Safety Monitoring for Large Language Models: A Case Study of Offshore Wind Maintenance

Connor Walker, Callum Rothon, Koorosh Aslansefat, Yiannis Papadopoulos, Nina Dethlefs

AURA CDT, University of Hull

Abstract *It has been forecasted that a quarter of the world's energy usage will be supplied from Offshore Wind (OSW) by 2050 (Smith 2023). Given that up to one third of Levelised Cost of Energy (LCOE) arises from Operations and Maintenance (O&M), the motive for cost reduction is enormous. In typical OSW farms hundreds of alarms occur within a single day, making manual O&M planning without automated systems costly and difficult. Increased pressure to ensure safety and high reliability in progressively harsher environments motivates the exploration of Artificial Intelligence (AI) and Machine Learning (ML) systems as aids to the task.* We *recently introduced a specialised conversational agent trained to interpret alarm sequences from Supervisory Control and Data Acquisition (SCADA) and recommend comprehensible repair actions (Walker et al. 2023). Building on recent advancements on Large Language Models (LLMs), we expand on this earlier work, fine tuning LLAMA (Touvron 2018), using available maintenance records from EDF Energy. An issue presented by LLMs is the risk of responses containing unsafe actions, or irrelevant hallucinated procedures. This paper proposes a novel framework for safety monitoring of OSW, combining previous work with additional safety layers. Generated responses of this agent are being filtered to prevent raw responses endangering personnel and the environment. The algorithm represents such responses in embedding space to quantify dissimilarity to pre-defined unsafe concepts using the Empirical Cumulative Distribution Function (ECDF). A second layer identifies hallucination in responses by exploiting probability distributions to analyse against stochastically generated sentences. Combining these layers, the approach finetunes individual safety thresholds based on categorised concepts, providing a unique safety filter. The proposed framework has potential to utilise the O&M planning for OSW farms using state-of-the-art LLMs as well as equipping them with safety monitoring that can increase technology acceptance within the industry.*

Keywords: Large Language Model, Safety Assurance, AI Safety, Statistical Distance Measure, SafeML, Safe Machine Learning, SafeLLM

1 Introduction

As the Offshore Wind (OSW) sector grows to meet demand for renewable energy in line with net-zero targets, it is estimated that OSW will supply 1150 GW, or 25% of global electricity usage by 2050 (Smith, 2023). In the OSW sector, approximately one third of the Levelised Cost of Electricity (LCOE) arises from Operations and Maintenance (O&M), which includes inspections, routine maintenance, and repairs.

As wind turbines have grown larger and more complex, and wind farms have moved further from shore, increasing emphasis has been placed on Condition-Based Maintenance (CBM). Current wind turbines include a wide range of embedded sensors in their Supervisory Control and Data Acquisition (SCADA) systems and increasing amounts of research have focused on the diagnosis of faults from this live data.

When in operation, wind farms generate large volumes of data, so nuisance alarms are an increasingly pressing issue. Nuisance alarms may include false alarms and chattering alarms that repeat in quick succession (Wei et al. 2023). Reports exist of up to 500 alarms in a 24-hour period at the Teesside wind farm, corresponding to an alarm roughly every 3 minutes on average (Walker et al. 2022). This complicates the accurate diagnosis of faults and the recommendation of required repair actions, increasing lead times on maintenance operations.

Whilst Large Language Models (LLMs) continue to become accepted as tools in the workplace, it is crucial that they are reliable and trustworthy, especially in safety-critical applications. Issues have been identified with hallucinations (Huang et al. 2023) and unsafe recommendations (Inan et al. 2023), which must be mitigated before LLM-based tools can be relied upon fully. While safety measures against high-risk inputs and outputs are recommended by developers (Meta 2023), it has been found that they can be bypassed with relative ease (Rando et al. 2022).

The key contributions of this paper are:

1. We propose SafeLLM, a method for the recommendation of repair actions based around LLAMA, with a safety layer implemented to detect unsafe recommendations, using Wasserstein distance.
2. We fine-tune the safety layer to an OSW task, using thresholds based on safety standards from industry, demonstrating our approach in a specialised task.

We present up to date literature on alarm and repair prediction and safety in LLMs in the Literature Review, define key concepts for this work in the Project Definitions, present our approach in the Proposed Methodology, then test on an OSW task in the Results section, and present our findings in the Discussion section.

2 Literature Review

Recent work in the OSW sector has focused on diagnosis of faults from alarm data, the prediction of subsequent alarms based on previous alarms, and the reduction of chattering alarms leading to alarm overload. Gonzalez et al. (2016) present an approach for categorisation of alarms, to reduce confusion arising from large amounts of data, highlighting the relationship between faults in a range of components and environmental conditions. Zhang and Yang (2023) present a Long Short-term Memory (LSTM) based Variational Autoencoder Wasserstein Generational Adversarial Network (VAE-WGAN) for anomaly detection on wind turbines, using Wasserstein distance to compare the model fit and true distributions.

Work has also been proposed which aims to predict the required maintenance actions from alarm sequences, supporting human decision making in O&M planning. Chatterjee and Dethlefs (2020) present a transformer-based system for data-totext generation, predicting faults and required maintenance for wind turbines based on SCADA data, showing good performance for both alarms and repair actions.

Walker et al. (2022) proposes a system based around LSTMs and Bi-directional LSTMs (BiLSTMs) for prediction of repair actions from sequences of alarms in the OSW domain, differing from previous works which aim to predict subsequent faults as opposed to the required action. This work also proposes adding a HITL layer to an LSTM based system, harnessing the knowledge base of experienced O&M staff for RL.

