Journal of Intelligent Manufacturing Systems manuscript No.
(will be inserted by the editor)

Discovering Critical KPI Factors from Natural Language in Maintenance Work Orders

Madhusudanan Navinchandran · Michael E. Sharp · Michael P. Brundage · Thurston B. Sexton

Received: date / Accepted: date

Abstract Optimizing maintenance practices is a continuous process that must take into account the evolving state of the equipment, resources, workers, and more. To help streamline this process, facilities need a concise procedure for identifying critical tasks and assets that have major impact on the performance of maintenance activities. This work provides a process for making data investigations more effective by discovering influential equipment, actions, and other environmental factors from tacit knowledge within maintenance documents and reports. Traditional application of text analysis focuses on prediction and modeling of system state directly. Variation in domain data, quality, and managerial expectations prevent the creation of a generic method to do this with real industrial data. Instead, text analysis techniques can be applied to discover key factors within a system, which function as indicators for further, in-depth analysis. These factors can point investigators where to find good or bad behaviors, but do not explicitly perform any anomaly detection. This paper details an adaptable procedure tailored to maintenance and industrial settings for determining important named entities within natural language documents. The procedure in this paper utilizes natural language processing (NLP) techniques to extract these terms or concepts from maintenance work orders and measure their influence on Key Performance Indicators (KPIs) as defined by managers and decision makers. We present a case study to demonstrate the developed workflow (algorithmic procedure) to identify terms associated with concepts or systems which have strong relationships with a selected KPI, such as time or cost. This proof of concept uses the length of time a Maintenance Work Order (MWO) remains open from creation to completion as the relevant performance indicator. By identifying tasks, assets, and environments that have significant relevance to KPIs, planners and decision makers can more easily direct investigations to identify problem areas within a facility, better allocate resources, and guide more effective analysis for both monitoring and improving a facility. The output of the analysis workflow presented in this paper is not intended as a direct indicator of good or bad practices and assets, but instead is intended to be used to help direct and improve the effectiveness of investigations determining those. This workflow provides a preparatory investigation that both conditions the data, helps guide investigators into more productive and effective investigations of the latent information contained in human generated work logs, specifically the natural language recorded in MWOs. When this information preparing and gathering procedure is used in conjunction with other tacit knowledge or analysis tools it gives a more full picture of the efficiency and effectiveness of maintenance strategies. When properly applied, this methodology can identify pain points, highlight anomalous patterns, or verify expected outcomes of a facility's maintenance strategy.

Keywords Maintenance · Key Performance Indicators · Natural Language Processing · Intelligence Augmentation · Artificial Intelligence

Systems Integration Division
Engineering Laboratory
National Institute of Standards and Technology
Gaithersburg MD USA
E-mail: michael.sharp@nist.gov

1 Introduction

Optimizing and evaluating a facility's maintenance strategy (e.g., run-to-failure vs. preventive vs predictive strategies) is crucial, and to be done properly, must incorporate information from a wide array of sources. Without periodic evaluation, even the best-made maintenance procedures and policies can become outdated and obsolete. Investigating the conditions around resource or time intensive activities can yield valuable insight into the efficacy of any enacted procedures or prescriptive tasks [14, 15].

Artificial Intelligence (AI) technologies have given cause for reevaluating previously underutilized sources of information, particularly natural language text. Manually entered data is prevalent in maintenance, but has classically been underutilized or excluded from many computer analysis methods due to the difficulty in processing. Even a small facility will have thousands of Maintenance Work Orders (MWOs) addressing tasks from the routine to the unexpected and unique. The often erratic nature of information found in MWOs includes unstructured technical-jargon, incorrect grammar, inconsistent formats, and missing field entries [31]. We provide some example MWOs in Table 1.

Such issues have traditionally been barriers for developing computer-automated means of extracting both the explicit and implicit knowledge contained within them. However, if this data can be processed, it can be used to find information about what machines frequently need maintenance, the severity of failures, parts or procedures most frequently cited, and a plethora of other information associated with logistics, time, workers, assets, and resources.

To be useful to manufacturers, this information must be extracted for actionable decision support. This step requires performing analyses tailored to the goals of the decision makers and availability of information. In practice that means adapting any analysis to focus on a chosen performance indicator and all the factors that affect it. For example, if each MWO had a total 'cost' or dollar amount spent on assets and labor associated with it, then a decision maker may want to know which assets have the strongest relationship with cost, or what factors around a task have the most influence on cost. This analysis could directly identify costly tasks, specific tools that help ensure lower job costs, or even if some tasks have a tendency of unexpected additional costs.

The directed analysis workflow described in this paper presents a method to extract important terms and concepts from a MWO and constructs an explanatory model to relate these terms and concepts to a Key Performance Indicator (KPI). Analysts can then rank the concepts based on their relationship with the KPI and relay that information to decision makers. MWOs are particularly suited to this task because they hold a variety of information related to many decision support relevant metrics, values, and performance indicators. Some examples include: the cost of a maintenance procedure, useful in helping prioritize procedures based on Return on Investment (ROI); time requirements for a procedure, useful in both scheduling and to help compare equivalent procedures for inefficiencies; determination of problem tools, systems, or procedures, useful to help guide revisions to protocols, resource allocation, and other logistic strategies. As a proof of concept, the procedure described here is demonstrated on real MWO data to identify terms relating to systems and concepts that have a strong relationship with the amount of time a MWO remains open after initial creation. Because each facility and management team will have different needs, customization options for the analysis are addressed in the methodology section of this paper.

This paper provides guidance in performing analyses on sources of natural language such as MWOs for identifying significant relationships with maintenance strategy Key Performance Indicators (KPIs). The contribution of this work is to create a high level analysis workflow for analyzing real, industry MWO data. Most research in the literature addresses part (or parts) of this workflow, but fails to provide an in depth discussion on how to tie the pieces together. This workflow does not replace previous research, but instead enables these tools to be swapped and used together to best convert the raw data of industrial MWOs into actionable intelligence.

2 Background and Literature Review

Machinery maintenance is an expensive proposition for most manufacturers, costing an estimated \$ 57.3 billion in 2016 [38]. Manufacturers relying on reactive maintenance had 3.3 times more downtime, 16 times more defects, and 2.8 times more lost sales than those that relied on more advanced maintenance strategies [38].

2.1 Maintenance Strategy Analysis

Implementing proactive maintenance strategies, such as preventive and predictive maintenance, is important to improving manufacturing performance [36] and improving equipment and system longevity. These strategies provide decreased downtime and increased productivity resulting in a strong return on investment. A number of works provide an in depth view into different advanced maintenance strategies [5, 24, 26, 37]. However,

Table 1 Example Maintenance Work Orders

Asset ID	Problem	Open	Closed	Remarks
162545	HP and LP pumps INOP	2/09/07 07:57	2/13/07 06:23	Checked / No Problem Found
150428	Broken door clamp -hook bolt	2/09/07 08:34	2/11/07 13:19	camera ordered. Delivery 7/14
156997	St#5 motor inop/humming	-/--/-- --:--	-/--/-- 10:22	camera ordered. Delivery 7/14
150428	Saw blade spun on hub	2/12/07 06:12	2/11/07 13:52	
150428	Speed limit @ Spindle A exceeded	2/12/07 08:27	2/12/07 --:--	Complete
164243	Broken chain on loader	2/12/07 09:49	-/--/-- --:--	
156551	Encoder coupling broken	2/12/07 --:--	2/12/07 13:35	Remove Vacuum Plug
150428	Emergency retract solonoid failure	2/12/07 13:45	2/24/07 13:45	Replaced Spray Nozzles

despite the known opportunity gain for advanced maintenance, many companies struggle to implement proactive maintenance strategies in their organizations due to the persistent barriers of high cost, labor investments, lack of in-house expertise, and lack of general guidance towards implementing maintenance strategies correctly [14].

Some companies are also daunted by the shear volume of the problem and are unsure where best to apply these strategies in their facilities. While there have been successes in implementing advanced maintenance strategies in some companies, guidance and guidelines are needed to achieve success more broadly across industry. This includes identification of not only which strategies are right for a facility, but prioritizing where and how to implement them.

A major barrier to uncovering this knowledge is that much of the information about a facilities maintenance and reliability information is contained in semi-structures or erratic data sources, such as maintenance requests or job logs. Previous work has sought to address the inconsistencies in reliability information structuring and utilization within a manufacturing facility [32, 40], but to date there are no consistent cross-industry guides for recording and analyzing the myriad of unstructured information sources within an industrial facility. Some current research focuses on directly analyzing MWOs to help manufacturers evaluate and advance their maintenance strategies, but less research addresses the selection and prioritization of where and how to implement them. Most manufacturers have access to MWOs, which contain the health history of different assets on the floor, action logs, and other related information about performance and health of a facility. Analyzing these MWOs can capture this information and help provide tailored guidance on implementation and use of advanced maintenance strategies to more manufacturers. This work provides a guideline to structure and process this data in a reproducible way so manufacturers can better uncover this information and focus individual analyses tailored to the goals of the facility.

2.2 Structured MWO Analysis

MWOs are notoriously difficult for commercial off-the-shelf products to read, interpret, and analyze due to the often highly individualized nature of the syntax and unique word choice that is created within facilities. Sipos et al. [34] were able to demonstrate the viability of equipment failure models built from a corpus of event logs in critical medical devices. However, that particular investigation had the luxury of using *event codes* which rarely exist in such a standardized or consistent fashion in manufacturing maintenance data sources. The most ubiquitous information source an analysts can count on is free form text. Free form text in MWOs is prone to misspellings, unique abbreviations, short hand (i.e., site specific jargon), and inaccuracies from typos, misrepresentations, and misunderstandings from the human operators [11, 31].

