Machine Learning (2024) 113:843–861
https://doi.org/10.1007/s10994-023-06398-7

Automotive fault nowcasting with machine learning and natural language processing

John Pavlopoulos[1] · Alv Romell[3] · Jacob Curman[3] · Olof Steinert[2] · Tony Lindgren[1,2] · Markus Borg[3] · Korbinian Randl[1]

Received: 14 February 2023 / Revised: 11 July 2023 / Accepted: 16 August 2023 /
Published online: 2 October 2023

Abstract
Automated fault diagnosis can facilitate diagnostics assistance, speedier troubleshooting, and better-organised logistics. Currently, most AI-based prognostics and health management in the automotive industry ignore textual descriptions of the experienced problems or symptoms. With this study, however, we propose an ML-assisted workflow for automotive fault nowcasting that improves on current industry standards. We show that a multilingual pre-trained Transformer model can effectively classify the textual symptom claims from a large company with vehicle fleets, despite the task's challenging nature due to the 38 languages and 1357 classes involved. Overall, we report an accuracy of more than 80% for high-frequency classes and above 60% for classes with reasonable minimum support, bringing novel evidence that automotive troubleshooting management can benefit from multilingual symptom text classification.

Keywords Automotive fault nowcasting · Natural language processing · Multilingual text classification

1 Introduction

Fault diagnosis is the task of detecting the fault that caused a problem or unexpected behaviour to a subject. If the subject is a human being and the nature of the problem is medical (e.g., COVID-19), a reasonable diagnostics process comprises the physician who reads or listens to the patient's symptoms, looks at any radiographs or echocardiograms, studies the medical records, and so on. Then, the physician may conclude regarding the root cause and suggest the right treatment. In the automotive industry, a similar process is followed, because ensuring functional safety over the product life cycle while limiting maintenance costs has become a major challenge (Theissler et al., 2021; Zhao et al., 2021). Natural Language Processing (NLP) has facilitated medical diagnostics (Irving et al., 2021; Izquierdo et al., 2020) and issue management in engineering contexts, e.g., telecommunications (Jonsson et al., 2016) and banking (Aktas & Yilmaz, 2020). In this work, by taking a vehicle

Editors: Fabio Vitale, Tania Cerquitelli, Marcello Restelli, and Charalampos Tsourakakis.

Extended author information available on the last page of the article

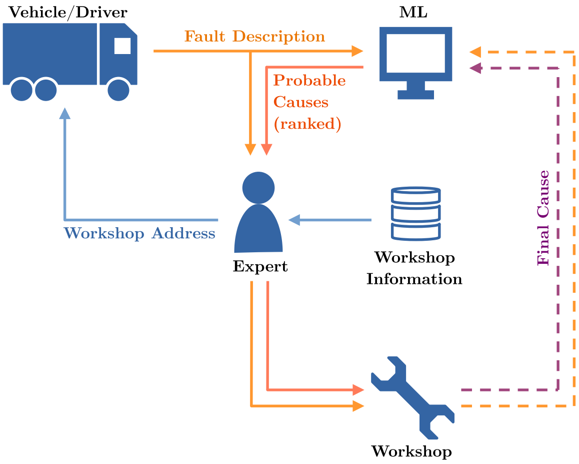


Fig. 1 Human-in-the-loop-architecture of automotive fault nowcasting

fleet as a subject, we show that NLP-assisted troubleshooting management is feasible also in the automotive domain. In line with previous work, we show that it can serve as an additional channel to serve corrective maintenance and health management (Nath et al., 2021; Safaeipour et al., 2021; Theissler et al., 2021; Vaish et al., 2021).

1.1 Automotive fault diagnosis

Upon a fault (e.g., mechanical) a driver typically communicates with the fleet manager, i.e., the one responsible for the vehicles in the company's fleet throughout each vehicle's life cycle. The driver shares the details of the problem as a text message (email, SMS, voice mail, etc.) and the department advises the driver to move the truck to a dedicated support centre (workshop) nearby. An expert is assigned to diagnose the root cause of the fault and when the diagnosis is complete, the problem can be fixed (e.g., by ordering replacement parts) so that the driver can continue the job routine. The time of the aforementioned process is not short. The driver might be suggested a service-centre that is suboptimal for the fault in question or a technician that does not have the right skill set (e.g., high voltage for EVs), which leads to a longer time before the truck is repaired. However, if the problem at hand was known at an early stage, the company could plan accordingly and find an empty slot in a workshop, order spare parts, prepare invoices, etc.

We propose introducing an assistive large language model that can aid a human expert tasked with coordinating fleet repairs in making informed decisions at an early time. An overview of the resulting human-in-the-loop architecture is shown in Fig. 1. In specific, the fault description, which is usually written in natural language (e.g., emails, SMS), before arriving at the company's front desk could first be passed through a text classifier, trained to detect the fault behind the claim. At this stage, neither the end-user nor the company knows the problem. A human expert will then decide on the most fitting service centre based on the original fault description and a ranking of the most probable causes. They then communicate the workshop information to the driver and fault description and probable ranking to the workshop. The assumption, however, is that a classifier can learn to predict the underlying fault based solely on the textual claim while a system-predicted fault could: (a) assist the mechanics/diagnostics teams toward reaching faster to the root cause of the problem (speedier troubleshooting); (b) reduce the human error, given that the tired or inexperienced expert will be assisted with the system-prediction; (c) allow ordering of parts in a timely manner, by detecting early patterns in the fault reports, hence leading to better organisation of the logistics. Finally, the problem description together with the detected fault can be reused as pre-training and fine-tuning examples for an updated language model.

1.2 Contribution

This study focuses on data shared by Scania, attempting to classify the textual descriptions of the problems, as these were registered through work orders in workshops, regarding the actual root cause for the vehicle malfunction. We formed a dataset of 452,071 texts, written in 38 languages, and classified manually into 1357 classes. We then investigated whether large-scale text classification could assist with faster resolution of faults, hence leading to a better working environment for drivers and mechanics, improved logistics, and better troubleshooting management overall. Although AI-enabled prognostics and health management are well-studied fields (Zhao et al., 2021), in this work we show that NLP can open a new path for automotive troubleshooting management in *effectiveness (diagnostics assistance)*, *efficiency (speedier troubleshooting)*, and *management of decision-making trade-offs (better-organised logistics)*.

