments, showcasing its all-around capabilities in terms of accuracy and efficiency.
- **LSTM** excelled in scenarios where understanding temporal patterns was vital, specifically in the Recursive Feeding Method.
- **SBT**, while generally trailing in accuracy, demonstrated potential for long-term predictions and showed improvements when handling missing values, making it a viable option in specific scenarios.

The two experiments, when viewed collectively, reveal a layered understanding of each model's strengths and weaknesses. DBT emerges as a strong candidate for most scenarios, while LSTM and SBT each have their unique niches where they can be advantageous.

6 CONCLUSION

This study represents a substantive contribution to the field of sensor data prediction in MHD vehicles, particularly in addressing missing sensor data. Through the rigorous evaluation of three distinct machine learning algorithms, we have demonstrated that a specifically tailored neural network architecture possesses significant potential for accurately filling both single and multiple sensor data gaps. The findings indicate a notable advancement in

this domain and point towards the capability of such machine learning models to contribute meaningfully to vehicle performance and safety.

While the current research offers conclusive insights, the application of the findings presents several promising avenues for future exploration:

(1) **Real-world Model Integration:** Future work could focus on implementing the trained models within operational MHD vehicles, such as a 2014 Kenworth T270, to assess real-world applicability and effectiveness.
(2) **Malfunction Simulations:** An intriguing extension of this work would involve real-time, on-road experiments designed to simulate sensor malfunctions. Such simulations can further validate the real-time responsiveness and fault tolerance of the models
(3) **System Resilience Evaluation:** An important consideration for future investigations would be to reintroduce the model-predicted values back into a vehicle's control network. This would allow for a holistic assessment of the impact on operational resilience, augmenting the existing scope of this research.

In closing, the study underscores the potential of machine learning techniques in enhancing the cyber-resilience of MHD vehicles. These findings provide a robust foundation for future work in this important area.

ACKNOWLEDGMENTS

This work was supported in part by funding from NSF under the Award Number ATD 2123761, CNS 1822118, ARL, Statnett, AMI, NewPush, and Cyber Risk Research.

REFERENCES

[1] Emad Aliwa, Omer Rana, Charith Perera, and Peter Burnap. 2021. Cyberattacks and Countermeasures for In-Vehicle Networks. *ACM Comput. Surv.* 54, 1, Article 21 (mar 2021), 37 pages. https://doi.org/10.1145/3431233
[2] Cesar Bernardini, Muhammad Rizwan Asghar, and Bruno Crispo. 2017. Security and privacy in vehicular communications: Challenges and opportunities. *Vehicular Communications* 10 (2017), 13–28. https://doi.org/10.1016/j.vehcom.2017.10.002
[3] Bosch. 2023. *CAN Specification.* https://hdl.handle.net/2027.42/151379 Accessed: 2022, Dec 12.
[4] Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch. 2016. Truck Hacking: An Experimental Analysis of the SAE J1939 Standard. In *Workshop on Offensive Technologies.*
[5] Rik Chatterjee, Subhojeet Mukherjee, and Jeremy Daily. 2023. Exploiting Transport Protocol Vulnerabilities in SAE J1939 Networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS).* https://www.ndss-symposium.org/wp-content/uploads/2023/02/vehiclesec2023-23053-paper.pdf
[6] Stephen Checkoway, Damon McCoy, Danny Anderson, Brian Kantor, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings USENIX Security 2011*, David Wagner (Ed.). USENIX. https://checkoway.net/papers/car2011
[7] Matt Gorbett, Hossein Shirazi, and Indrakshi Ray. 2023. Sparse Binary Transformers for Multivariate Time Series Modeling. Proc. of ACM SIGKDD '23.
[8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[9] IJISC. 2023. *Remote Exploitation of an Unaltered Passenger Vehicle.* https://www.ijisc.com/year-2015-issue-2-article-12/ Accessed: 2022, Dec 12.
[10] Tencent KeenLab. 2021. *Experimental Security Research of Tesla Autopilot.* https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf Accessed: 2023-08-31.
[11] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations.*
[12] Abdullah Zubair Mohammed, Yanmao Man, Ryan Gerdes, Ming Li, and Z. Berkay Celik. 2022. Physical Layer Data Manipulation Attacks on the CAN Bus. In *International Workshop on Automotive and Autonomous Vehicle Security (AutoSec), collocated with NDSS.* 1–5. https://beerkay.github.io/papers/Berkay2022PhyManipulationAutoSec.pdf
[13] Subhojeet Mukherjee, Hossein Shirazi, Indrakshi Ray, Jeremy Daily, and Rose Gamble. 2016. Practical DoS Attacks on Embedded Networks in Commercial Vehicles. In *Information Systems Security.* Springer International Publishing.
[14] Pal-Stefan Murvay and Bogdan Groza. 2018. Security Shortcomings and Countermeasures for the SAE J1939 Commercial Vehicle Bus Protocol. *IEEE Transactions on Vehicular Technology* 67 (2018), 4325–4339.
[15] United States Department of Strasportation. 2023. *Cybersecurity Research Considerations for Heavy Vehicles.* https://rosap.ntl.bts.gov/view/dot/38822 Accessed: 2022, Dec 12.
[16] Brady W. O'Hanlon, Mark L. Psiaki, Jahshan A. Bhatti, Daniel P. Shepard, and Todd E. Humphreys. 2013. Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals. *NAVIGATION* 60, 4 (2013), 267–278. https://doi.org/10.1002/navi.44 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/navi.44
[17] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. 2015. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* 11, 2015 (2015), 995.
[18] SAE International. 2023. *The SAE J1939 Standards Collection.* https://www.sae.org/publications/collections/content/j1939_dl/ Accessed: 2022, Dec 12.
[19] Hossein Shirazi, William Pickard, Indrakshi Ray, and Haonan Wang. 2022. Towards Resiliency of Heavy Vehicles through Compromised Sensor Data Reconstruction. In *Proceedings of the Twelfth ACM Conference on Data and Application Security and Privacy* (Baltimore, MD, USA) *(CODASPY '22).* Association for Computing Machinery, New York, NY, USA, 276–287. https://doi.org/10.1145/3508398.3511523
[20] Hossein Shirazi, Indrakshi Ray, and Charles Anderson. 2019. Using Machine Learning to Detect Anomalies in Embedded Networks in Heavy Vehicles. In *Proc. of 12th International Symposium on Foundations and Practice of Security*, Vol. 12056. Springer.
[21] Systems Engineering, Colorado State University. 2023. *Heavy Vehicle CAN Data.* "https://www.engr.colostate.edu/~jdaily/J1939/candata.html" "Accessed: 2022, Feb 02".
[22] Marko Wolf and Robert Lambert. 2017. Hacking Trucks - Cybersecurity Risks and Effective Cybersecurity Protection for Heavy Duty Vehicles. In *Automotive - Safety Security 2017 - Sicherheit und Zuverlässigkeit für automobile Informationstechnik*, Peter Dencker, Herbert Klenk, Hubert B. Keller, and Erhard Plödererder (Eds.). Gesellschaft für Informatik, Bonn, 45–60.
[23] Marko Wolf, André Weimerskirch, and Christof Paar. 2006. *Secure In-Vehicle Communication.* Springer Berlin Heidelberg, Berlin, Heidelberg, 95–109. https://doi.org/10.1007/3-540-28428-1_6
[24] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. of AAAI Conf. on AI* (2021).