2.3 Unstructured Text Analysis

Multiple efforts towards utilizing the unstructured sources of information in MWOs are being pursued in both the maintenance and manufacturing domains. Examples include developing diagnostic fault trees from historic maintenance logs[20] and fusing data from historical maintenance data with standard malfunction codes to identify past and ongoing trends in maintenance[18]. This research helped verify associations between recorded codes and problems aiding in only relevant data being used in additional analyses. The data analyzed from that research contained standardized malfunction codes, which removed the difficult commonly faced from ambiguity in language commonly found in free form natural language. Other work that did focus on natural language was performed by Bokinsky et al. They used natural language analysis on Maintenance Action Forms and compared the actual maintenance action against that listed in a manual to check if best practices are being followed, reducing out-of-service times for aircraft[1]. Although each of these research efforts look into maintenance analysis, none address the relation of cost or time to maintenance actions or have other tailored decision-making criteria analyses.

One method that has been successful in providing insights into maintenance problems in manufacturing, and similar domains, is the analysis of free-form text found in MWOs and associated values with highly specialized tools. Because of the eccentricities of human derived text, most natural language algorithms require significantly more training exemplars of text entries to learn such a specific dialect than are available at most facilities, especially the small to medium sized enterprises (SMEs) [27]. This challenge along with other barriers have made in-depth analysis of MWOs require large investments of effort compared to the expected return. This has led researchers to study and apply various Natural Language Processing (NLP) techniques to help extract and contextualize information to improve maintenance decision making [33, 29, 11, 3, 17]. In support of that, Sharp et al. and Sexton et al. created a procedure to efficiently clean and annotate MWO short text for use in analysis [33, 29]. This work was used to develop conceptual maintenance KPIs in [3], however, this paper stopped short of showing how to calculate these KPIs. Lukens et al. described a procedure for capturing the data quality of such metrics given the inaccuracies in the MWO data in [17]. Some work has been done to show how the data from MWOs can be used to determine reliability from various assets, however no formal procedure for generalizing this analysis is discussed [11, 30].

Much of the previous work aims to mitigate the problems unique to MWO style documents with respect to cleaning and contextualizing of MWOs, namely: brief short-form text entries, high levels of misspellings and jargon, and relatively small amounts of unique exemplars (1000s to 100,000s of data points). However, while these works aid in the preparation for analysis, there are few works which address the formal steps of correlating MWO elements with performance indicating factors, such as cost or time.

2.4 NLP tools for Analysis

Much of the work described in this paper relies on tools such as Word2Vec [19], Bag of Words [39], and others. None of the specific software or algorithms mentioned are necessarily being endorsed as the best for a given application, but are simply put forth as examples of the style of tools that could fulfill the requirements of the analysis. Introductions mentioning such algorithms can be quickly found online at public websites, making them a typical choice for practitioners. The core function of such algorithms gives a numeric representation to words, ideas, and concepts within free form text. This allows for automated contextualization and processing via statistical modeling tools. Once a meaningful representation of the information within each MWO exists, more generic sets of explanatory modeling tools can be used to identify or utilize found relationships.

Work presented in this paper relies on the notion that the relative ability of a text based term or concept to predict some target value or KPI relates directly to the strength of the relationship between that concept and the KPI. Terms and concepts used as input in this analysis can be direct words and phrases as well as any piece of information collected and recorded on the MWO, such as physical asset name/type, designations of actions taken, tools/resources used, etc. Through analysis, the relative effect each input element has can be evaluated and ranked, giving a loose interpretation of their 'importance' to the system. As presented in this work, importance is strongly analogous to sensitivity as understood from sensitivity analysis [25]. The use of 'importance' or 'importance values' in this document is a relaxed form of the formal definition intended to fit the more colloquial and intuitive expectations of the word. It is loosely used as the amount that any concept or term used as input helps improve the ability of a model to predict or classify the selected KPI. Some examples of measures of this include expected variance reduction, information entropy measures, and other similar measures. The particular formal measure of influence or importance can be tailored to the particular needs of the practitioner, but the general concepts of preparing, ranking, and interpreting those values remain the same.

This paper describes a workflow for taking the raw data from MWOs and determining important features within the data. This procedure allows analysts to use various models to tailor their analysis for their facility. Previous literature has explored in depth the different techniques and tools to help improve portions of this workflow. The intent of this work is to bring these pieces together to analyze real industrial data in a reproducible and understandable way. The goal of this work is not to further develop novel algorithms or technologies for analyzing MWOs, but to establish a standard best practice for structuring analyses for maximum gain with currently available and easily obtainable tools.

3 Analysis Workflow

This section highlights the process of extracting and identifying important text based references to concepts or assets from MWOs. These can be analyzed in the context of any KPI. The primary process resembles an input sensitivity analysis, but is tailored to the context of MWO and applies domain knowledge. More generally it is the quantification of amount of information related to the KPI imparted by a given piece of captured text. A schematic is provided in Fig. 1.

3.1 Preprocessing

Preprocessing MWOs has 2 primary steps: 1) the selection of a KPI, such as hours spent or cost, and then the 2) digitization and standardization of the data contained in the MWOs to a form that can be processed by the models and analysis algorithms. This second step includes managing and cleaning both numeric and extracted language information.

Select KPI of Interest

The first step is to select some performance indicator associated with the MWO on which to base any resulting analysis. When selecting the performance indicators for analysis, it is necessary that each MWO have a corresponding value. These need not be directly in the MWO itself, but can come from an external source. To be most effective, the performance indicators need to be represented on some ordinal scale, either numeric or categorical. The selection of the performance indicator will greatly affect the interpretation of the resulting analysis by becoming the basis for all importance values. Because the goal of this workflow is to identify broad trends and relationships, it is okay if these indicators are more qualitative or difficult to accurately assign. The mechanisms of the workflow allow for low accuracy to be overcome through enough examples.

The final step ensures that the MWO is able to be read by a computer, then interpreted in an unambiguous way. The challenges associated with both the natural language of MWO entries and human-based data collection are not trivial and should not be ignored. In order to prepare the raw field entries of the MWOs, basic cleaning and text clarification is necessary in order to standardize and structure the free-form language and any nonstandard numeric values.

Clean & Standardize Data

This often erratic nature of human data entry presents a significant challenge for interpreting the numeric fields of MWOs [3]. Impossible or unrealistic values often result from misplaced values (i.e., values recorded in the wrong place), misinterpreted field headings, missing values, or other misrepresentations and erroneous entry of the information [17]. Even the more mundane challenge of inconsistent formats, which can largely be addressed with standard algorithms and tools, still can produce values that are incomprehensible to the computer if there are not strict requirements put on the data entry [11].

Identifying incomprehensible or infeasible values, such as negative MWO completion times, is the first step in rectifying any numeric field entries. This step can generally be accomplished with a basic set of logical gates set to screen out missing or undesirable values within the data set. Where applicable, it is possible to additionally apply some basic rules to aid or undo common errors. For example, when logging temporal data, a common mistake may be to switch the entries for start time and completions time, yielding a negative duration of the logged task. An intuitive rule to ensure such data is not lost would be to take the absolute value of any calculated task duration. Another common rule for rectifying the length of time recorded for a task might be to impose a lower bound on calculated values, such as allow no value to be below 5 minutes. These rules could be as complicated or simple as deemed necessary, and do not need to capture 100 % of the related data points to be useful. If any information about the the ranges, limits, or physically sensible values, etc. exists that relates to the numeric value being rectified, this should be employed to design and formulate corrective rules.

If needed, human interpretation or estimation of missing critical values can be used to help expand the number of usable examples within an MWO data set. Algorithmic estimation or imputation of missing values is not generally recommended for this procedure because if done incorrectly it could have undesired effects on the ultimate sensitivity and influence analysis that is integral to the goal of this work. If such methods are implemented, it is strongly recommended that some form of semi-supervised learning method is applied with a human in the loop to verify output and mitigate or prevent bias.

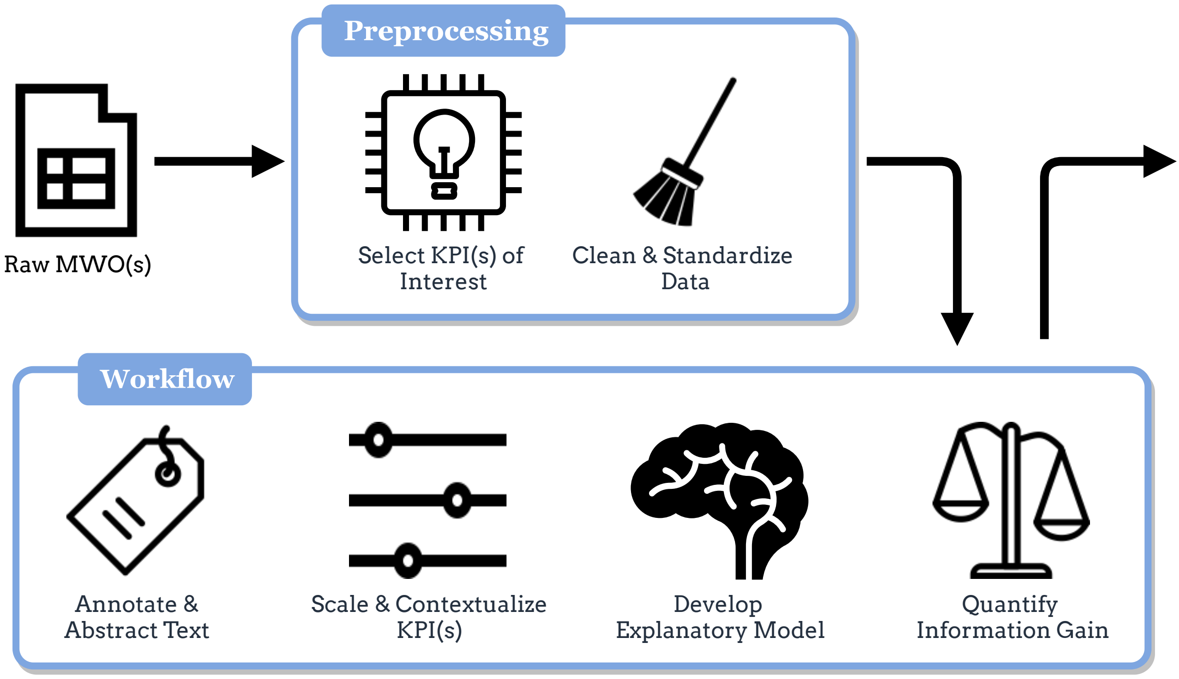
Preprocessing
Raw MWO(s)
Select KPI(s) of Interest
Clean & Standardize Data
Interpreted Results
Workflow
Annotate & Abstract Text
Scale & Contextualize KPI(s)
Develop Explanatory Model
Quantify Information Gain

Fig. 1 A schematic of the analysis workflow. Different analysis methods can be applied at each step.