Different from medical fault diagnosis, which is most often based on image input, the large-scale multi-class and -linguality nature of automated fault report management in the automotive domain, combined with the terminology, constitute a specific task and a challenging problem from an NLP perspective. By training a text classifier to produce helpful rankings of probable causes of these faults, we show the feasibility of a novel human-in-the-loop system for fault detection in the automotive industry. In this context, this study makes three contributions in light of previous work:

- We present the first large-scale study that demonstrates the applicability of automated fault report management in the automotive domain, reporting promising results for an industrial case at Scania.
- We show that state-of-the-art NLP methods can handle effectively multi-lingual fault diagnosis, hence unlocking the use of pre-trained masked language Transformer models (Conneau et al., 2019; Devlin et al., 2018) for fault diagnosis in the automotive domain and beyond, where customer support receives textual fault claims.
- We propose an ML-assisted human-in-the-loop workflow for handling text-based fault reports that improves on current industry standards and show its feasibility.

Additionally, we present a comparison of approaches for treating multilingual texts in an automotive context. By comparing the efficiency of a multilingual model to a single-language model on pre-translated texts, we show that while the former is better for well supported classes, the latter can be more efficient for classes with low representation in the training data.

In the remainder of this article, Sect. 2 presents the related work and Sect. 3 presents the dataset. Section 4 provides an empirical analysis and Sect. 5 discusses the findings under the light of an error analysis. The paper concludes with our findings and suggested directions for future work.

2 Related work

Fault diagnosis is a well-studied problem (Safaeipour et al., 2021) and it can be part of corrective maintenance, defined as the task of repairing a system after a failure occurred (Theissler et al., 2021). Fault diagnosis is also related to failure detection and predictive maintenance (Carvalho et al., 2018), and prognostics and health management (Biteus & Lindgren, 2017; Nath et al., 2021). Thanks to digitalisation, the management of fault reports in information systems has provided organisations with new opportunities to increase the level of automation in related work tasks.

Guided by the "big data" mindset, research and practice have successfully used machine learning (ML) to automate fault report management. In large organisations, the inflow of textual fault reports often contains actionable patterns for ML models to learn. The software engineering community was an early adopter of this approach and numerous papers on training classifiers to analyse bug reports have been published. Common applications include duplicate detection, bug prioritisation, and automated team assignment for rapid bug fixing (Borg & Runeson, 2014).

NLP is also used to facilitate business processing, but most often through the development of task-directed dialogue systems (chatbots), e.g., to assist user satisfaction assessment (Borsci et al., 2022) or troubleshooting (Thorne, 2017). Although machine learning is present in such studies, they do not aim to assist a diagnostics process but rather tasks such as intent classification (Adamopoulou & Moussiades, 2020). The broader potential of NLP in prognostics and health management, however, is not disregarded, with tasks such as keyword detection in maintenance records and prediction of the failure type remedied, proposed in Fink et al. (2020).

Most previous work that attempted to employ NLP to address fault diagnosis relied on simple bag-of-words models or the TFIDF statistical measure followed by standard techniques available in open-source ML libraries. Jonsson et al. (2016), for example, compared most techniques available in WEKA[1] for training classifiers for telecommunications fault reports at Ericsson. Aktas and Yilmaz (2020) presented a similar study for fault reports in the context of İşbank, the largest bank in Turkey. Vaish et al. (2021) trained various ML classifiers for fault reports in the domain of power systems. Recently, deep learning has also been applied to classify fault reports, including recurrent and convolutional neural networks (Zhang et al., 2022) or transfer learning (Qian et al., 2022). All these approaches, however, are outdated, since BERT (Devlin et al., 2018) set a new state of the art in several NLP tasks.

3 The multilingual and multiclass dataset

Our dataset consists of textual descriptions of malfunctions in trucks. We extracted 452,071 texts from a database containing work orders placed in workshops over the past years. Each text is labeled corresponding to a *main group* and a *class*. The main group refers to what overarching segment the faults belong to (e.g., 'engine' or 'chassis') and the class reflects the particular sub-part that has been identified as the root cause for the vehicle malfunction in a workshop (e.g., 'oil pump' or 'yoke'). The average support of unique texts per class is

[1] https://www.cs.waikato.ac.nz/ml/weka/.

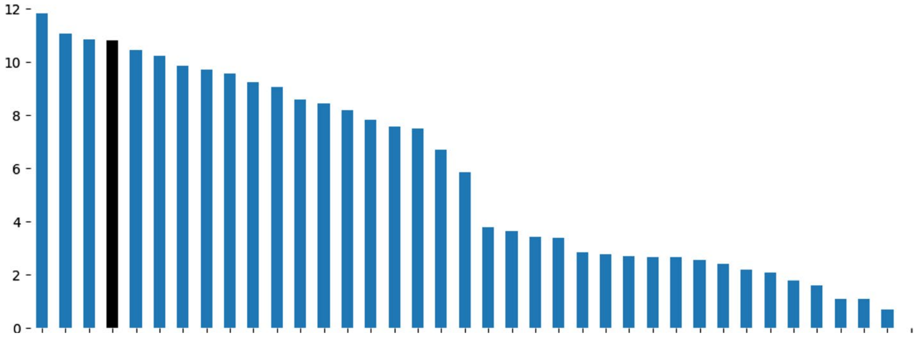


Fig. 2 The (log) frequency distribution of the languages in the data. Portuguese, German, and English are the most frequent languages of the texts, followed by an undefined language (in black)

267.2, ranging from 1 to 6875 texts corresponding to the same class/problem. Furthermore, the same text may be classified into multiple classes, with a single text describing 1.04 classes on average ($min. = 1, max. = 77$). In all our experiments, we used 52% of the data for training, 24% for development and 24% for testing. The shortest text contains as little as one word, whereas the longest one consists of over 350 words.

3.1 Language distribution

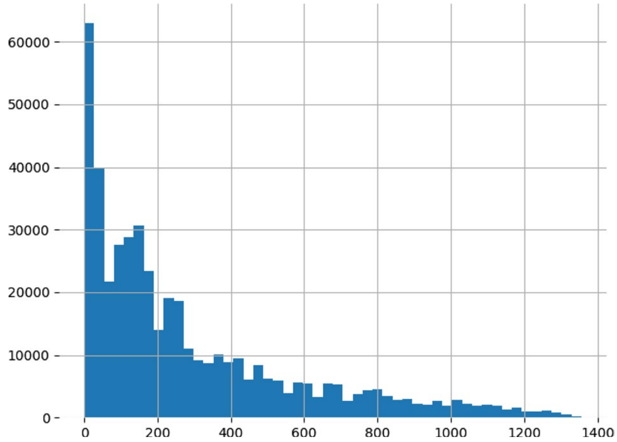
Thirty-eight languages have been identified in the data with the help of the Amazon Translate language detection service.[2] In Fig. 2, which shows the most frequent languages, a large imbalance between the languages can be observed. The ten most frequent languages make up 93.3% of the total samples while the ten least frequently predicted languages make up 0.01%. The ten most frequent languages are the following (unordered)[3]: English, German, Swedish, Finnish, Norwegian, Danish, French, Dutch, Portuguese, and Italian.