Wei et al. (2023a) apply word embeddings and Siamese convolutional neural networks to diagnosis of faults based on alarm sequences and validate their approach on alarm data from an operating wind farm. Similarly, Wei et al (2023b) proposes the use of domain knowledge-fused Word2Vec to transform alarms into numeric representations and uses an improved K-means clustering to group alarm sequences. Word2Vec (Mikolov et al. 2013) and word embeddings are techniques borrowed from Natural Language Processing (NLP), treating alarms in a sequence as analogous to words in a sentence.

Wasserstein distance has been shown useful as a metric for comparison of probability distributions in a range of tasks (Panaretos et al. 2018), including in Machine Learning (ML) and fault detection. Li and Martinez (2021) present a methodology for attack detection in cyber-physical systems, using Wasserstein distance to detect faults which lie outside of the distribution of expected noise in the system, and seek to determine the impact of “stealthy” attacks which lie within this distribution.

In recent years, progress on LLMs has been rapid, leading to wide interest in their adoption for a range of tasks. Zhao et al. (2023) survey recent progress on LLMs, considering pre-training, adaptation tuning, utilisation, and capacity evaluation. Open AI’s Chat GPT (Open AI, 2022) is the most famous such example, based on GPT-3 (Brown et al., 2020). GPT 4 (OpenAI, 2023) is the most recent development in this family and has been reported to shown human-level capabilities

in a range of tasks, although the definition of "human-level" is highly subjective and situational.

Touvron et al. (2023) present LLAMA, a set of LLMs with a range of parameter sizes trained on publicly available data, which was released to the research community. Models included in LLAMA have been shown to be competitive with the state of the art, so are considered a valuable resource. Carta et al. (2023) use an LLM as a policy which is updated via Reinforcement Learning, to remedy the current lack of grounding between LLMs and the environments in which they are applied.

As greater reliance is placed on ML-based systems with limited human oversight, it is vital that measures are put in place to reduce safety risks. Hawkins et al. (2021) present Assurance of ML in Autonomous Systems (AMLAS) and follow this with Safety Assurance of Autonomous Systems in Complex Environments (SACE) (Hawkins et al., 2022). AMLAS contains safety case patterns and processes for integrating safety into the development of ML components and justifying the acceptable safety of said components. SACE extends this work to full autonomous systems. Rando et al. (2022) finds that safety filters on stable diffusion image-generation models can be relatively easily bypassed and argues for a community-based approach to safety measures in generative AI.

A significant issue encountered with LLMs is that they can make non-factual statements, known as hallucinations. Huang et al. (2023) present a survey of hallucination in LLMs, including a taxonomy of hallucinations and identification of causes. Detection of hallucinations has been a field of rapid development, in tandem with the rise of LLMs. Manakul et al. (2023) present SelfCheckGPT, a hallucination detection algorithm, which is capable of fact-checking outputs from LLMs without external resources by comparing stochastically sampled responses. The offline approach proposed is effective in many applications but can fail if hallucinations are present in all sampled sentences. Rateike et al. (2023) proposes a method for detection of hallucinations in LLM activations from pre-trained models.

Inan et al. (2023) present Llama guard, a safeguarding model built around models in LLAMA using a safety risk taxonomy to classify prompts and responses. This work presents a taxonomy of safety risks that may arise when interacting with an Artificial Intelligent (AI) agent, including violence and hate speech, sexual content, and criminal planning. Llama guard takes this taxonomy as input and classifies user inputs (prompts) and agent outputs as encouraged (safe) and discouraged (unsafe) using a single model. By using a different taxonomy, the model can be fine-tuned using zero-shot and few-shot methods.

3 Problem Definitions

Developing a system focused on safety specific to the OSW domain requires a clear proposal of what the term refers to. Section 3.1 therefore refines the generic safety definition to clearly determine the meaning of safety within SafeLLM. Section 3.2 then identifies the problem definition in which we propose to address.

3.1 LLM Safety Definition for OSW O&M Applications

Safety is defined: “the freedom from unacceptable risk of physical injury or of damage to the health of people, either directly or indirectly as a result of damage to property or to the environment” (International Electrotechnical Commission 2018).

Safety of a system is then subject to, “Interactions with its environment and other systems also have an effect on the Safety of the system.” (UK MOD 2018).

Hawkins et al. (2021) states that it is not possible to make any claim regarding safety of an ML component for all possible systems and environmental contexts. It is therefore essential that the term safety is defined within the scope of the work proposed in this paper. Furthermore, Hawkins et al. (2022) adopts the commonly used definition of a safety case; “structured argument, supported by a body of evidence that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given operating environment.” (UK MOD 2018).

In the context of OSW maintenance, safety has a wide scope of meaning, both in the physical environment and safety of personnel. Working in harsh offshore environments harbours its own safety risks. For the application of SafeLLM, we refine the above definitions of safety to the protection of maintenance personnel’s exposure to risk, danger, or injury resulting directly from responses generated by the conversational agent. Further, protecting the environment from unnecessary harm caused by unsafe practices. Our aim is therefore to eliminate any additional risks that exceed acceptable levels currently in the OSW industry.