The main goal in preprocessing text within MWOs is to condense single- or multi-word phrases that describe a unique action, concept, or asset into one unified representation [29, 33]. This entails cleaning any typos, aliasing any alternate versions or abbreviations, and contextualizing any ambiguities of repeated language such that each identified element represents a single expressive entity.

Natural language found within MWOs can come with a variety of structures and potential errors formed during input. For example, the MWO description from an actual manufacturing MWO, *'Hi Pressure coolant faults'* has a nonstandard spelling for *high* and is not a complete English sentence. Similarly, *'60/40 lo presure'* has nonstandard spellings for *low* and *pressure* and also contains a domain specific term that has no interpretable meaning outside the context of the work order with the generating facility. As a final example, *'Station #3 incomming conveyor fault'* has both incorrect spelling and an organization specific location.

Standard commercial and open source tools for cleaning text have a difficult time due to the unique, yet often contextually correct, eccentricities found in the language of MWOs. Standard tools for spell checking and extracting the base form of words (i.e., stemming) [16] will often misinterpret or misidentify site specific lexicon, which can lead to incorrect conclusions of models of relationships within the MWOs. *NESTOR*[1] is an open source tool designed with these limitations in mind. It has been developed by the National Institute of Standards and Technology (NIST) to clean and annotate eccentric text such as found in MWOs [29, 28]. The tool presents a domain expert with a list of similarly identified words to quickly sort related words into a single representative alias. This single representative can then be classified as a *problem*, *solution* or *item* in the context of that MWO.

The context categories within this work were chosen because of their intuitive applicability to specific decision making tasks. For instance, high impact *items* can help drive inventory management decisions for component inventory management (e.g., gears, ball bearings) or impact scheduling decisions of maintenance resources (e.g., moving gantries for maintenance tasks). *Solutions* of high importance can be used to create new standard operating procedures if the solution occurs frequently and have high impact on time duration (e.g., creating a standard operating procedure around inspecting a particular asset to reduce completion time variability). Lastly, investigating *problems* can direct sensing and monitoring needs (e.g., monitoring flow for hydraulic leaks). This paper suggests this method and other methods of contextualization to aid in capturing and directly capitalizing on tacit knowledge both from the operators and analysts captured in the MWOs. Annotating of aliased elements with context categories in this manner is not strictly necessary, but can provide stronger models and more insightful interpretations through analysis.

[1] https://www.nist.gov/services-resources/software/nestor

This procedure will be discussed in greater detail as part of the primary process later in this section.

3.2 Main Workflow Procedure

The primary procedure for developing influence analysis can roughly be summarized as 1) annotate and abstract the concepts from the natural text, 2) scale and contextualize an explanatory KPI in preparation, 3) construct an explanatory or predictive model, upon which the analyst will 4) perform a sensitivity-type analysis to infer influence and importance of the identified concepts so that they can finally 5) interpret the results in context of the facility.

As part of these listed steps, the procedure also will address a number of sub-tasks that will be expanded upon in the respective following sections. The first is imparting domain context via annotating and tagging text elements with pertinent classifications. Next, any data imbalances need to be identified and managed. Performance indicators need to be represented as a format that is both informative and conducive to any predictive model. Select the architecture of that model to both best suit that data and the end goals of the analysis. Finally, the developed model is tested for the strength of the relationship each of the input MWO elements has to the KPI so that critical and important factors can be identified and interpreted by an analyst. This section explores each of these stages, and makes some general observations and recommendations for specific implementations of this generic workflow.

Annotate & Abstract Text

The manual abstraction of text elements can be characterized as assigning a single label to a large group of related entities within the text. This largely overlaps the preprocessing step identified earlier of collapsing words and phrases that refer to a single concept together (e.g., Oil filter, Oil Filt, and OF325 can all be aliased as Oil-Filter). This step takes that idea further than collapsing misspellings and things that unambiguously reference the same concept, and allows for broader categorical concepts to be created. Extending the above example, if OF325 refers to a specific oil filter on asset 325, it may be more informative to group it together with all oil filters under a single abstract alias than making one for each oil filter location. Conversely, if there are comparatively few specific oil filter assets and the decision maker wants to uncover specific relationships within those oil filter assets, it may be better to give them each their own individual alias. Similar approaches can be used on actions (e.g., all 'digital-adjustment'), locations (e.g., 'east-building-a'), or any other concept you want to interrogate ('morning-event' vs 'evening-event'). The level of specificity of the analysis should be reflected in the alias abstraction of the text.

Annotating extracted text, though not always necessary, can add valuable information to the explanatory model and eventual interpretation of the analysis [33]. This step can be done by adding contextual information about identified elements that are extracted from the MWOs, such as by tagging each with some broad categorical identifier. Categories should be kept comparatively simple and relevant to both the intended goal of the analysis and the specific facility where the data comes from. Any given text-based element could correspond to multiple categorical tags, but ideally any associated tags should come from independent sets of representative information. For example, the text element *BF HEATING UNIT 4* could correspond to the label *item* - indicating it is a physical asset, and the label *Bottom Floor* - indicating its location within the facility. Both of these tags are relevant, but independent, precluding the option for having classification ambiguity during modeling. Whenever possible, tags and metadata associated with identified text elements should have unambiguous categories.

NIST's open source tool *NESTOR* allows users to categorize text elements into simple but informative classes; *problem*, *solution* or *item* [29, 28]. Within this context, a *problem* represents a fault symptom or observed failure mechanism that inspires some investigative or corrective action. These *problems* can be known, suspected, or potential in a way that merits prescriptive preventative actions. *Solutions* are actions taken in response to *problems* and can be responsive or preventative in nature. Finally, *items* are any physical assets that were operated on or were otherwise used during the course of the *solution* actions taken. This simple architecture works well for tagging most text found within MWOs, but could easily be extended or adapted to focus and inform on other pertinent information about the facility, system, or practices. Some intuitive additional tagging schemes might include locations, active process names, dispatched workers, tools needed, etc.

When modeling, these metadata tags can be utilized in several ways. One of the more obvious ways is as direct additional inputs into the explanatory model. This step holds the value of simplicity in implementation, but has potential drawbacks depending on the type of explanatory model used to capture relationships in the data. These drawbacks could include obfuscating interpretation of results, compounding difficulty in optimizing hyper-parameters, and producing potential biases towards the metadata if the model is not constructed correctly. An alternative method for utilizing this information is to embed it directly into the structure of the model. Unfortunately this embedding is only possible when and if the the explanatory model is suited to that type of structuring, such as in the case of Neural Networks that can be layered and combined to process various groups of tagged data separately. An example of this step is shown in the case study (see Figure 3).

These tags can be very important when selecting data for analysis, as they can be used as both filters to focus analysis and/or as descriptors to aid in interpretation of results.

Autonomous Abstraction In a broad sense, the tags identified for each text element are a form of conceptualization, or abstraction of each individual element into a broader concept. Manual (or semi-manual) tagging allows you to have direct control over these concepts, and the text elements that get associated with them. It is important to note here, that for certain types of models, autonomous forms of text abstraction can also be employed with various effectiveness depending on the broader situation. Some examples of this type of abstraction include things like *Word2Vec*, text auto-encoders, latent semantic analysis, and others that aim to push text into some collapsed vector space [19, 39]. Most of these methods require a quantity of quality training exemplars that is unavailable in a typical MWO data set, making them difficult to use out of the box. Many require intimate knowledge of their inner-workings during construction and use to ensure that they do not produce misleading or erroneous results. Despite these drawbacks, and the fact that there is the potential for some loss of physical interpretablity, these automated methods do hold potential to improve many explanatory models' performance and do not require the labor intensive human investment of manual tagging. The use of autonomous abstraction can also be used in conjunction with manual tagging to provide a 'best of both worlds' scenario, but still must be used with caution to accommodate the comparatively low number of examples for a typical text element found in an MWO. This concept is explored further in the case study of this paper.

Scale & Contextualize KPI(s)

Performance indicators should be represented in a manner meaningful to the decision maker interpreting the outcomes of the analysis. To do this may mean some translation or transformation of metrics is needed. In many situations, the intrinsic scale or context of a metric or KPI is not the form best suited to analyze. This could be because the natural scale of the numbers does not reflect the scale of the impact on the system or the value to the questions being investigated, or perhaps the numbers themselves are not ordinal in nature despite being represented as such. For example a change in 'work hours needed' from 7 to 8 may not be the same resource investment as a shift from 8 to 9, because the 9 hours spans multiple work shifts and thus requires significantly more resources. Similarly there are many occasions where continuous numeric values that are ordinal are better represented as discrete sets or categorical data. In the above example, it might be more informative to represent the 'work hours needed' in categorical groups such as *Trivial*, *Quarter Shift*, *Half Shift*, *Full Shift*, and *Multiple Shifts*.

Performance indicators can be numeric, categorical, qualitative, quantitative, continuous, or discrete [13, 35, 3]. The choice of how best to represent the performance indicator is influenced by the desired outcome of the investigation, selected model architectures, volume and proportions of the data itself, as well as other concerns. The choice to convert from continuous values to discrete groupings might may also be driven by the context of the investigation more than the KPI itself. For example, consider the case where there are a high amount of work orders being over budget by $ 20, the minimum changeable amount for extra hours. If the investigations main goal is to determine factors affecting resource allocation, it may be appropriate to categorize all 'over budget' actions into a single category because in terms of the investigation $ 20 or $ 200 over budget still indicates less than ideal planning. Alternatively, it may be that a small overage has little impact and they actually would prefer to place it in the 'near cost' bin. This shows how context of a tracked value can be not only site specific, but goal and investigation specific.