The translation service was not able to identify a language for 49,652 texts in our data. These were assigned an unknown language (UNK; highlighted in black in Figs. 2 and 4). Out of the texts assigned to the unknown language, 35,162 were empty strings that were removed. Out of the remaining 14,490 texts, 9165 were unique. Through manual inspection, it is possible to identify flaws in language detection and to find translations of cases where a language has been identified while the translation was not accurate. An example is “110-5002.06 SKARVKOPPLING 6 MM PLAST”, which is Swedish predicted to be Vietnamese; or “TMI 04 15 02 19” and “tpm 48180”, which were incorrectly predicted as Haitian and Indonesian, or the detected language “Esperanto”, which was not part of the dataset. Although we cannot exclude the case that another translation service would handle such cases better, we find that the translation of these texts is hard. In Table 1 the mentioned examples are shown together with two correctly identified languages (English and German) at the last and second to last row.

[2] https://aws.amazon.com/translate/.

[3] The order, along with further information, is not revealed to protect sensitive company information.

Fig. 3 Histogram of the classes, the y-axis denotes the number of observations and the x-axis the class label



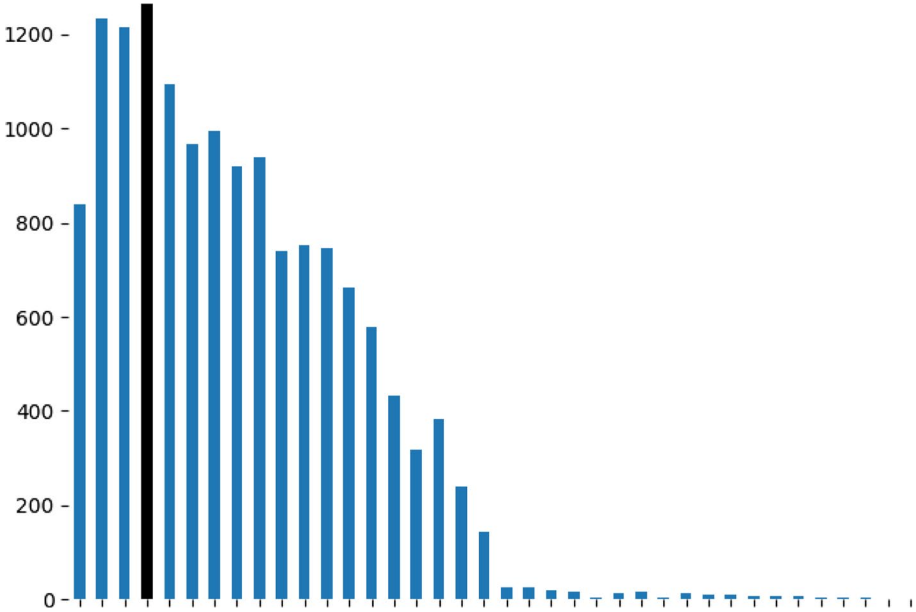


Fig. 4 Number of classes per language. The black bar corresponds to an undefined language (UNK)

3.2 Class distribution

There are 1357 unique classes in the available data, each indicating a subpart that has been identified as being the root cause of a problem in a workshop work order. Class examples are ‘water valve’, ‘yoke’, and ‘oil pump’. The distribution between the classes is heavily imbalanced because some errors and faults are more likely to appear earlier in the life cycle of a truck than others. The histogram in Fig. 3 shows the right-skewed distribution.

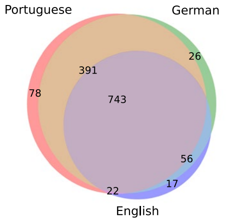


Fig. 5 Overlap of represented classes for the three most frequent languages, Portuguese, German, and English

Table 1 Examples taken from the dataset

Group	Original text	Class
14	110-5002.06 SKARVKOPPLING 6 MM PLAST	4
4	TMI 04 15 02 19	38
8	Noise from rear axle	1227
15	Luftverlust am Fahrersitz	223

The predicted language of the topmost example was Norwegian (“110-5002.06 JOINT COUPLING 6 MM PLASTIC”) and of the last it was German (“Air loss at driver’s seat”)

The class distribution varies among languages and the number of unique classes that are represented in each individual language differs (Fig. 4),[4] with German, English and UNKNOWN (i.e., where the translation fails) being the ones with the most classes. The ten most frequent classes make up 14.8% of total samples, while the one hundred least frequent classes on the other hand make up less than 0.3%. The Venn–diagram in Fig. 5 shows the class overlap between the three most frequent languages, vis. Portuguese, German and English (see Fig. 2). Although we do not limit the number of classes, to those shared by all or certain languages, we note the fact that classes are not represented equally in the languages.

A manual inspection of the data showed that relatively common phrases exist, such as “customer complaint”, “driver complaint”, “attend to”, “vehicle presenting”, which do not provide any information regarding the actual fault. Inspecting to what extent these phrases are present in the data showed that around 7% of the training data comprised at least one of these phrases.[5]

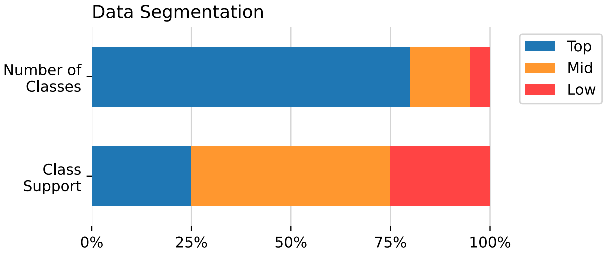
4 Empirical analysis

The task of diagnosing a fault from its textual description can be approached as a large-scale text classification problem. Given that the dataset we experimented with comprises texts in multiple languages, the task is a multilingual large-scale text classification (LSTC)

[4] The diversity of the class distribution across languages is probably due to the different specifications and utilisation of trucks in different markets (hence, languages).