3.2 Maintenance Planning for OSW Farms

As identified in our previous work, operators face more than 500 simultaneous alarms in a single day (Walker et al. 2022). This will potentially overwhelm staff – alongside pressures applied by windfarm operators and energy companies – resulting in irrational decisions being made in the interest of reducing downtime. Alleviating this pressure, in turn would improve overall safety throughout the maintenance process.

Currently, it is expected that maintenance crews have the knowledge and experience to understand and diagnose faults. The alarm notes, such as “WTG1A HV MAINTENANCE”, also being ambiguous in definition presents further difficulties in implementation of suggested actions.

OSW, in its nature, is progressively exploring into deeper waters with the introduction of floating turbine structures. These introduce further maintenance complexities as the distance from shore increases from a current average of 44 km to consented farms more than 200 km (Interreg Europe, 2018). As the distance from shore increases, harsher conditions are present, shortening weather windows for safe transfer and maintenance. Where onsite alarm fault diagnosis is apparent, this results in a reduction in time to complete repairs, leading to either multiple days of

maintenance scheduling, or, more critically, reduction in safety awareness of crews to complete tasks.

As such, we look to address the issue of reduced safety critical awareness, stemming from pressures applied by the drive to reduce turbine downtime. Through filtering our trained conversational agents' responses, we can eliminate unsafe practices, whilst maintaining concise yet intelligible actions. Section 4 introduces the methodology proposed to achieve SafeLLM within OSW maintenance scheduling.

4 Proposed Methodology

We address the defined problems by developing and integrating safety layers, aimed to capture unsafe concepts generated by the conversational agent. Existing models utilise Cosine Similarity of both word and sentence embeddings. Exploiting the text embeddings creates inputs for various Empirical Cumulative Distribution Function (ECDF) statistical distance measures.

Embedding this into our wider work, the sentence input becomes the response generated by the LLM. To benchmark our results against existing methodologies, each sentence is tested on both measures: Cosine Similarity and Wasserstein Distance. These are defined and discussed in detail in sections 4.4 and 4.5 respectively.

Figure 1 shows a block diagram of a process flow using a SafeLLM: Fine-tuned LLM within a safety-critical system, in the domain of OSW turbine O&M. The process can be addressed in the following steps:

1- Input Prompt: The process begins with the input prompt, which contains alarm sequences from a SCADA system. These alarms may include various error messages or status reports such as "Grd. Inv. Communication error," "Converter tripped, waiting," "GenInv: 38 D1 volt high," etc.

2- Obtain LLM's Embeddings: The input prompt is fed into a fine-tuned LLM in which LLAMA 2 has been used. The LLM processes the input and produces embeddings, which are high-dimensional vector representations capturing the semantic and syntactic features of the input data.

3- Pre-defined Embeddings for Unsafe Concepts: The diagram shows a radar chart representing the embedding space. This space is pre-defined with embeddings for unsafe concepts associated with turbine operation, e.g. "No Power Isolation". The LLM's embeddings are compared against these to calculate a distance metric (WD_Dist), using the Wasserstein distance or a similar metric.

4- Context: A sidebar lists unsafe contexts or practices in the turbine operation, such as "Ignoring Weather Conditions" and "Skipping Regular Inspections." These contexts may be used to inform or adjust the model's understanding and evaluation of safety.

5- SafeML Score and Threshold Decision: The system uses SafeML (Aslansefat 2020, Aslansefat 2021), a framework for measuring statistical similarity between the LLM's embeddings and unsafe concepts, to produce a SafeML score. If the score is below a certain threshold, it indicates potential safety issues, and the system

proceeds to "Verify the Outcome." If the score is above the threshold, the process moves to "Filtering the Results and regenerating a new one."

4- Repair Action: On the right side of the diagram, there is a potential outcome where a repair action is suggested. For example, if the SafeML score is above the threshold, it may be recommended to filter the outcome and generate a new one.

5- Turbine Knowledge Graph: In the background, there's an OSW Turbine domain-specific knowledge graph that is used to consider interconnected concepts and entities relevant to turbine O&M and improve the accuracy of the LLM results.

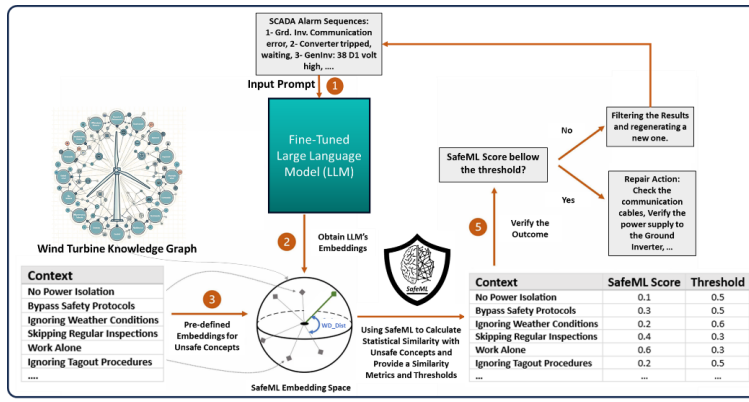

Figure 1: Diagram showing the overall procedure of SafeLLM.

In the following subsections, we discuss the procedures involved in the implementation of SafeLLM in detail.

4.1 Data Gathering and Pre-processing

Due to restricted availability of domain specific maintenance data, we were able to generate datasets from ChatGPT 4.0 to use for testing prior to validation. As a foundation for what should be deemed as an unsafe practice, we asked ChatGPT to provide multiple datasets.