The case study explored for this work shows why and how to discretize the continuous variable of 'duration of maintenance activity', defined in this case as the amount of time a MWO remains open from creation to the completion of the final associated task. In this case the justification for representing the values as discrete bins is both to counteract data imbalance and, more importantly, to match the KPI to time frames that allow more intuitive decisions with more context than the natural scale of time would provide. Additional details about the discritization are presented in the case study section of this paper. The scale and bin demarcations were chosen to reflect the rough temporal scales at which decisions are made. By collapsing the data into scales like 'trivial duration', 'less than a day job', or 'long term job', decision makers can tailor the analysis to gain actionable information specific to their needs. Such scales are the foundation of the analysis. The initial model can provide direct feedback in relevant and actionable terms, and the full sensitivity analysis will reflect the importance implicitly within the selected scale and representation of the KPI. These selections of not only the KPI, but also the scale and formal representation can help fine tune knowledge and insights gained as well as help direct the form and style of the model created for the analysis.

Selection of proper training data is key to obtaining useful results from the analysis. The selection of data can manage or mitigate unwanted biases, aid in focusing on key aspects of the end goals of the analysis, and even help ensure maximum performance by leveraging any intrinsic attributes of the explanatory model architecture. All training data must be taken from MWO natural language that can be associated with a corresponding metric or KPI (e.g., cost or time spent). Any additional information you want to include should also be available for each MWO entry (e.g., technician or operator). General rules and guidance for data selection would follow the same as those generally presented for whichever style of explanatory model selected for the analysis [21]. The key is to ensure both a diversity of tasks represented (often by human inspection), as well as a good representative selection of the relevant KPI.

In many cases the selection of training data is restricted to the available data on hand without much room for optimization. Ideally the data should span the full range of the selected KPI as uniformly as possible. When there are large clusters strongly disrupting the uniformity of the KPI, then scaling or discretization can, as presented in the previous section, help to circumvent any biasing towards those values in the analysis.

When there are ample amounts of relevant data, simple filtering rules can help to dictate or fine tune the focus of the analysis. For example, a decision maker may only be interested in tasks that are above a certain cost threshold, or want to hyper focus the analysis only on tasks performed on a particular asset. By filtering to only the selected cases the results of the analysis can yield more specific insights. However, the authors caution not to do this filtering if it would cause there to be a low number of examples or a deficiency in the distribution of data across the KPI as this can cause unexpected and misleading results if special measures are not taken. While currently no hard rules exist for what constitutes ‘too few’ examples from which to draw training exemplars, the authors suggest that in most cases fewer than 100 well-distributed training exemplars is not recommended for this workflow. This number can vary greatly with the quality of the data, the model, the application, and other situational variables. More work beyond that presented here is needed to give stronger guidance for the minimum number of usable training exemplars required in the data sets [21, 2].

Develop Explanatory Model

The selection of what style of explanatory model is used in the analysis can have significant effects on the outcome. In some cases, the choice of what style of explanatory model to build will be obvious and led directly by the form and amount of data available for the analysis, especially when the amount of data is limited. In other cases, there is more flexibility and thus model choice should be based on the end goals of the analysis, user expertise, and any auxiliary goals or known synergies that could be leveraged to improve the analysis. Three common classes of explanatory models that could be used for this workflow are regression models, conditional probability driven, and ‘black box’ style classifiers. This sections gives a brief overview of using each one, with a short summary in Table 2.

For this workflow, the primary inputs to an explanatory model should be some representation of the significant words present in the MWO. These could be logical vectors indicating the presence or absence of each concept or word alias (one hot encoding[10]), or an encoded vector representative of the concepts within the text made via a computer assisted abstraction technique (e.g., BERT Encoding[7]). However, baring very specific circumstances, the raw text of the document should not be used in the model. Refer to the previous section on preprocessing for more details and justification. Any additional information, such as concept tagging, can also be included: either as an explicit input, or as a structural input that guides the development of the explanatory model. The target output of the explanatory model should always be the selected KPI as either a continuous number or a categorical classification where appropriate. It is important to ensure any selected model can accommodate the format and volume of both the text input and the KPI output.

Continuous Regression Models The most familiar, and easy to interpret class of explanatory models are either based upon simple regressions, or develop some simplistic set of analytic equations relating the input to the output [8]. The authors encourage strong caution when designing and developing this class of explanatory models as they do not lend themselves well to the nature of natural language and are prone to precipitating misleading results. This is especially true with linear regression-based models which treat text derived elements as additive inputs to the KPI. Special design of the model, knowledge of the specific application, or more

Table 2 Explanatory models discussed in here with examples, along with requirements for using them to calculate the value being explained (i.e., a KPI).

Model Type	**Example**	**Requirement**
Regression	Support Vector Regression	Continuous
Conditional Probability	Cox Proportional Hazards Model	Continuous or Discrete Ordinal
Classifiers	Neural Network Classifier	Discrete

complex nonlinear regressors could help to circumvent these problems, but require specific circumstances to work and are generally not recommended for the uninitiated. As an example, the phrases "change left head lamp" and "change right head lamp" are very similar grammatically. However, while an operator would know that there is significant difference in the time required for these two tasks, there is no way to determine that from the grammar of the text alone. If the target KPI relates to the time of each task, it would be very likely that a simple regression model would not be able to capture that large difference in expected time to repair. Most regression models would be unable to identify this type of edge case without a significant amount of data and would generally not give clear indication of such a poor performer. With the nature of natural text and maintenance, it is expected that this type of scenario would represent a significant portion of the cases.

Probabilistic Models One class of models that does not suffer from this type of problem is the type that rely on probabilistic methods to relate the inputs and outputs. Some of these adjust expected distributions of the target KPI, such as the Cox Proportional Hazards Model [6] or a Bayesian Belief network [12]. In this style of explanatory model, the inputs of annotated tags and their corresponding aliased words (or vectorized concept space in cases of computer aided abstractions) are used as conditional modifiers to adjust the base belief value (or distribution) of the selected KPI. Other probability based models, such as decision trees [20], aim to maximize information gain from the inputs to classify and predict the target. One major appealing factor for probabilistic models in this workflow is that many intrinsically provide intuitive mechanisms for identifying important input factors, such as Gini importance or co-variate ranking. However, not every data set is conducive to these models. Before selecting any model for use, the authors suggest reviewing the strengths and weaknesses of that model relative to the specific data planned for use.

Black-Box 'Universal Function Approximator' Models The final major class of explanatory models are those that are difficult for a human to interpret based solely on the internal processes of the model. These 'black box' models are often based on neural networks and machine learning [9]. These types of data driven models are the most broadly applicable, and highly adaptable, but also typically have larger requirements for training to fully develop with any degree of high confidence.

The general inclination when developing an explanatory model is to optimize the performance of that model. While this is true to some degree when utilizing this workflow, it does not need to be the focus. Intensive optimization is only a concern if the predictive output of the explanatory model is an additional goal of your investigation. In general for this workflow, even a suboptimal model can provide valuable insights so long as a minimum level of performance is met. Explanatory model performance below the levels that would be functionally usable in a facility can still hold valuable relationship information and be useful for inferring the relative importance of influencing factors. So long as the performance of the model is *broadly* correct, then the relationships derived within it can be used to extract insight into the relative importance of the various factors used as input for the model. This is especially true and useful when employing a 'black box' style modeling method. As described in the next section, methods exist that rely on relative changes in performance rather than absolute performance of a model. There are many KPIs that are difficult to predict with certainty on an absolute scale utilizing only the information contained in an MWO. To overcome this, the ability to predict relative values or the expected change in values becomes much more important than absolute predictive capability.

Quantify Information Gain

In the context of this workflow, importance is a measure of how much information relating to the selected KPI an observed text element can provide. In the case of an explanatory model, this is its effectiveness as a predictor of the modeled KPI. Importance is used as an approximation for strength of the relationship between the KPI and some asset or action listed in MWOs.

There are two primary ways to get these measures of predictive capability; either from measures intrinsic to the architecture of the model, or by performing variational tests agnostic to the model. This second option requires altering the input sets to exclude various elements and observing the effect on the model's performance. This process is often slow and may require retraining the model multiple times, but can work for nearly every model architecture.

Various ways exist to alter inputs to test their effect on a model, but for the purposes of this workflow, element inversion provides an easy to implement, intuitive mechanism for testing the predictive capability of any given input element. This process involves switching the indicator of that element from 'present' to 'not present' and from 'not present' to 'present' in each entry of the data, re-training the explanatory model, then observing the change in performance. Typically this process will have a more dramatic effect on the performance than simply removing of the indicator from the input, and allows for easier quantification of performance change without the need to normalize to the frequency of the element being tested. In this example, if after inverting the indications for 'Burner', the model shows a 20 % drop in performance, then we can say that the element 'Burner' has a relative importance of 20. Note that the importance is a unit-less number, effective only in comparison to similar values describing the other inputs to this model, and is inversely related to change in model performance. To fully characterize the important factors relating to the selected KPI, this test must be repeated for every significant identified element in the MWO, after the cleaning and aliasing stage of pre-processing, but typically before any autonomous abstraction algorithms are used as these make results less intuitive to invert.

Measures of predictive capability that are more model-specific can also be used to help identify importance of input elements. The specific use and interpretation of these will be unique to the model architecture and the authors encourage research into different models nuances appropriate to the application. In a broad sense, any measure of ability to predict the relevant KPI will work, but some methods are better suited to particular applications than others. As an example, in the case study for this work, decision trees are developed and the corresponding Gini importances[22] (i.e., Mean Decrease in Impurity) are identified and used to express relative importances for the various assets and concepts found in the MWOs. A rough outline of the procedure is provided in the numbered list below. More details about that process are given in the case study section.