[5] These phrases were investigated in the context of translated data, to avoid the cumbersome work of searching in all possible languages.

Fig. 6 Segmentation based on the number of instances (lowermost), showing the percentage of samples (horizontally) per segment (colored). The same is shown on top when segmenting based on the number of classes (Color figure online)



problem. Opted text classification methods were multilingual or they operated on the data after these were translated into English (ET).

4.1 Benchmarks under the light of translation

Our study aims to draw the baseline performance for this multiclass task, but also to investigate the benefits of employing translation as a service. Hence, we opted for multilingual methods, as well as for methods operating on texts translated into English. Our main multilingual method was XLM-R, a BERT-based model with state-of-the-art performance on multilingual tasks (Conneau et al., 2019). Our main method employing ET was a pre-trained DistilBERT model (Sanh et al., 2019), preferred over the original BERT (Devlin et al., 2018) due to it being more lightweight and having significantly fewer parameters at almost the same classification performance. This helps reduce fine-tuning time (Sanh et al., 2019; Shaheen et al., 2020). We compare two versions of DistilBERT, one fine-tuned on the translated training set and a second one fine-tuned on the original untranslated texts. As a base for these transformers, we use the sequence-classification models provided by the transformers-library for python and initialise them with pre-trained. weights from huggingface[6] We also experimented with a multinomial logistic regression on top of FastText (FTX) embeddings (Joulin et al., 2016) and with a Convolutional Neural Network (CNN). While the FTX was only trained on the translated training data, we based the CNN on a SentencePiece tokenizer trained on all of the available English translations. Although recurrent neural networks work well for the NLP tasks where comprehension of long range semantics is required, CNNs work well where detecting local and position-invariant patterns is required, such as key phrases (Minaee et al., 2021; Wang et al., 2018). Overall, we experimented with six models: FTX-ET, CNN-ET, DistilBERT-ET, DistilBERT, CNN and XLM-R. A majority baseline classification is also used to highlight the task difficulty.

4.2 Evaluation

Our task is a multiclass problem with few classes outweighing thousands of others (see Fig. 3). Some of the rare classes occur in only a few observations, which are not of great interest for our troubleshooting management use case.[7] By contrast, issues related to fewer yet frequently occurring classes can reveal trends that possibly transfer across countries and can be effectively addressed with early troubleshooting management. Hence, for evaluation

[6] DistilBERT: `distilbert-base-uncased` XLM-R: `jplu/tf-xlm-roberta-base.`

[7] Scania owns complementing tools to handle rare classes.

Table 2 Accuracy per model

	Total (%)	Low (%)	Mid (%)	Top (%)
Majority	2.46	0.32	1.03	9.80
FTX-ET	19.6	8.5	22.7	24.5
CNN-ET	49.5	22.8	53.0	68.3
DistilBERT-ET	61.4	35.5	62.2	78.8
DistilBERT	61.1	33.0	64.3	82.3
CNN	52.3	22.8	56.3	72.4
XLM-R	61.3	29.8	66.6	82.0

Classification accuracy of models trained on English translations (ET) and on raw data (lowermost). Accuracy on all classes has been computed, as well as on three clusters formed based on class support

purposes, we opt for top-k accuracy, which counts the number of times the true label is among the k classes predicted with the highest probability. To better present and analyse the results of this large-scale multiclass problem, we introduce support and language based zones, by segmenting the evaluation data based on their class-support and language. We show results with $k = 1$, but Appendix A comprises results with more values, as well as with precision, recall and F1.

Segmentation based on class support is performed at test time by clustering the classes based on their size, and then evaluating per cluster. We used the 1st (25%) and 3rd (75%) quartile as our two thresholds, in order to yield three class zones, shown with the lowermost bar of Fig. 6. The low-support zone (in red) comprises 1,076 classes whose total number of instances (classified in one of those classes) does not exceed the first quartile of our data. The top zone (in blue) is similar but using 27 high-support classes. The mid zone comprises the rest. The top and low support zones comprise the same number of texts, in order to set a scene where low-support classes are of similar interest to management as high-support classes. Segmenting allows us to analyse the effect of class-support on the model performance. We did not segment based on the number of classes (upper horizontal bar in Fig. 6), instead of their support (lower), as this leads to heavily imbalanced zones.

4.3 Experimental results

Table 2 presents the accuracy per model. A majority baseline that achieves a very low score shows the task difficulty, with the top-zone being the easiest due to fewer classes being considered. FTX-ET and CNN perform poorly, but with the latter being clearly better. A DistilBERT fine-tuned on English translations is the best (61.4%), despite the fact that approximately 5% of the instances are missed due to the inability of the translation service to produce a translation. When ignoring these during the evaluation, the accuracy drives up to 61.9%. The multilingual XLM-R follows closely (61.3%), despite the fact that it does not employ any translation service, operating on texts presented with their original language in which they were written. Furthermore, when we fine-tune DistilBERT on all (multilingual) data, the performance improves for mid/top classes while it drops in low. This drop can be explained by the fact that there aren’t enough data to learn during finetuning. By contrast, better-supported classes are better handled without any translation, probably because the model has enough data to learn to trust the terms ignoring the rest.

Table 3 Accuracy per language

Language	#1	#2	#3 EN	#4 UNK	#5	#6	#7	#8	#9	…
DistilBERT-ET	71.3	61.3	**61.0**	–	48.3	52.4	**62.8**	55.3	61.0	…
XLM-R	**72.4**	**61.4**	58.5	**44.3**	**48.5**	**53.2**	60.5	**54.0**	**61.4**	…

Language	#10	#11	#12	#13	#14	#15	#16	#17	#18	#19
DistilBERT-ET	**57.4**	**38.0**	62.4	**57.8**	61.7	**55.1**	46.2	**51.4**	**54.5**	40.8
XLM-R	57.1	36.0	**64.8**	55.5	**62.0**	50.5	**54.6**	46.3	49.7	40.8

Bold values indicate the best model per language

Accuracy (%) per language of the best performing monolingual (DistilBERT-ET) and multilingual (XLM-R) model. Only languages with log frequency above five are considered and DistilBERT is not computed when the language was undefined