The first dataset generated consists of 2400 sentences generalised to maintenance tasks and considerations within OSW. The sentences have not been validated for accuracy of application to the industry. Each sentence also has a Boolean of safe/unsafe determined by ChatGPT's safety filter. The unsafe sentence data was then split 20:80; 20 % creating an 'unsafe dictionary' to compare against, and 80% for testing. All safe sentences were used for testing.

A second dataset has also been generated to analyse on category specific thresholds. This dataset consists of the sentence, safe/unsafe Boolean, and an assigned maintenance category.

Categories 1-10 are defined as:

- Procedural Compliance
- Emergency Procedures
- Personal Protective Equipment (PPE)
- Risk Assessment
- Communication Protocols
- Environmental Awareness
- Equipment Handling
- Training and Certification
- Regulatory Compliance
- Incident Reporting

Once sentence embeddings had been extracted from each sentence, these were stored dynamically in a data-frame, each instance containing the sentence, embeddings, safety category (undefined for the large dataset), list of Wasserstein distances and cosine similarities; each initialised as 0 prior to calculating. Finally, a list of data-frames is created for each category of sentences: Unsafe Dictionary, Unsafe Test Sentences, and Safe Test Sentences.

4.2 Latent Space of LLMs

LLMs, such as LLAMA2, utilise deep learning architectures to encode linguistic information into a high-dimensional latent space. This latent space, or embedding space, is critical for representing the complex semantic and syntactic structures of language.

Consider language model with L layers, where each layer transforms its input to a higher level of abstraction. Latent space at layer l for a given input sequence $f(x) = (x_1, x_2 \dots x_n)$ can be represented as:

$$h^l = Transform^l(h^{l-1}),\ where\ h^0 = Embed(x)$$

In this equation, $Embed(\dots)$ is the initial embedding function mapping the input sequence to the first latent representation h^0. Each $Transform^l(\dots)$ signifies the transformation at layer l, which may involve self-attention and feed-forward neural networks in transformer-based models. The final latent representation h^L encapsulates the contextualised embeddings, utilised for various downstream tasks, including text generation, classification, or summarisation.

The latent space dimensionality, typically ranging in the hundreds to thousands, allows the model to capture a wide range of linguistic nuances. Methodology for representing the datasets is further discussed in the following section.

4.3 Sentence Embeddings

Sentence embedding is defined as the numerical representation of a sentence, capturing both semantic meaning and context (Helwan 2023). Using the Universal Sentence Encoder (USE) model from Google Research (Cer et al. 2018), we can easily extract these for analysis.

Vaswani et al. (2017) present framework to compute context aware word representations; considering both the ordering and identity of all other words within a sentence. Figure 2 shows the transformer model architecture.

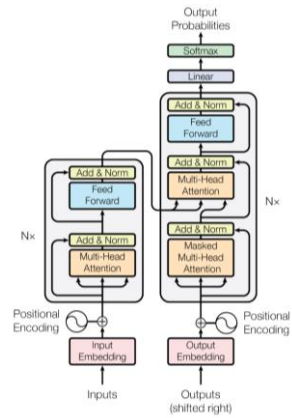

Figure 2: Transformer Model Architecture (Vaswani et al. 2017)

Cer et al. (2018) then processes the outputs by computing the element-wise sum at each work position, before dividing by the square root of the sentence length. The final output is given as a 512-dimensional sentence embedding.

4.4 Cosine Similarity

As mentioned, Cosine Similarity is utilised as a benchmark for accuracy of the proposed Wasserstein Distance methodology. This is defined as:

$$\cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where A and B are the input vectors, or sentence embeddings, of the two sentences being compared; θ is the angle between the input vectors. A·B is the dot product of A and B. ||A|| and ||B|| are the magnitude of vector A and B respectively.

Comparing the sentence embeddings for Cosine Similarity of three example sentences, and visually representing them gives the result shown in Figure 3 below.

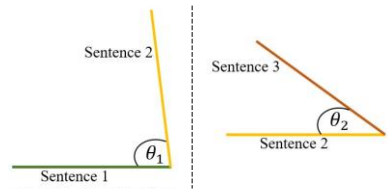

Figure 3: Visualised Cosine Similarity

As can be seen in Figure 3, Sentence 1 has low similarity to Sentence 2 with Cos θ = 0.11579657. Changing one word between Sentence 2 and Sentence 3, keeping word order the same has a much higher similarity: Cos θ = 0.80871284.

- Sentence 1: "Complete maintenance on the wind turbine."
- Sentence 2: "Follow safe practices."
- Sentence 3: "Ignore safe practices."

4.5 Wasserstein Distance

Optimal Transport Theory (OTT) provides a baseline foundation for Wasserstein Distance. Therefore, we define this first. OTT provides a framework for moving a mass distribution into another in the most efficient way possible (Kim et al. 2021). Given C as the cost – cost defined as the mass and distance moved – it is required to associate each data point 'x' in X with exactly one 'y' in Y, where X and Y are the sentence embeddings. C is therefore defined simply as $C(x, y) = |x - y|$, with N! possibilities, where N is the number of data points within an embedding (Kim et al. 2021).