Workflow Summary

1. Pre-processing
   (a) Select KPI
   (b) Clean and Standardize Data
2. Main Workflow Procedure:
   Infer Relative Importance of Text Based Elements
   (a) Annotate and Abstract Text
   (b) Scale and Contextualize KPI
   (c) Develop Explanatory Model
   (d) Quantify Information Gain From Input Elements
3. Interpret Results

4 Interpret Results

Any analysis methodology is best tailored with a specific goal in mind. The key indicator of interest must be identified before influential factors and trends can be extracted from MWOs. It is then within the context of that goal that the results must be interpreted.

4.1 Utilizing Results for Decision Making

Once relative importance to the KPI has been established via an information content measure, the next step is to synthesize that information into actionable insights about the system and direct critical decision making. By understanding the primary influences on the target KPI, steps can be taken to ensure that unexpected or undesired influences can be addressed and that positive practices or relationships are reinforced. If the analysis does not provide the levels of insight desired for a particular aspect of the facility, this may indicate a need for more direct monitoring of those areas, tasks, or assets. Each explanatory model and KPI selection will have nuanced interpretations of results tailored to how and why the analysis was constructed, but a basic understanding of identifying assets, practices, or concepts that strongly relate to the KPI is a straight forward process.

As an example, consider the KPI of 'Total Cost of Task' where we are only concerned with using physical equipment and tools extracted from MWOs as the inputs to the explanatory model. Let importance be calculated as the percent loss in model performance with input inversion as described previously. If 'Burner', 'Valve', 'Access_Panel', 'Inline_Filter', and 'Pressure_Vessel' have importance values of 20, 13, -8, 7, and 4 respectively, and we have contextual knowledge that all these text elements relate to the hot water pressure system, then there are several obvious insights that can be gained. We have contextual knowledge about the pressure vessel is a large system with many different tasks that are performed on it. This along with the low calculated importance value of 'Pressure_Vessel' indicates that knowing a maintenance task is performed on the pressure vessel gives little insight to how much it will cost. By the same logic, a listing of 'Access_Panel' provides no knowledge at all about the cost of the procedure. The negative value here indicates that the model actually *improved* (i.e., the change in loss was less than zero) with the inversion of that input. Conversely, knowing that the 'Burner' is involved with the task can greatly improve your confidence in the expected cost of the task. This analysis does not implicitly impart knowledge of the *value* associated with the cost (i.e., if it will be high or low), but instead that tasks involving the burner have a more predictable cost. Knowing that a job involves a burner makes the cost easier to predict or model, thus we can infer that there is something related to burners that strongly influences the cost of associated tasks.

The determination of positive or negative influence on a KPI is both very difficult in a general sense and very easy in specific applications. In some cases a text element is just a conditional modifier, and in others it is related to higher or lower values of a KPI. In most situations, with context and domain knowledge determining if an individual MWO element has a positive, negative, or conditional effect on a KPI is relatively easy and can be verified with either modeled analysis or reference data lookup. Methods such as correlation analysis or probability distribution modeling are typical of the tools one could use to assess this information. While a full prescription of this task is beyond the scope of this workflow, the information derived from this workflow can instigate and focus such analyses.

Relative importance values can help to identify good versus bad practices, highlight anomalies in task identification, spark investigations into sensor placement, and provide other decision support. Adding information and context based tags, or logically grouping the text elements within an MWO, can help not only during model development, but also help during the eventual interpretation of results by connecting groups of actions and resources in the minds of decision makers. For example, if the *problems* with the highest importance are all frequent tasks that negatively impact the cost KPI, a decision maker might investigate adding specific preventative maintenance tasks to help prevent those problems, adding inspections to catch them earlier, or other mitigation strategies. In a similar context, elements labeled as *solutions* could be used to investigate logistics or standardization of procedures. Solutions that frequently occur and have low, or even negative importance could indicate procedures that are highly erratic and are in need of reevaluation or standardization. The most intuitive would be the text elements referring to physical assets or *items*, as these can lead to information on spare parts management, pain point discovery, etc.

Table 3 Real datasets used for the case study.

Dataset	# of MWOs	Industry
A	47797	Automotive
B	13268	Lighting
C	3437	Automotive Supplier

5 Case Study

The following section explores a case study of this workflow as applied to the completion times of MWOs from creation to resolution in actual manufacturing facilities. Data for this case study comes from manufacturing facilities and is representative of data typically found in other industries throughout the maintenance domain. Three datasets are used and described in Table 3.

The data sets are each processed and interpreted via the workflow described in the previous section. For this analysis, the MWO are not sorted or segregated by any intrinsic quality of the MWOs, such as looking at only preventative maintenance work orders versus corrective work orders, or looking only at work orders related to one system in the facility. Although such investigations could yield interesting results, this analysis will be done with the most broad groupings of the data and as few a priori assumptions as possible to help demonstrate the importance of contextualization after the analysis. Multiple tools and algorithms are demonstrated as a comparison of the customization and potential benefits to selecting proper modeling methods in typical circumstances.

5.1 Case Study: Preprocessing

The target KPI for this case study is the duration between the open and close of the MWO, a KPI that reflects the effectiveness of the enacted task at resolving the initiating issue. The longer an MWO is open, compared to similar MWOs, the less efficient the enacted tasks. This is a useful approximation for effectiveness of performing maintenance tasks. Additionally, it showcases the process through representing one of the more challenging cases that may be encountered, because human recorded time values are often inconsistent and unreliable. Finally, in the absence of better KPIs, it can be viewed as an analog to labor or resource cost associated with the task. Such cost values can be useful in logistics planning, resource allocation, and other operational plant decisions.

Although any pertinent value could be used with this processes, investigations such as this one into the factors influencing the duration of an open MWO can provide valuable insights. These can allow managers to verify procedure effectiveness, isolate problem assets or tasks, infer event criticality, and pinpoint justifications for alterations to resource management decisions such as spare part stocks, personnel allocations, and needed redundancy of systems. By first identifying the factors strongly related to completion time, subsets could then be further analyzed to isolate those that have direct positive or negative relationships. Factors that cause increases in MWO completion time could be isolated and targeted for improvement or reevaluation. For example, if having ‘inspect’ in the MWO task description has a strong influence on its duration, and additional analysis shows it has a negative correlation to the value, then it is safe to conclude that performing inspections, in general, has the effect of lowering MWO duration. The importance analysis leads you to discover which terms effect the system, then the correlation analysis tells you how they effect the system. The importance analysis helps rank which correlations should be focused on.

Once the performance indicator for the MWOs is selected, those indicator values, along with the free-form text of the MWOs must be extracted and cleaned in preparation for the analysis.

Clean & Standardize the Data For this work, extracting the duration of an MWO is accomplished by finding the difference between the earliest recorded time on the MWO from the latest in an attempt to capture the entirety of the duration that the MWO is active. This is not a perfect process, as many of the MWOs were inconsistently filled out with times relaying the symptom or fault discovery time, the estimated actual occurrence of the incipient event, the time corresponding to when the MWO was initiated, and other temporal events that could be associated with the incipient event, the repair action, and the MWO itself. Thus by taking the largest difference between time, we are creating the most conservative estimation of the recorded KPI.

Further complicating the matter, there is the chance that some of the times were misreported by the human reporter. Coupled with typos and format inconsistencies, special care had to be taken to ensure as many sensible values as possible were obtained from the records. Even with such precautions, greater than 20 % of the MWOs available were rendered uninterpretable or were missing altogether. A significant number of impossible durations also appear in the data. These appear as tasks ending before they started, zero span duration, or tasks only lasting a few seconds. Such misreportings are unfortunately typical in a factory setting, and can either be grouped as ‘trivially short’ tasks, or at the discretion of the investigator, thrown out entirely. Although the explanatory model could help identify such anomalies, the full discussion of that process is out of scope for this work and may be addressed in future papers.

Table 4 Count comparison of identified important terms before and after human-in-the-loop aliasing

Dataset	Terms Extracted	After Aliasing
A	577	65
B	388	29
C	330	24

With this information, two very simple and obvious rules for rectifying the KPI values can be made. The first rule intuitively holds taking the absolute value of any entries or calculated to ensure no negative values persist.Because negative time values have no physical meaning in this context, it is safe to assume that any negative entries are most likely typos or missordered entries. The second rule is to set any duration below a chosen minimum, in this case 5 minutes, as taking that minimum duration. This avoids some unrealistic answers and helps increase the number of useable cases while adding minimal additional ‘noise’.

To standardize the word choice and correct spelling errors found within the work order data set, the authors employed a previously developed text cleaning strategy. This centered around correcting misspellings, identifying common synonyms and shorthand for various words, then aliasing them all into a single representative word. This was largely accomplished via computer assistance with the NIST developed NESTOR program [28] mentioned in previous sections. Ignoring inconsequential terms such as linking verbs or numeric values (i.e., Stopwords), this tool identified important action words and allowed misspellings, abbreviations, etc. to be categorized under a single representative alias that was additionally classified as either a *problem, item*, or *solution*. For example, ‘hi pressure’, ‘High pressure’, and ‘Hi-Press’ could all be aliased as *High Pressure* and categorized as a problem.

For simplicity in this work, multi-word phases or concepts were not directly identified, instead opting for important entries representable by a single trackable term. Three distinct sets of MWOs were abstracted and annotated this way. On average across the sets, this process dropped the number of trackable terms by around 90%, greatly simplifying the problem space and allowing much more efficient analysis. For a full list of reduction of trackable terms, see Table 4.

5.2 Case Study: Main Workflow

After collecting and managing both the KPI and the actionable text terms found in the MWOs via the preprocessing steps, the main workflow of this methodology to infer important terms and concepts can be enacted. All three data sets are compiled and processed identically.