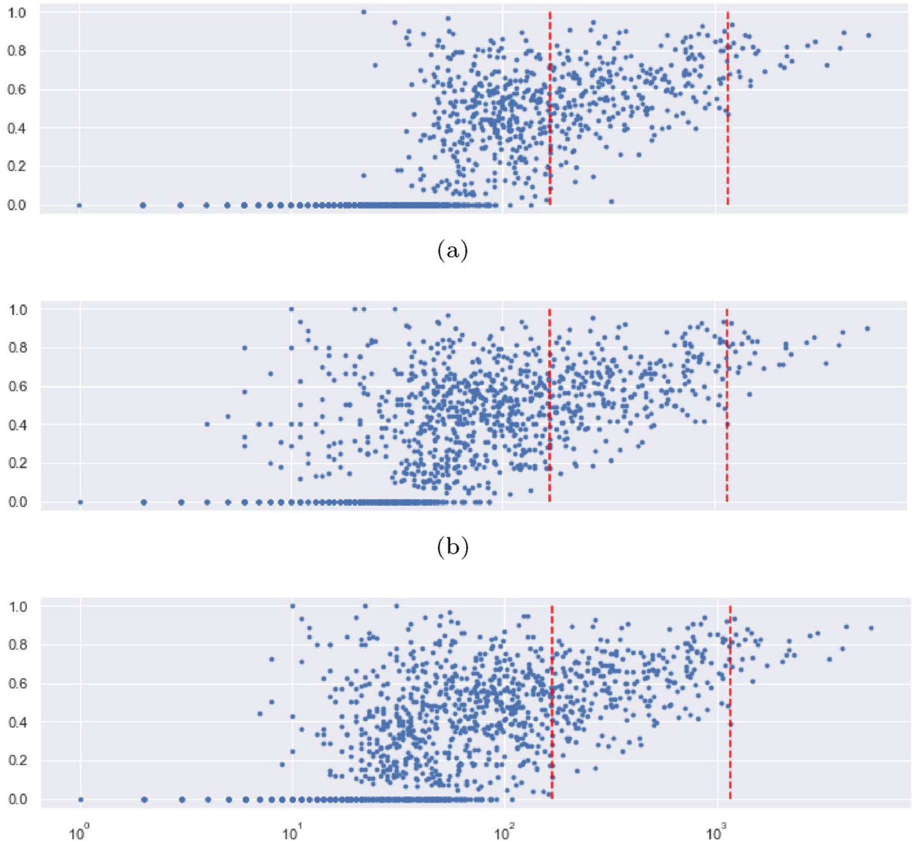
4.3.1 Class-based assessment

The performance per class-support zone is shown in the three rightmost columns of Table 2. CNN performs clearly better than FTX-ET but both fall behind the other models. DistilBERT-ET is better than DistilBERT only in the low-support zone. This means that when the class support is low, this monolingual pre-trained masked language model benefits (+ 2.5) from using English translations as input, instead of the raw data. By contrast, when the class support is higher the translation step is not only redundant but also harms the results in the mid (− 2.1) and in the top zone (− 3.5). For frequently occurring classes (top segment, last column of Table 2), DistilBERT is the best, followed closely by XLM-R (−0.3) and DistilBERT-ET (−3.5). Using the raw input information, by disregarding the language and the translation, appears to provide a better input signal, which is also shown with the superiority of CNN (72.4%) over its translation-based counterpart (68.3%).

4.3.2 Language-based assessment

Table 3 presents the Accuracy (%) per language of the best performing monolingual (DistilBERT-ET) and multilingual (XLM-R) model. Only languages with a log frequency above five are shown, since below that threshold the support significantly drops (Fig. 2). At the same threshold, the number of unique classes is also reduced (Fig. 4).

The five most frequent languages are all better addressed by the multilingual XLM-R except from English (3rd in row), the language DistilBERT is pre-trained on. XLM-R also performs well (44.3%) for texts that the translation service fails to provide an English translation (UNK), which are texts that DistilBERT is incapable of handling. When we use the original texts for samples whose translation is not available, DistilBERT gives 61.4% in total and 37.4%, 63.4% and 81.2% for low, mid and top classes respectively, which means that the low class is improved while mid and top are harmed. DistilBERT, however, is better for the majority of the thirteen less frequent languages with 2.8 units on average, ranging from 0.3 added units for the 10th language to 5.1 for the 17th. This unexpected finding



(a)

(b)

(c)

Fig. 7 XLM-R's F_1-score per class (point) based on class support (horizontally; log10-scaled) when oversampling classes with a support lower than 0 (**a**), 30 (**b**) and 50 (**c**). Red dashed lines separate low (left), mid and top (right) support zones (Color figure online)

shows that a simple lightweight translation-based model, outperforms overall its multilingual counterpart for low-represented languages in this domain.

4.4 Oversampling low-represented classes

To study the possibilities of mitigating the effects and learning issues caused by intrinsic and extrinsic factors through the use of simple methods, we oversampled under-represented classes through random sampling with replacement. Classes with fewer instances than a threshold were augmented by duplicating their instances.[8] We varied this threshold from 10 to 50 and found that oversampling classes with less than 30 instances yields the most benefits for the low-segment (+2) without harming any of the other segments and

[8] Instances were selected randomly, up to twenty per class.

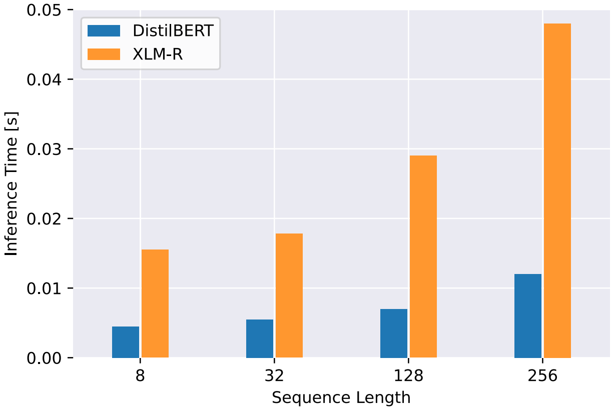


Fig. 8 Inference time for different sequence lengths with a batch size of 1 on an Nvidia T4 GPU

the overall accuracy.[9] Fig. 7 shows XLM-R's F_1-score for each class plotted against its frequency in the training data. When no oversampling is employed (a), in the low-support zone on the left, there is a cut-off point below which classes are incorrectly predicted. When oversampling classes with a support lower than 30 (b) and 50 (c), the same region is more densely populated by classes with high F_1-values. Indicatively, the total accuracy of XLM-R increased with oversampling from 61.3% to 62.5% (threshold equal to 10) to 62.6% (20) to 62.7% (30) to 63.2% (50).