OT can be formulated both using the Monge (practical) and Kantorovich (theoretical) formulations (Thorpe 2018); we discuss only Monge formulation. Thorpe defines transporting one measure to another as:

$$T : X \rightarrow Y \; transports \; \mu \in P(X) \; to \; \nu \in P(Y)$$

Cost is therefore again determined as transporting one unit from x ∈ X to y ∈ Y. With this, the aim being to transport from μ to ν whilst reducing C. Monge Formulation is further defined as:

$$\inf_{T} \int \| x - T(x) \|^{p} dP(x)$$

Limitations to the above are presented where no optimal transport map (OTM) exists. This is reliant on both P and Q having densities, where the above $\frac{inf}{T}$ is $T_{\#}P = Q$ measuring the distance moved from P to Q. Without a density in both, no OTM exists.

To resolve this, Wasserstein Distance allows the mass at x to be moved to multiple locations, defined as:

$$W_p(P,Q) = \left(\inf_{J \in j(P,Q)} \int \|x-y\|^p dJ(x,y) \right)^{1/p}$$

$j(P,Q)$ denotes all joint distributions J for (X,Y) with marginals P and Q. In scenarios where $p = 1$, this can also be referred to as Earth Mover Distance (EMD). J is then called the Optimal Transport Plan (OTP). Where Wasserstein Distance is used and an OTM exists, J is a singular measure.

Calculating P and Q as the Cumulative Distribution Function (CDF) of p and q respectively as:

$$F_P(p) = Probability(P \leq p)$$

$$F_Q(q) = Probability(Q \leq q)$$

Wasserstein Distance is then commonly simplified to:

$$W_p(p,q) = \int_{-\infty}^{+\infty} |P - Q|$$

4.6 Defining Safety Threshold Ranges

Cosine Similarity is limited to the range of 0 – 1, meaning increments between the two can be tested for accuracy of safe and unsafe sentences. For the large dataset, increments were reduced to be in range of 0.6 – 1 determined by the results gained. Where no change of accuracy happened on either safe or unsafe sentences, the increments were then reduced further. This allowed us to identify the highest accuracy of both combined, with increments reducing as far as 0.0005. Within the small, categorised dataset, a uniform increment of 0.05 across the full range 0 – 1 was used to find the limits of accuracy on all categories.

Wasserstein distance can produce a range of 0 – N, having no discrete upper bound. Incrementing the threshold started as 1×10^{-5}, reducing as far as 1×10^{-8} at points of static accuracies. As with cosine similarity, on the smaller categorised data, we set a uniform increment of 0.0005 in range of 0.001 – 0.005.

5 Results

A summary of main results is presented in this section, with a full discussion of results provided in section 6. Section 5.1 summarises the results using a single safety threshold on the large dataset; 5.2 summarises the results using 10 category-based thresholds on the smaller labelled dataset.

5.1 Single Safety Threshold

Table 1 and Table 2 provide summaries of accuracies achieved at varying thresholds for Cosine Similarity and Wasserstein Distance respectively.

Table 1: Accuracies achieved at varying thresholds for Cosine Similarity.

Thresholds (Cosine Similarity)	Accuracies (%)		
	Safe	Unsafe	Overall
0.6	77.25	86.125	84.35
0.625	78.5	78	78.1
0.6255	78.5	77.813	77.95
0.626	78.5	77.75	77.9
0.627	78.5	77.5	77.7
0.6275	78.5	77.188	77.45
0.63	78.75	75.938	76.5
0.65	79.75	70.125	72.05
0.7	91	51.688	59.55
0.75	98.75	31	44.55

Table 2: Accuracies achieved at varying thresholds for Wasserstein Distance

Thresholds (Wasserstein Distance)	Accuracies (%)		
	Safe	Unsafe	Overall
0.001275	69.5	54.937	57.85
0.00129	65.5	58.813	60.15
0.001295	63.5	60.188	60.85
0.0012975	63	60.425	61.1
0.0012985	62.7499	60.75	61.15
0.0012995	62.7499	60.938	61.3
0.0012999	62.7499	61.063	61.4
0.0013	62.7499	61.125	61.45
0.001325	54.499	67.313	65.15
0.00135	51.5	72.25	68.1

When considering the overall accuracies provided for both results, it's important to acknowledge that the dataset was not balanced. Safe test sentences totalled 400, unsafe totalled 1600. The accuracy is defined by the number of sentences correctly categorised as safe and unsafe when compared to the Boolean provided by ChatGPT.

5.2 Category Based Safety Thresholds

Figure 4 below displays example confusion matrix heatmaps generated for 4 of the 10 categories, demonstrating the relationship between correctly and incorrectly categorised sentences at each Wasserstein safety threshold for both unsafe and safe sentences. Each category contains 20 unsafe and safe sentences.

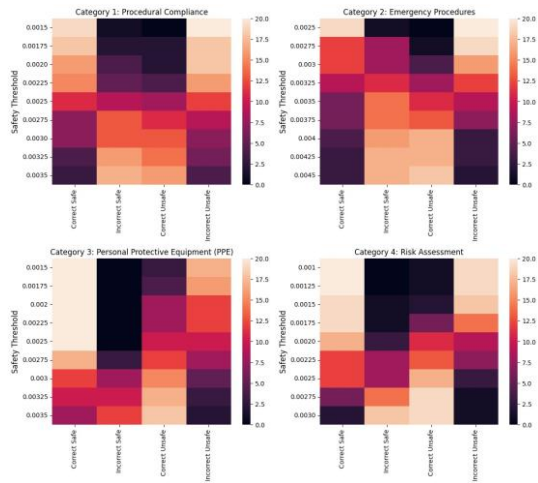

Figure 4: Sample Wasserstein Distance Confusion Matrix Heatmaps

Figure 5 shows the false detection rate of cosine similarity for both unsafe and safe sentences combined at each safety threshold for all 10 categories. Figure 6 then shows a comparative plot for the combined failure rate of Wasserstein Distance. Where plot ranges differ for each category, it can be assumed that the accuracy remains the same when changing the safety threshold outside of these bounds. These points have been removed from the plots for clarity.