Annotating and Abstracting the Free Form Text In this case study the annotating of trackable terms was accomplished simultaneously with the language clarification step. Here, the rudimentary trackable term identification and aliasing software also allowed for a human to classify each term as either a *problem, item,* or *solution.*

After condensing the trackable terms, a further level of abstracting the terms into 'concepts' was performed via neural networks and auto-encoding. In this work, this step is done mainly as an exploratory and comparative exercise, with the authors noting that there are other available tools for representing terms with semantic based similarity that may be better suited in specific applications. This abstraction was done in this case study using freely available software found in the SciPy [23] data analysis platform. Using the recommended standard methods, three separate auto-encoders were trained, one for each classification: *problems, items*, and *solutions.* Adopting this structure for the auto encoding uses domain knowledge captured in the classification to help create more meaningful vectorized representations of the important concepts within the data.

We investigate the importance values both with and without the step abstracting concepts with auto-encoders to help highlight the proper use of this technique as well as to provide an example or expected differences. The results are presented in later sections both with and without the auto-encoding as indicated.

Scale & Contextualize KPI The first step in contextualizing the KPI is to visualize it. In this work the range of open MWOs ranged from less than five minutes, to several months for work orders that were delayed due to various reasons, such as ordering replacement parts. Figure 2 shows that there is a clear split in the number of MWOs that take more than and less than approximately 8 hours (one working shift). Labeled on the chart are the five human intuitive time spans that cross this data: *Hour, Day, Week, Month, Year.* Based on the distribution of the data, and for simplicity of interpretation, these time frames were used to create five classes of duration for the explanatory model to predict. The choice to predict classes instead of pure numeric predictions of time was made both because this is generally more informative to an operator (i.e., will this be a quick task, an all day task, a multi day, a multi week, or a multi month task) and also because this simplified the data set for the explanatory model and allowed for more easily justified choices in the selection of the explanatory model. This also accounts for proper scaling of the KPI, because explanatory model selection for mapping ordinal classifications can be done so that the range and scale of the different classes does not depend on the order of the classes.

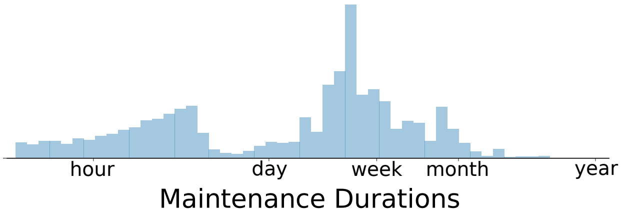

Fig. 2 The log scale distribution of MWO duration across all case studies.

As an additional comparative exercise, we also performed the analysis with with only 4 time designations: *Under an hour, an hour to a day, day to week,* or *more than a week.* This also follows the goal of simplification for both the user in terms of usefulness and ability to interpret, as well as simplifying the required task and complexity of the explanatory model.

Develop Explanatory Model To help compare some of the possible types of explanatory models, both a neural network classifier and a decision tree model were created. The first was created as a complex hierarchical model, created specifically to incorporate the domain knowledge of the analyst and to leverage the investigated structure of the data itself. Conversely, the decision tree was constructed to be an 'out of the box' style explanatory model which could easily be employed with minimum investigations and setup. There are benefits to both styles, and the individual needs and goals of each application of the workflow will determine the better choice.

The neural network classifier has an architecture that operates in stages to capitalize on both the domain knowledge of the situation and the structure of the data found through visual inspection. Shown in Figure 3, the network is comprised of three processing stages. The first stage is considered a preprocessing step, as described earlier, to separate the tracked terms by their respective labels and abstract them into a semantic vector space via auto-encoders. This step is performed so that the tracked terms can be represented by more concise vectors that provide more meaning to the next layer, making it easier to train. The second layer is a binary discriminator that tries to predict if the MWO will be a *'long job'* requiring more than a single workday, or a *'short job'* that can be accomplished in a single shift or less. Looking back at the distribution of times in Figure 2, there is a natural split in the data at this gap, with sufficient examples to characterize both groups. The final layer is a pair of classifiers that further categorize the *'long jobs'* into the either 'week' or 'over a week' and the *'short jobs'* into 'under an hour' or 'more than an hour' tasks.

The second model consists of a decision tree, which is easily interpretable by human experts, and can efficiently classify both ordinal and non-ordinal categories. In this instance, the SciPy Python-based analytics package was used to create the model. Two decision trees were created, one that used the vector representations from the auto-encoders, and another that did not. This step was done to help compare the differences between various model architectures.

Table 5 Recall performance values for developed explanatory models for each dataset. By number of target classes, and use of auto-encoder preprocessing step (AE).

Model	**Dataset**		
	A	B	C
Neural Network			
4-class	0.63	0.57	0.58
5-class	0.62	0.55	0.67
Decision Tree			
4-class	0.68	0.65	0.62
4-class+AE	**0.69**	0.66	0.66
5-class	0.66	**0.70**	0.71
5-class+AE	0.66	**0.70**	**0.73**

Figure 5 shows the resulting performance of each of the tested models in terms of Recall. Recall is defined as the number of correctly predicted MWO duration class divided by the total number of MWOs, meaning that explanatory models with values closer to 1 were able to correctly match more MWOs with their expected category of task duration [4]. From this graph, it is clear that the a decision tree using the abstracted concepts developed through auto encoding to split the MWOs into five categories of task length provided the most correct classifications. However, for this effort of determining relative importance, the absolute performance of the explanatory model is secondary to discovering which inputs are the best at helping the model classify. The thresholds for usefulness can be calculated as one over the number of classes, in this case 0.25 and 0.20 for the four and five class scenarios respectively. These values are the expected number of correct assignments if each MWO were randomly assigned into one of the categories. Thus, we can also determine from Figure 5 that all of the tested models are able to extract some useful information regarding the KPI, making them useful for our workflow. Counterintuitively, a higher recall does not necessarily indicate the model will provide more useful rankings of the tracked terms. As long as the explanatory model is above the usefulness threshold, the usefulness or accuracy of term ranking is difficult to determine at this stage of the workflow.

Also, as may be inferred from the 'lower' recall values of the presented models, the authors are not expressing that any of these explanatory model architectures is necessarily the best for any generic data set, or even this select case study. These are presented as examples to demonstrate the explanatory block in the developed workflow and that a time need not be spent on optimizing a high performance explanatory model to get useful results from this workflow. In general, model selection and optimization should reflect the goals of the analysis, the needs of the data, and the skill limits of the analyst.

Quantify Information Gain From Input Elements The final processing step in the workflow is to relate the impact of the tracked terms to their relative effect on the output KPI values. Sometimes called sensitivity analysis, this is the process of quantifying how much inputs effect the ability of the explanatory model to predict the KPI. This analysis should be structured to output a relative ranking of the predictive influence of each tracked term on the KPI. The rest of the analysis is presented for Dataset A.

For the neural network, a form of inversion analysis was performed. In this analysis, each indication of a tracked term within the data set is replaced with its opposite, then any change in performance of the explanatory model is noted. For example, if the term 'gear' appears in MWOs A, B, and C but not in MWO D, then to assess the importance of the term 'gear' for the model, the data set would be altered to show that 'gear' appeared in D, but not A, B, or C, while keeping all other values and indicators the same. If after retraining, the performance improves, or there is no effect on the performance of the explanatory model, then the term 'gear' is not a significant indicator of the modeled KPI. This method of obtaining the relative importance of terms is largely agnostic to the type of model it is being applied to and thus can be applied widely.

Figure 4 shows the results of the information inversion sensitivity analysis for the four class neural network explanatory model. Based on Figure 5, this was the model with the lowest reported recall. In the figure, the tracked terms are represented by dots with *problems, solutions* and *items*, represented by red, blue, and green respectively. The further left the dot on the chart, the worse the model's recall after inverting the indicative

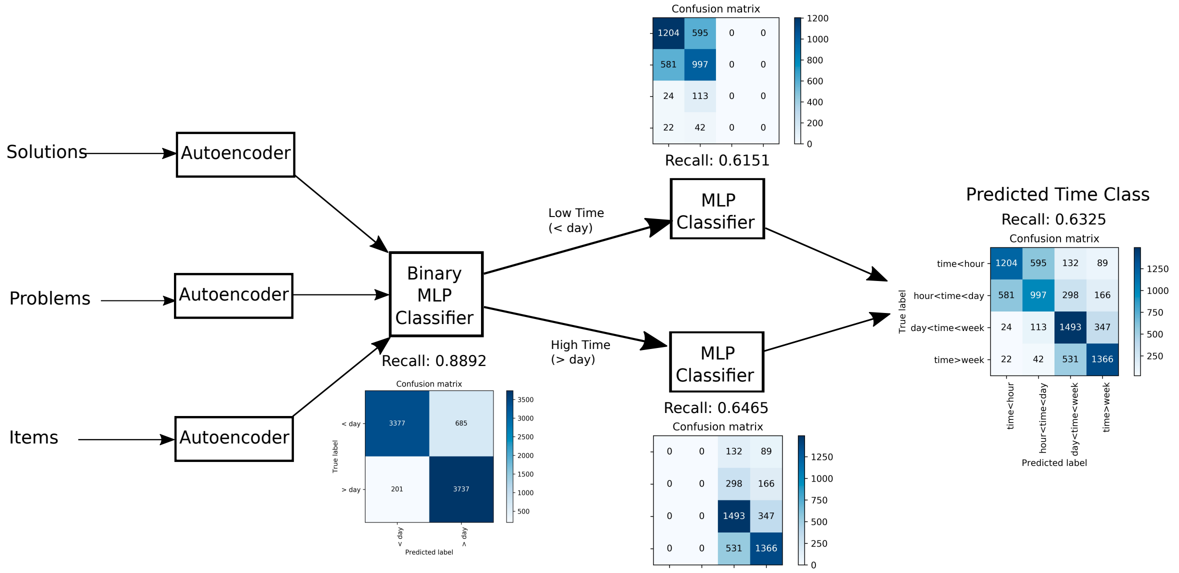

Fig. 3 Neural Network Architecture. This multi stage discriminator attempts to predict the category of time that an MWO would require based on the word use in the document. The structure is designed to capture domain knowledge and have decisions be more human interpretive

information of the term, and thus the more important the term is. The leftmost dots on the chart are those that provide the **most information** about the KPI.