5 Discussion

As was shown in Sect. 4, DistilBERT operating on texts translated to English achieved the best results overall (see Table 2). This approach, however, carries the extra cost of translation. The multilingual XLM-R and the monolingual DistilBERT followed closely. Looking at support zones, DistilBERT is the best for high-frequency classes, outperforming the multilingual XLM-R and the translation-based DistilBERT-ET. The vast amount of data for a limited number of classes, make the original languages a better input space compared to English translations, despite the fact that this method is pre-trained on an English corpus. XLM-R is also left behind for the low-support-zone, outperformed by both DistilBERT models. However, its better performance for the bigger mid zone, makes it the best option overall when translation is not an option. On the other hand, DistilBERT, which is the best in the low and top zones, is also far more lightweight. In specific, the fine-tuning time of XLM-R on an Nvidia T4 GPU is approx. 26 h while that of DistilBERT is 6. Also, as can be seen in Fig. 8, XLM-R is slower during inference.[10] In general, the results show that the workflow shown in Fig. 1 is generally feasible: The results for top-1 accuracy of the tested transformer-based models are already exceeding baselines. If the human expert is presented with a ranking of the 5 most probable causes, the probability of showing the right class again improves substantially (see Table 4). This means that the system is

[9] Besides oversampling, we also added English translations of the texts, but preliminary experiments showed that this approach was overall worse.

[10] The same finding was verified when we used the CPU. The difference was smaller for an Nvidia Tesla P100 GPU.

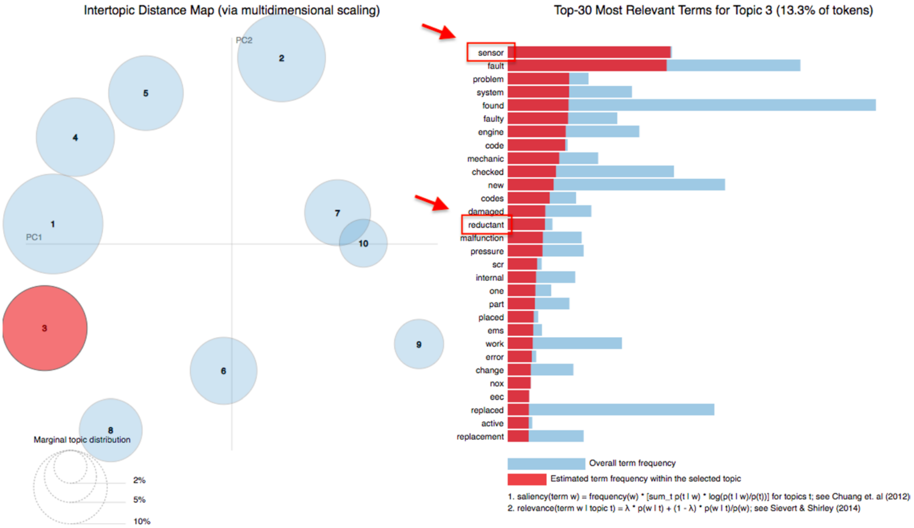


Fig. 9 Topic modeling with Latent Dirichlet Allocation on the validation data. The focus in on the third topic, shown in red. Topics are represented by circles whose size reflects the marginal distribution. The term-frequency is shown in blue bars overall and in red for the topic in question (Color figure online)

generally able to support a person with domain knowledge in making meaningful decisions by suggesting possible causes based on historical data. This is especially useful as the system is able to transfer knowledge from different language domains, that may not be directly accessible by the expert.

5.1 Language-based assessment

As can be seen in Table 3, a multilingual Transformer can work equally well or better than a model operating on texts translated to English. This is important if costs are an important factor while, as already discussed, this comes hand in hand with improved efficiency. Accuracy ranges from above 70% for the most frequent class to around 40% eighteen positions lower in the rank. This score is only five points above the best performing model for the low-zone, which achieved 35.5%. By oversampling the low-supported classes, we observed that better scores are feasible (Fig. 7), even with simple mechanisms, such as instance duplicating.

5.2 Error analysis

An error analysis for the most frequent classes revealed that both our best performing models, DistilBERT (monolingual) and XLM-R (multilingual), perform well, correctly classifying most of the instances of the respective class (the confusion matrices can be found in Appendix B). Although misclassifications are visible in the confusion matrices, these are quite similar for both models, which confused, for example, the following three pairs: (2, 12) that stands for ‘reductant hose’ and ‘NOx sensor’, (140, 159) that stands for ‘sealing ring’ and ‘seal’, and (5, 16) that stands for ‘yoke’ and ‘support’. A linguistic analysis revealed that the classes of the two former pairs concerned the same mechanical malfunction, which means that the prediction could have been even higher if we grouped classes together. In specific, a 'sealing ring' is a circular seal, which can be considered a more specific concept (hyponym) of 'seal'. A 'sealing ring' that is classified as a 'seal' is now considered as a mistake. Also, a 'yoke' is a yoke-shaped plate or such, on which other components can be attached. It can be considered as a subtype of 'support', which is defined as a device that supports and helps hold a unit in a certain position. Finally, a 'NOx sensor' and a 'reductant hose' are two components of the same system, the exhaust gas after-treatment that lies below the fuel and exhaust system. According to a domain expert, the symptoms of a faulty 'NOx sensor' and a 'faulty reductant hose' are similar but not identical. An analysis with topic modeling (Blei et al., 2003), however, revealed that 'sensor' and 'reductant' co-exist in the same topic (Fig. 9).[11]

5.3 Limitations

Due to the sensibility of the used data with regard to Scania's economic interests, we are not able to release the data to the public. Also, we only explored the technical feasibility of the proposed workflow. In order to assess the realistic overall gain from the workflow, however, we note that further studies are required. Besides these two limitations, we also discuss three more:

Time locality Errors and faults are more likely to appear earlier in the life cycle of a truck compared to others. More generally, certain faults might cluster in time, which means that their respective claims will not be independent. For example, if component A brakes in Truck X, perhaps component B is more likely to break next. In this work, we ignored "time locality", which we plan to investigate further in the future, along with its implications to training and evaluating machine learning models.