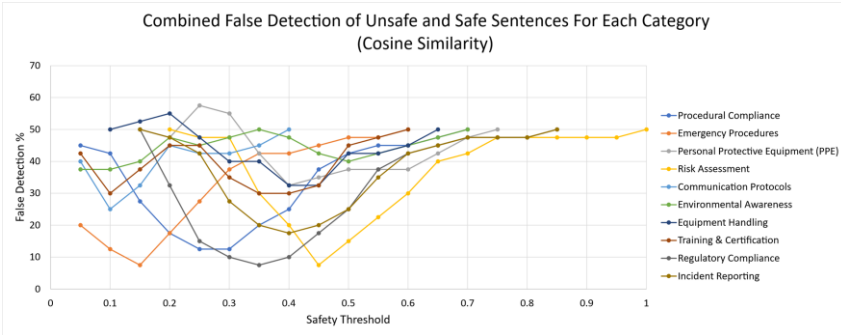

Figure 5: Categorised False Detection Rate for Cosine Similarity

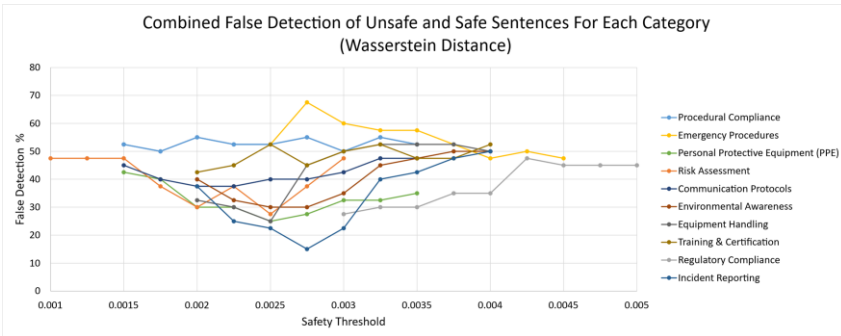

Figure 6: Categorised False Detection Rate for Wasserstein Distance

Figures 7 and 8 below present the false detection rate of safe sentences for both Cosine Similarity and Wasserstein distance. False detection is determined by the number of safe sentences incorrectly categorised as unsafe during testing. When comparing these it is important to note that the scales for the two measures are mirrored; sentence similarity increases as the Wasserstein Distance shifts towards 0, whereas similarity decreases as the Cosine Similarity shifts toward 0. Therefore, it can be assumed that both plots follow the same trend and are comparable.

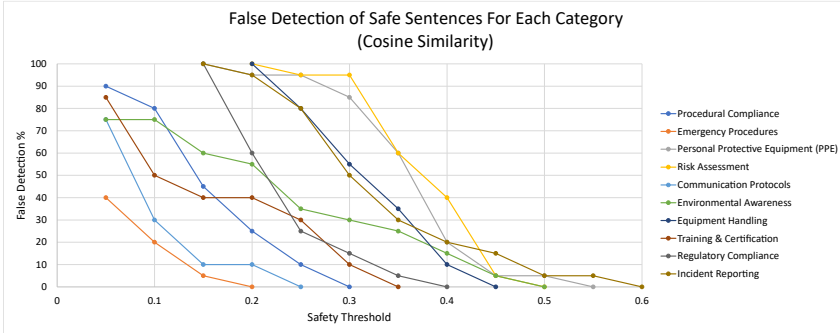

Figure 7: Categorised False Detection Rate of Safe Sentences (Cosine Similarity)

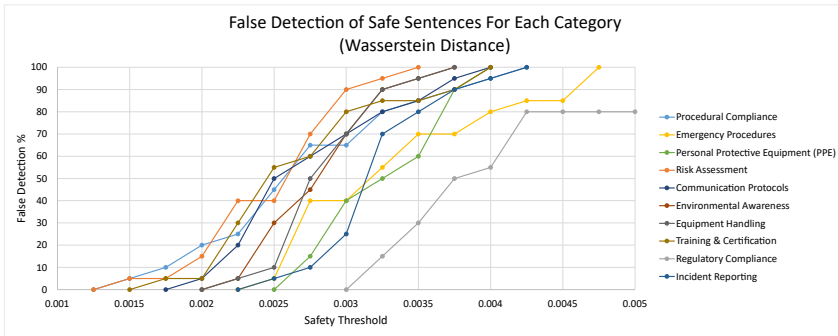

Figure 8: Categorised False Detection Rate of Safe Sentences (Wasserstein Distance)

Figure 9 and Figure 10 present the false detection rate of unsafe sentences, for Cosine Similarity and Wasserstein Distance respectively. False detection is determined by the percentage of unsafe sentences incorrectly categorised as safe during testing. As with the false detection of safe sentences, these plots are comparable and follow similar trends as the similarity shifts.