For the decision tree, there is a convenient indicator called the Gini importance that relates the relative importance of the tracked terms to the KPI. This useful value, intrinsic to decision trees, informs how often an MWO would be incorrectly labeled if the term in question did not exist in the dataset.

Figure 5 shows the decision tree's top terms, ranked according to Gini importance for each. These values have been normalized between the sets of *problems, solutions* and *items* so that they can be more easily ranked. By dividing each value of a given category (*problems, solutions* and *items*) by the highest value term in that category it becomes easy to visualize the relative importance of each tracked term. The top ten are displayed on the graph, but the associated value for any tracked term can easily be obtained.

5.3 Case Study: Interpret Results

When interpreting the ranked values of tracked term importance, it is important to remember that these are not direct indications of how or why the term affects the KPI, only that there is some relationship between them. Additionally, while these rankings can indicate there is a mathematically important relationship between the term and the KPI, that does not always mean that there is an interesting relationship in the physical world. For example, all models tested identified 'fault' as the most important tracked term for indicating the duration of the completion of a MWO. While this is mathematically true, it is also uninteresting due to how intuitively obvious this result is given the domain. It is not hard to assume that MWOs resulting from a fault or that have some fault listed in their description will typically take longer. There is little practically useful information gained by this discovery in the analysis, other than perhaps confirmation of our already well held assumption.

Another strength of this analysis is the ability to verify our assumptions about what influences our KPI. For example, if we thought faults had a strong influence on the length of completion of a MWO, but it did not appear in the most important terms of our analysis, that would certainly be cause for deeper investigations as to why. In effect, this part of the analysis is very powerful in telling what is not practically important in effecting KPI. Any term or collection of terms could be similarly queried to verify their impact or, more definitively, the lack of influence on the KPI.

In addition to confirmation probes such as those, the analysis from this workflow can also provide information that would indicate the need for deeper analysis by examining the top terms without specific expectations of what will be there. For example, in Table 6 the most important items with relationships to the MWO duration are 'gantry', and 'operator'. Using the contextual knowledge that *gantry operator* is a meaningful term in these data sets, it is easy to infer that MWOs that relate to a gantry operator are fairly consistent in the

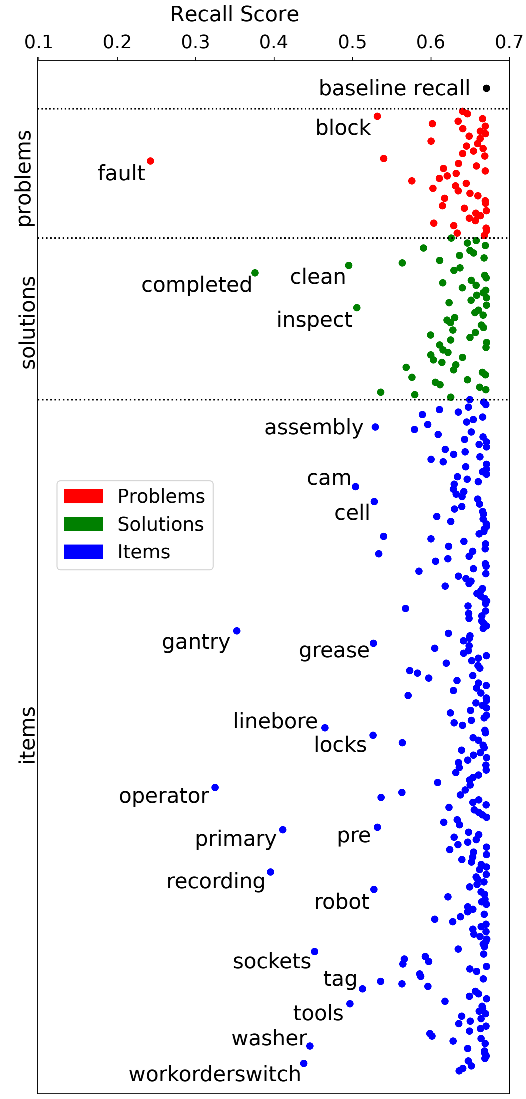

Fig. 4 Change in Recall Performance Values for 4 Class Neural Network Model of Dataset A. Lower scores in the graphic indicates greater impact on prediction performance when that term was removed.

amount of time a given job would take. An additional analysis might find all instances of co-occurrence of these words and merge them into a single expression of 'gantry-operator' in order to gain more insight. Finding significant multi-word phrases can help to better highlight significant relationships by adding clarity to the concepts around them.

'Clean' is the number two ranked solution, meaning that any given "cleaning" job would be expected to take a fairly consistent the amount of time. This does not mean that all MWOs relating to cleaning take the same amount of time. It means that, for example, cleaning the radiator will always take the same amount of time, and that may be different than the amount of time to clean the rear shaft bearings, but cleaning those bearings also generally takes a consistent amount of time. Understanding the output of the analysis can highlight that similar cleaning jobs will have similar resulting times, not that all cleaning jobs take the same amount of time, meaning that the presence of the term can give a much stronger ability to predict the amount of time a task will take. This is an important relationship between the term cleaning and the selected KPI. Once a term has been identified as important, further investigations can help uncover the nature of the relationship to the KPI, like making it higher, lower, or conditionally easier to infer.

Table 6 Top tracked terms (see Fig. 4) having strongest relationships to MWO duration in Dataset A. Matching ranks marked in **bold**.

Problems	Solutions	Items	Rank
Neural Network			
fault	**completed**	operator	1
block	**clean**	gantry	2
	inspect	recording	3
		primary	4
Decision Tree			
fault	**completed**	gantry	1
block	**clean**	operator	2
	replace	washer	3
		primary	4

Finally the authors want to point out that, as shown by Table 6, that both the high performance model and the lower performance model have basically the same rankings for top tracked words. There are some minor differences, but overall most terms will show nearly the same ranking despite being ordered by completely different models. By looking at both Figure 4 and Figure 5, we can see that seven of the ten top ranked items are the same for both models. This will be expected for most models so long as they are capturing some information about the relationships to the KPI and are above the threshold of 'random guessing'. This highlights that the most important step of this workflow is not in the development of an explanatory model, but instead in the interpretation of the sensitivity (i.e., importance ranking) results. If done correctly, the existence of strong relationships should still be discoverable even with less than perfect models.

5.4 Implications for Maintenance Decisions

The workflow and subsequent outputs have major implications for the maintenance decision process. In a perfect world, a maintenance analyst could analyze every item throughout the facility to improve some aspect of the

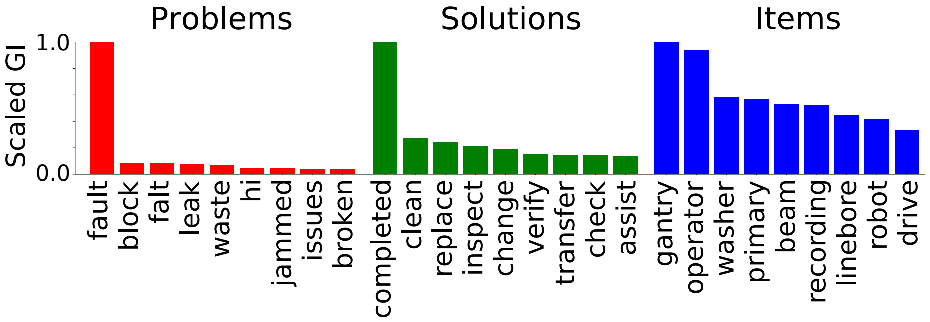

Fig. 5 Top Ranked Gini Importance for Tracked Terms in Decision Tree Model of Dataset A

process. However, this type of analysis is frequently not practical in a real industrial setting.

Analysts are frequently forced to rely solely on tacit knowledge or potentially make decisions on flawed metrics or analysis. Take for example Table 4, if an analyst was forced to use the raw terms extracted from MWOs, they could potentially focus on the wrong items, due to the misspellings, abbreviations, jargon (e.g., studying hydraulic without merging other terms such as hyd, hydaulic, hyrdaulics). Previous works have addressed how to link these concepts, but stopped short of explaining how to further analyze the data with specific KPIs in mind [29].

This workflow provides guidance for analysts on how to take the raw text from the MWOs and help correlate important items, problems, solutions to the metrics of interest within their facility. The goal is to make the outputs explainable and repeatable for any KPI of interest, such as cost or time. This process enables further analysis on the output of the workflow. As an example, the output of the workflow for the case study shows a high relationship between gantry and time. This output allows an analyst to ask 'why?', further investigating why this relationship is happening and improving the usage of gantries in their system. Without this workflow, the analyst may not have known the impact of the gantry system on the maintenance time or focused on other aspects on the manufacturing floor that may not have had as big of an impact.

The workflow described provides managers with peace of mind that this analysis will be done consistently. The process also allows for analysts to swap in different tools at various steps to fit their data and KPI needs. It serves as a best practice guide on how to analyze this MWO data and how to interpret this data to improve maintenance decisions.

6 Conclusions and Future Work

This work presents a procedure (i.e., workflow) for investigating relationships found in the tacit knowledge of Maintenance Work Orders (MWOs) corresponding with performance indicators. There are five major steps to identify informative relationships about a system based on the natural language contained in descriptions within MWOs. First: clean, annotate, and abstract the text into relevant trackable terms or concepts that are meaningful to the end user. Second: scale and contextualize the selected KPI so that it can be easily modeled to develop intuitive association with found relationships. Sometimes even low fidelity measures, such as a simple 'good' or 'bad' indication KPI, can be informative. Third: develop an explanatory model of the relationships between abstracted terms to the KPI, typically by using the first to predict the second. This model development does not need to be highly precise, so long as minimal thresholds of accuracy are met. Fourth: utilize methods of sensitivity or importance analysis to rank and relate the quantifiable effect each tracked term or concept has on the overall performance of the explanatory model. Finally, and most importantly, it is the job of the analyst to reconcile these rankings with the physical system through both interrogative and investigative analysis and domain knowledge.