Non-optimized unilingual model Our application of DistilBERT assisted in the exploration of the proposed framework's resource efficiency. At the same time, however, we note that it may underperform compared to XLM-R. Future research should explore whether a non-optimized unilingual model, such as native BERT, can meet the performance of the multilingual XLM-R.

Metadata We only considered the text of the problem or the symptom, in this study, ignoring any metadata. We note, however, that predictive power may exist in the metadata in the issue management system. The location, the vehicle model, the time of the year, etc. can assist the classification and improve the performance. The assessment of the value of using metadata as input for automotive fault nowcasting is a research direction that should be explored in future work.

6 Conclusion

We presented the first large-scale study that demonstrates the applicability of automated fault report management in the automotive domain and show a possible workflow for applying such a system in an automotive industry scenario. This study showed

[11] We used: https://pyldavis.readthedocs.io, and the following LDA implementation by Gensim: https://radimrehurek.com/gensim/models/ldamodel.html.

that the textual descriptions of symptoms can be used to early diagnose the root causes and hence facilitate effectiveness, efficiency and decision-making of the automotive industry and management therein. Empirical findings, using data from vehicle fleets, revealed that Transformer-based models can adequately address this large-scale multilingual multiclass text classification task, opening the way for the application of the same workflow on similar domains. Our findings show that translation-based data assist low-represented classes while, when translation is not an option, using a model pre-trained on multilingual data or fine-tuning a model pre-trained on English data can perform equally well, or even better for high-frequency classes. Also, the relatively low performance of low-frequency classes can be improved with oversampling, a direction we plan to investigate further in future work, along with hierarchical classification and the exploitation of more (labeled and unlabeled) data. Further directions for future work comprise the study of time locality, the assessment of metadata as input, and the use of conformal prediction to assist the human expert. Also, although the focus of this study is within the automotive domain, the applicability of the proposed method goes beyond, concerning any troubleshooting point where fault claims are received in textual form. Therefore, a final direction for future work is the application to other areas, where results are of high value for all actors with a need for the diagnosis of complex systems, especially when the actor is global. In addition to heavy vehicles, this comprises the light vehicle industry, rail-based vehicle industry, process industry, defence industry, and consumer products.

Appendix A: Evaluation

Table 4 presents the top-k categorical Accuracy of XLM-R and DistilBERT, which considers a prediction as true if the correct class is within the top-k predictions. Experimenting with k equal to three and five, XLM-R is better in both.

Tables 5 and 6 show the Precision, Recall, and F1 scores of XLM-R and DistilBERT, macro-averaged per zone. XLM-R has a better F1 score than DistilBERT for top and mid classes, due to its better Recall. However, in low classes DistilBERT is superior in all metrics, lifting also the overall performance (1st column of Table 6).

Table 4 Top-3 and Top-5 categorical Accuracy of XLM-R and DistilBERT

	DistilBERT (%)	XLM-R (%)
Top-3 accuracy	75.4	**77.9**
Top-5 accuracy	79.2	**82.6**

Bold values indicate the best model per row

Table 5 Precision, Recall and F1 of XLM-R

	Total (%)	Top (%)	Mid (%)	Low (%)
Precision	28.4	74.5	57.9	20.4
Recall	26.4	**80.2**	**63.0**	16.6
F1	25.4	**77.0**	**59.3**	16.2

Precision, Recall and F1 of XLM-R macro-averaged per zone. In bold the best results compared to the ones of DistilBERT in Table 6

Table 6 Precision, Recall and F1 of DistilBERT

	Total (%)	Top (%)	Mid (%)	Low (%)
Precision	**44.0**	**74.6**	**60.0**	**39.7**
Recall	**35.6**	79.4	60.1	**29.0**
F1	**36.4**	76.5	58.5	**30.5**

Precision, Recall and F1 of DistilBERT macro-averaged per zone. In bold the best results compared to ones of XLM-R in Table 5

Appendix B: Confusion matrices

Tables 7 and 8 below show the confusion matrices for the DistilBERT model and the XLM-R model for the ten most frequent classes. The high values on the diagonal of the matrices imply that both models predict the top 10 classes well.

		Predicted Class									
		5	**16**	**12**	**2**	**0**	**41**	**159**	**140**	**64**	**90**
True Class	**5**	2154	88	0	2	1	2	0	1	0	1
	16	113	1534	0	0	0	0	0	0	0	1
	12	0	0	1374	53	0	3	0	0	0	0
	2	0	0	41	1181	0	2	0	0	0	0
	0	1	0	1	1	1126	1	0	0	0	0
	41	2	0	0	2	0	1081	0	0	2	0
	159	1	1	0	1	0	0	735	128	0	1
	140	4	0	1	1	0	2	161	762	0	1
	64	10	1	0	0	0	0	0	2	890	0
	90	2	1	0	1	0	0	1	2	0	645

Table 7 Confusion matrix for the top-10 classes for DistilBERT

Predicted Class

True Class	5	16	12	2	0	41	159	140	64	90
5	2305	67	0	0	1	2	0	2	2	0
16	159	1595	0	2	0	0	0	0	1	1
12	0	1	1487	42	2	2	0	0	0	0
2	0	0	94	1216	0	1	0	0	0	0
0	0	0	1	1	1134	0	0	0	0	0
41	2	0	0	2	0	1083	0	0	1	0
159	1	0	0	1	0	0	794	117	1	0
140	7	1	1	0	0	2	184	790	1	1
64	12	2	0	0	0	1	0	1	906	0
90	3	0	0	0	0	2	0	1	0	660

Table 8 Confusion matrix for the top-10 classes for XLM-R

Author contributions JP wrote the first draft and undertook the initial experiments; AR and JC undertook extensive experiments and co-authored; OS, TL, and MB supervised the experiments and co-authored; KR contributed with the presentation and co-authoring.

Funding Open access funding provided by Stockholm University. Funding for this research has been provided by the European Union's Horizon Europe research and innovation programme EFRA (Grant Agreement Number 101093026). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Commission-EU. Neither the European Union nor the granting authority can be held responsible for them.

Data availability The data are not publicly available.

Code availability The code is not publicly available.