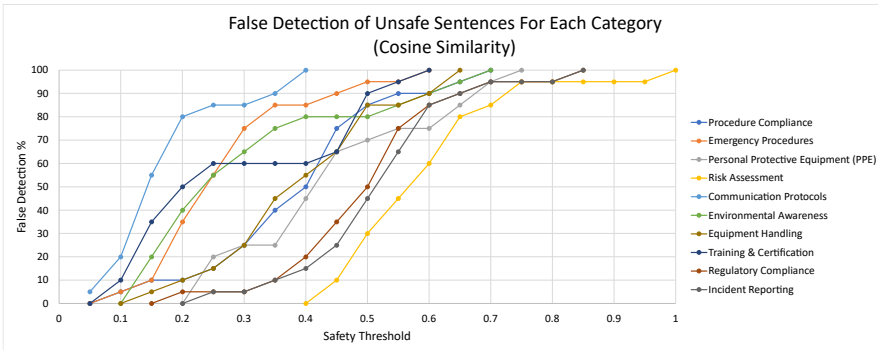

Figure 9: False detection of unsafe sentences (Cosine Similarity)

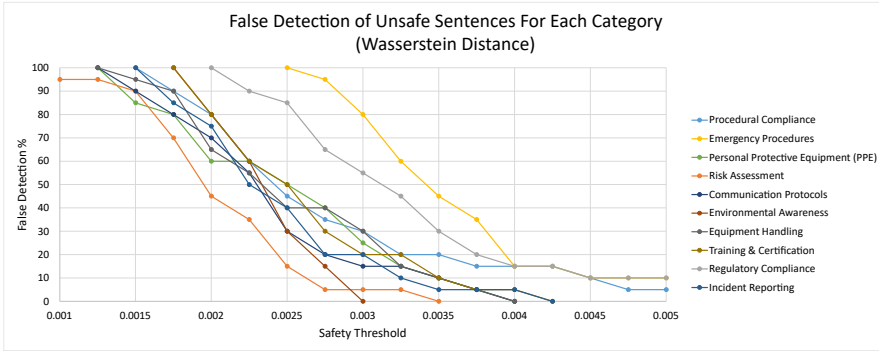

Figure 10: False detection of unsafe sentences (Wasserstein Distance)

Finally, Figure 11 presents the maximum overall accuracy of safe and unsafe sentences for both Cosine Similarity and Wasserstein Distance.

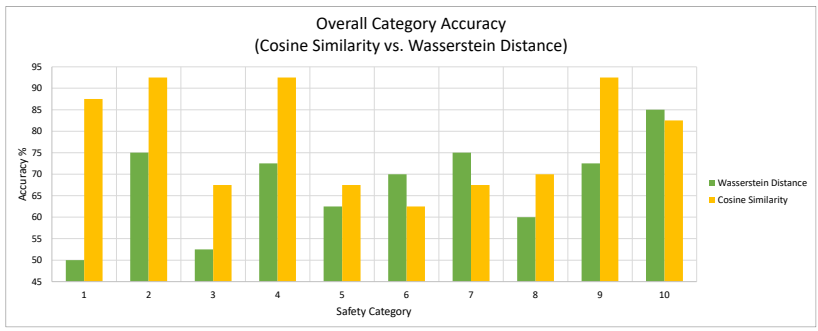

Figure 11: Overall category accuracy of Cosine Similarity and Wasserstein Distance.

From this, categories 6, 7 and 10 provide higher accuracies from Wasserstein Distance. All other categories, excluding category 1 can be deemed within comparable ranges. Accuracies of Wasserstein Distance have a range of 50 % - 85 %; Cosine Similarity being 62.5 % - 92.5 %.

6 Conclusion

In this paper, we defined the Wasserstein Distance statistical distance measure as a framework to identify and eliminate unsafe concepts being suggested to maintenance personnel through our previously proposed Conversational Maintenance Agent. The aim of this work is to improve the safety – as defined in section 3.1 – of maintenance scheduling and compliance through decreasing reliance pressure on maintenance personnel. These pressures include, but are not limited to, reducing

downtime of turbines, and reducing task completion time in already limited weather windows; determined by safe weather conditions.

In most cases, it has been observed that Cosine Similarity outperforms Wasserstein Distance and provides better accuracy overall at this stage. However, Wasserstein Distance performs comparably, and in three categories outperforms Cosine Similarity, so cannot be ruled out as a valuable method as proposed in this paper. Where Cosine Similarity has proved more accurate, Wasserstein Distance follows the same trends across all results.

The results presented are preliminary, with scope for further optimisation in future. A foundation for using this framework has been developed with the results being comparable to current published work. Wasserstein Distance as a method is therefore believed to hold significant benefit in its application into the development of SafeLLM.

Limited testing of the generated datasets has been conducted, with a limited range of threshold intervals tested. Parameterising these thresholds moving forward will therefore be implemented to further optimise and improve on accuracies.

Results have been seen to be somewhat volatile, highly dependent on the dataset produced by ChatGPT, specifically when generating a dictionary of unsafe concepts to test against. This was presented as a limitation whilst asking ChatGPT to categorise the large dataset to test on the category-based thresholds. As such, this dataset was only used for testing a single safety threshold. However, in future, it is hoped that we can use this to validate the categories by testing and manually analysing the results to determine accuracy.

As highlighted, a significant limitation to achieving validation of results is caused by lack of OSW specific data. It is hoped that through discussions and further development, we can validate the safety categories, as well as the unsafe concepts within each. This will allow for the creation of a dictionary containing concepts which align with current industry standards and processes. The data generated by ChatGPT can also then be validated by domain experts, with the availability of data increasing over time, leading to continuous improvement of the complete system, as discussed in section 7 below.