This workflow can easily be tailored to a wide array of applications. The procedure described in this document is intended as a 'best practice guide' on how to extract actionable insights from natural language text blocks in a way that can accommodate a wide array of specific questions. Searches and analyses could be limited only to documents pertaining to a certain asset or a particular location within a facility to derive high precision relationships or address very specific questions about that subset of the overall system. The analysis could also be filtered by auxiliary information, such as only processing Preventative Maintenance (PMs) actions, or filtered to specific tools having the biggest impact on routine maintenance performance. Investigations such as these could potentially lead to identifying where additional training needs to be given.

This workflow is also extensible by using information external to the MWOs. Where applicable, adding either categorical, quantitative, or qualitative data to the input of the analysis could help direct an even broader range of questions and investigations. For example, adding in the categorical markers for where the maintenance takes place, could verify if location has an effect on performance. Future work will address methods for linking in quantitative sensor values to help identify severity of sensed symptoms to response action performance.

Maintenance and system surveillance is a never ending process. By recognizing that humans are some of the best interrogators available, this work provides a way to translate that tacit and inferred knowledge into a set of quantitative relationships than can be rapidly analyzed by both a human and computer. By using both human driven and human-in-the-loop computer assisted analytics, the resulting decision support will be stronger than either could provide alone.

Conflict of interest

The authors declare that they have no conflict of interest.

NIST Disclaimer

The use of any products described in this paper does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that products are necessarily the best available for the purpose.

References

1. Bokinsky H, McKenzie A, Bayoumi A, McCaslin R, Patterson A, Matthews M, Schmidley J, Eisner L (2013) Application of natural language processing techniques to marine v-22 maintenance data for populating a cbm-oriented database. In: AHS Airworthiness, CBM, and HUMS Specialists' Meeting, Huntsville, AL, pp 463–472
2. Borovicka T, Jirina Jr M, Kordik P, Jirina M (2012) Selecting representative data sets. Advances in data mining knowledge discovery and applications pp 43–70
3. Brundage MP, Morris K, Sexton T, Moccozet S, Hoffman M (2018) Developing maintenance key performance indicators from maintenance work order data. In: ASME 2018 13th International Manufacturing Science and Engineering Conference, American Society of Mechanical Engineers, pp V003t02a027–v003t02a027, URL `https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=924883`
4. Buckland M, Gey F (1994) The relationship between recall and precision. Journal of the American Society for Information Science 45(1):12–19, DOI https://doi.org/10.1002/(SICI)1097-4571(199401)45:1⟨12::AID-ASI2⟩3.0.CO;2-L
5. Carvalho TP, Soares FA, Vita R, Francisco RdP, Basto JP, Alcalá SG (2019) A systematic literature review of machine learning methods applied to predictive maintenance. Computers & Industrial Engineering 137:106024, DOI https://doi.org/10.1016/j.cie.2019.106024, URL `https://www.sciencedirect.com/science/article/pii/S0360835219304838`
6. Cox DR (1972) Regression models and life-tables. Journal of the Royal Statistical Society Series B (Methodological) 34(2):187–220, URL `http://www.jstor.org/stable/2985181`
7. Devlin J, Chang MW, Lee K, Toutanova K (2019) Bert: Pre-training of deep bidirectional transformers for language understanding `1810.04805`
8. Fahrmeir L, Kneib T, Lang S, Marx B (2013) Regression models. In: Regression, vol 1, Springer, pp 21–72
9. Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2018) A survey of methods for explaining black box models. ACM computing surveys (CSUR) 51(5):1–42
10. Harris D, Harris S (2012-08-07) Digital design and computer architecture (2nd ed.). San Francisco, Calif.: Morgan Kaufmann
11. Hodkiewicz M, Ho MTW (2016) Cleaning historical maintenance work order data for reliability analysis. Journal of Quality in Maintenance Engineering 22(2):146–163
12. I BG (2007) Bayesian networks. in Ruggeri F, Kennett RS, Faltin FW (eds) Encyclopedia of Statistics in Quality and Reliability DOI 10.1002/9780470061572.eqr089
13. Iso (2014) Automation systems and integration — key performance indicators (kpis) for manufacturing operations management — part 1: Overview, concepts and terminology. Tech. Rep. Iso22400-1, International Organization for Standardization
14. Jin X, Weiss BA, Siegel D, Lee J (2016) Present status and future growth of advanced maintenance technology and strategy in us manufacturing. International journal of prognostics and health management 7(Spec Iss on Smart Manufacturing PHM), DOI https://doi.org/10.1051/mfreview/2016005
15. Li L, Wang Y, Lin KY (2021) Preventive maintenance scheduling optimization based on opportunistic production-maintenance synchronization. Journal of Intelligent Manufacturing 32(2):545–558
16. Lovins JB (1968) Development of a stemming algorithm. Mech Translat & Comp Linguistics 11(1-2):22–31
17. Lukens S, Naik M, Saetia K, Hu X (2019) Best practices framework for improving maintenance data quality to enable asset performance analytics. In: Annual Conference of the PHM Society, vol 11, DOI https://doi.org/10.36001/phmconf.2019.v11i1.836
18. Meseroll RJ, Kirkos CJ, Shannon RA (2007) Data mining navy flight and maintenance data to affect repair. In: 2007 IEEE Autotestcon, pp 476–481, DOI 10.1109/AUTEST.2007.4374256
19. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. `1301.3781`
20. Mukherjee S, Chakraborty A (2007) Automated fault tree generation: Bridging reliability with text mining. 2007 Annual Reliability and Maintainability Symposium pp 83–88
21. Nalepa J, Kawulok M (2019) Selecting training sets for support vector machines: a review. Artificial Intelligence Review 52(2):857–900, DOI https://doi.org/10.1007/s10462-017-9611-1

22. Nembrini S, König IR, Wright MN (2018) The revival of the gini importance? Bioinformatics 34(21):3711–3718, DOI 10.1093/bioinformatics/bty373
23. Oliphant T, Peterson P, Jones E (2013) Scipy. http://github.com/scipy/scipy/
24. Sakib N, Wuest T (2018) Challenges and opportunities of condition-based predictive maintenance: a review. Procedia CIRP 78:267–272
25. Saltelli A (????) Sensitivity analysis for importance assessment. Risk Analysis 22(3):579–590, DOI https://doi.org/10.1111/0272-4332.00040, URL https://onlinelibrary.wiley.com/doi/abs/10.1111/0272-4332.00040, https://onlinelibrary.wiley.com/doi/pdf/10.1111/0272-4332.00040
26. Savolainen J, Urbani M (2021) Maintenance optimization for a multi-unit system with digital twin simulation. Journal of Intelligent Manufacturing pp 1–21
27. Seale M, Hines A, Nabholz G, Ruvinsky A, Eslinger O, Rigoni N, Vega-Maisonet L (2019) Approaches for using machine learning algorithms with large label sets for rotorcraft maintenance. In: 2019 IEEE Aerospace Conference, pp 1–8, DOI 10.1109/AERO.2019.8742027
28. Sexton T, Brundage M (2019) Nestor: A tool for natural language annotation of short texts DOI https://doi.org/10.6028/jres.124.029
29. Sexton T, Brundage MP, Hoffman M, Morris KC (2017) Hybrid datafication of maintenance logs from ai-assisted human tags. In: Big Data (Big Data), 2017 IEEE International Conference on, Ieee, pp 1769–1777
30. Sexton T, Hodkiewicz M, Brundage MP, Smoker T (2018) Benchmarking for keyword extraction methodologies in maintenance work orders. In: Proceedings of the Annual Conference of the PHM Society, vol 10, DOI https://doi.org/10.36001/phmconf.2018.v10i1.541
31. Sexton T, Hodkiewicz M, Brundage M (2019) Categorization errors for data entry in maintenance workorders. Proceedings of the Annual Conference of the PHM Society, Scottsdale, AZ, URL https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=928437
32. Sharp M (2019) Observations on developing reliability information utilization in a manufacturing environment with case study: robotic arm manipulators. The International Journal of Advanced Manufacturing Technology 102:3243–3264, DOI https://doi.org/10.1007/s00170-018-03263-z
33. Sharp M, Sexton T, Brundage MP (2017) Toward semiautonomous information. In: Lödding H, Riedel R, Thoben KD, von Cieminski G, Kiritsis D (eds) Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing, Springer International Publishing, Cham, pp 425–432
34. Sipos R, Fradkin D, Moerchen F, Wang Z (2014) Log-based predictive maintenance. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining, Acm, pp 1867–1876
35. SMRP (2017) Society of Maintenance and Reliability Professionals (SMRP) Best Practices 5th. Edition. Standard, Society for Maintenance and Reliability Professionals, Atlanta, GA
36. Swanson L (2001) Linking maintenance strategies to performance. International Journal of Production Economics 70(3):237–244, DOI https://doi.org/10.1016/S0925-5273(00)00067-0, URL http://www.sciencedirect.com/science/article/pii/S0925527300000670
37. Szpytko J, Duarte YS (2020) A digital twins concept model for integrated maintenance: a case study for crane operation. Journal of Intelligent Manufacturing pp 1–19
38. Thomas DS, Weiss BA (2020) Economics of manufacturing machinery maintenance: A survey and analysis of us costs and benefits DOI https://doi.org/10.6028/NIST.AMS.100-34
39. Zhang Y, Jin R, Zhou ZH (2010) Understanding bag-of-words model: a statistical framework. International Journal of Machine Learning and Cybernetics 1(1-4):43–52
40. Zhong RY, Xu X, Klotz E, Newman ST (2017) Intelligent manufacturing in the context of industry 4.0: A review. Engineering 3(5):616–630, DOI https://doi.org/10.1016/J.ENG.2017.05.015, URL http://www.sciencedirect.com/science/article/pii/S2095809917307130