Declarations

Conflict of interest The authors have extensively collaborated with the following institutions within the past 3 years: John Pavlopoulos: Department of Computer & Systems Sciences, Stockholm University [su.se], Department of Informatics, Athens University of Economics and Business [aueb.gr], Google [google.com], Ca'Foscari [unive.it]; Alv Romell: Department of Computer Science, Lund University [lth.se], Strategic Product Planning and Advanced Analytics, Scania [scania.com]; Jacob Curman Department of Computer Science, Lund University [lth.se], Strategic Product Planning and Advanced Analytics, Scania [scania.com]; Olof Steinert Strategic Product Planning and Advanced Analytics, Scania [scania.com]; Tony Lindgren Department of Computer & Systems Sciences, Stockholm University [su.se], Strategic Product Planning and Advanced Analytics, Scania [scania.com]; Markus Borg Department of Computer Science, Lund University [lth.se]; Korbinian Randl Department of Computer & Systems Sciences, Stockholm University [su.se]

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethics approval Not applicable.

Ethical consideration The classifiers could in principle be used to assist the compilation of false claims. The class labels, however, are encoded and any released models classify indices and not class names.

References

Adamopoulou, E. & Moussiades, L. (2020). An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations* (pp. 373–383). Springer.

Aktas, E. U., & Yilmaz, C. (2020). Automated issue assignment: Results and insights from an industrial case. *Empirical Software Engineering, 25*(5), 3544–3589.

Biteus, J., & Lindgren, T. (2017). Planning flexible maintenance for heavy trucks using machine learning models, constraint programming, and route optimization. *SAE International Journal of Materials and Manufacturing, 10*(3), 306–315.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*(Jan), 993–1022.

Borg, M. & Runeson, P. (2014) Changes, evolution, and bugs: Recommendation systems for issue management. In *Recommendation systems in software engineering* (pp. 477–509). Springer.

Borsci, S., Malizia, A., Schmettow, M., Van Der Velde, F., Tariverdiyeva, G., Balaji, D., & Chamberlain, A. (2022). The chatbot usability scale: The design and pilot of a usability scale for interaction with ai-based conversational agents. *Personal and Ubiquitous Computing, 26*(1), 95–119.

Carvalho, T. P., Soares, F. A., Vita, R., Francisco, R., Basto, J. P., & Alcalá, S. G. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers & Industrial Engineering, 137*, 106024.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L. & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint* arXiv:1911.02116

Devlin, J., Chang, M. -W., Lee, K. & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint* arXiv:1810.04805

Fink, O., Wang, Q., Svensen, M., Dersin, P., Lee, W.-J., & Ducoffe, M. (2020). Potential, challenges and future directions for deep learning in prognostics and health management applications. *Engineering Applications of Artificial Intelligence, 92*, 103678.

Irving, J., Patel, R., Oliver, D., Colling, C., Pritchard, M., Broadbent, M., Baldwin, H., Stahl, D., Stewart, R., & Fusar-Poli, P. (2021). Using natural language processing on electronic health records to enhance detection and prediction of psychosis risk. *Schizophrenia Bulletin, 47*(2), 405–414.

Izquierdo, J. L., Ancochea, J., Soriano, J. B., Savana COVID-19 Research Group. (2020). Clinical characteristics and prognostic factors for intensive care unit admission of patients with covid-19: Retrospective study using machine learning and natural language processing. *Journal of Medical Internet Research, 22*(10), 1801.

Jonsson, L., Borg, M., Broman, D., Sandahl, K., Eldh, S., & Runeson, P. (2016). Automated bug assignment: Ensemble-based machine learning in large scale industrial contexts. *Empirical Software Engineering, 21*(4), 1533–1578.

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint* arXiv:1607.01759

Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys (CSUR), 54*(3), 1–40.

Nath, A. G., Udmale, S. S., & Singh, S. K. (2021). Role of artificial intelligence in rotor fault diagnosis: A comprehensive review. *Artificial Intelligence Review, 54*(4), 2609–2668.

Qian, C., Zhu, J., Shen, Y., Jiang, Q. & Zhang, Q. (2022). Deep transfer learning in mechanical intelligent fault diagnosis: Application and challenge. *Neural Processing Letters* 1–23.

Safaeipour, H., Forouzanfar, M., & Casavola, A. (2021). A survey and classification of incipient fault diagnosis approaches. *Journal of Process Control, 97*, 1–16.

Sanh, V., Debut, L., Chaumond, J. & Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint* arXiv:1910.01108

Shaheen, Z., Wohlgenannt, G. & Filtz, E. (2020) Large scale legal text classification using transformer models. *arXiv preprint* arXiv:2010.12871

Theissler, A., Pérez-Velázquez, J., Kettelgerdes, M., & Elger, G. (2021). Predictive maintenance enabled by machine learning: Use cases and challenges in the automotive industry. *Reliability Engineering & System Safety, 215*, 107864.

Thorne, C. (2017). Chatbots for troubleshooting: A survey. *Language and Linguistics Compass, 11*(10), 12253.

Vaish, R., Dwivedi, U., Tewari, S., & Tripathi, S. M. (2021). Machine learning applications in power system fault diagnosis: Research advancements and perspectives. *Engineering Applications of Artificial Intelligence, 106*, 104504.

Wang, W. & Gang, J. (2018). Application of convolutional neural network in natural language processing. In *2018 international conference on information systems and computer aided education (ICISCAE)* (pp. 64–70). https://doi.org/10.1109/ICISCAE.2018.8666928

Zhang, T., Chen, J., Li, F., Zhang, K., Lv, H., He, S., & Xu, E. (2022). Intelligent fault diagnosis of machines with small & imbalanced data: A state-of-the-art review and possible extensions. *ISA Transactions, 119*, 152–171.

Zhao, Z., Wu, J., Li, T., Sun, C., Yan, R., & Chen, X. (2021). Challenges and opportunities of ai-enabled monitoring, diagnosis & prognosis: A review. *Chinese Journal of Mechanical Engineering, 34*(1), 1–29.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

John Pavlopoulos[1] · Alv Romell[3] · Jacob Curman[3] · Olof Steinert[2] · Tony Lindgren[1,2] · Markus Borg[3] · Korbinian Randl[1]

✉ John Pavlopoulos
ioannis@dsv.su.se

Alv Romell
alvromell@gmail.com

Jacob Curman
curmanjacob@gmail.com

Olof Steinert
olof.steinert@scania.com

Tony Lindgren
tony@dsv.su.se

Markus Borg
markus.borg@cs.lth.se

Korbinian Randl
korbinian.randl@dsv.su.se

[1] Department of Computer and Systems Sciences, Stockholm University, Stockholm, Sweden

[2] Strategic Product Planning and Advanced Analytics, Scania CV, Södertälje, Sweden

[3] Department of Computer Science, Lund University, Lund, Sweden