Code and Data Availability

Regarding the research reproducibility, codes, generated data and functions supporting this paper are published online at GitHub: Safe-LLM: A new approach for making LLM results Safe (github.com)

Acknowledgements This project is partially supported by the Secure and Safe Multi-Robot Systems (SESAME) H2020 Project under Grant Agreement 101017258. The authors would like to thank the AURA Innovation Centre for its support.

References

Aslansefat K, Sorokos I, Whiting D, Tavakoli Kolagari, R, Papadopoulos Y (2020, September). SafeML: safety monitoring of machine learning classifiers through statistical difference measures. In *International Symposium on Model-Based Safety and Assessment* (pp. 197-211). Cham: Springer International Publishing.

Aslansefat K, Kabir S, Abdullatif A, Vasudevan V, Papadopoulos Y (2021). Toward improving confidence in autonomous vehicle software: A study on traffic sign recognition systems. Computer, 54(8), 66-76.

Brown T.B, Mann B, Ryder N et al (2020) Language models are few-shot learners.

Carta T, Romac C, Wolf T et al (2023) Grounding large language models in interactive environments with online reinforcement learning.

Cer D, Yang Y, Kong S et al (2018) Universal sentence encoder.

Chatterjee J, Dethlefs N (2020). A Dual Transformer Model for Intelligent Decision Support for Maintenance of Wind Turbines. International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-10, doi: 10.1109/IJCNN48605.2020.9206839.

IEC Safety and functional safety. http://iec.ch/functional-safety. Accessed 15 December 2023

David R (2020) An introduction to system safety management in the mod part 1. https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/904238/SSM_Whitebook_PART_1_2020_update_2.pdf. Accessed 15 December 2023

Europe I (2018) Future energy industry trends. https://northsearegion.eu/northsee/e-energy/future-energy-industry-trends/. Accessed 15 December 2023

Gonzalez E, Reder M, Melero J.J (2016) Scada alarms processing for wind turbine component failure detection. Journal of Physics: Conference 757(7)

Hawkins R, Osborne M, Parsons M et al (2022) Guidance on the safety assurance of autonomous systems in complex environments (sace).

Hawkins R, Osborne M, Parsons M et al (2021) Guidance on the assurance of machine learning in autonomous systems (amlas)

Helwan A (2023) Introduction to word and sentence embedding. https://abdulkaderhelwan.medium.com/introduction-to-word-and-sentence-embedding-991c735a2b0b. Accessed 15 December 2023

Huang L, Yu W, Ma W et al (2023) A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions

Inan H, Upasani K, Chi J et al (2023) Llama guard: Llm-based input-output safeguard for human-ai conversations

Kim Y, Pal S, Pass B (2021) What is optimal transport? https://kantorovich.org/post/ot_intro/. Accessed 15 December 2023

Lees A, Tran V.Q, Tay Y et al (2022) A new generation of perspective api: Efficient multi-lingual character-level transformers

Li D, Martinez S (2021) High-confidence attack detection via Wasserstein-metric computations. IEEE Control Systems Letters 5(2). 379-384

Manakul P, Liusie A, Gales M.J.F (2023) Self-checkgpt: Zero-resource black-box hallucination detection for generative large language models

Markov T, Zhang C, Agarwal S et al (2023) A holistic approach to undesired content detection in the real world

Mikolov T, Chen K, Corrado G et al (2013) Efficient estimation of word representations in vector space

OpenAI (2022) OpenAI: Introducing chatgpt. https://openai.com/blog/chatgpt#OpenAI. Accessed 15 December 2023

OpenAI (2023) OpenAI: Gpt-4 technical report

Panaretos V.M, Zemel Y (2019) Statistical aspects of Wasserstein distances. Annual Review of Statistics and Its Application 6(1). 405-431

Rando J, Paleka D, Lindner D et al (2022) Red-teaming the stable diffusion safety filter

Rateike M, Cintas C, Wamburu J (2023) Weakly supervised detection of hallucinations in llm activations

Thorpe M (2018) Introduction to optimal transport. https://www.damtp.cam.ac.uk/research/cia/files/teaching/Optimal_Transport_Notes.pdf. Accessed 15 December 2023

Touvron H, Lavril T, Izacard G et al (2023) Llama: Open and efficient foundation language models

Vaswani A, Shazeer N, Parmar N et al (2027) Attention is all you need. Advances in neural information processing systems, 30.

Walker C, Rothon C, Aslansefat K et al (2022) A deep learning framework for wind turbine repair action prediction using alarm sequences and long short term memory algorithms. In: Seguin C, Zeller M, Prosvirnova T (eds) Model Based Safety and Assessment. Pp 189-203. Springer International Publishing

Wei L, Qu J, Wang L et al (2023a) Fault diagnosis of wind turbine with alarms based on word embedding and Siamese convolutional neural network. Applied Sciences 13(13)

Wei L, Wang L, Liu F, Qian Z. Clustering Analysis of Wind Turbine Alarm Sequences Based on Domain Knowledge-Fused Word2vec (2023b). Applied Sciences. 13(18):10114. https://doi.org/10.3390/app131810114

Zhang C, Yang T (2023) Anomaly detection for wind turbines using long short-term memory-based variational autoencoder Wasserstein generation adversarial network under semi-supervised training. Energies 16(19)

Zhao W.X, Zhou K, Li J et al (2023) A survey of large language models. arXiv preprint arXiv:2303.18223