Studies to Predict Maintenance Time Duration and Important Factors From Maintenance Workorder Data

Madhusudanan Navinchandran[1], Michael E. Sharp[2], Michael P. Brundage[3], and Thurston B. Sexton[4]

[1,2,3,4] *Systems Integration Division, Engineering Laboratory,*
National Institute of Standards and Technology
Gaithersburg, MD 20899
madhusudanan@nist.gov
michael.sharp@nist.gov
michael.brundage@nist.gov
thurston.sexton@nist.gov

Abstract

Maintenance Work Orders (MWOs) are a useful way of recording semi-structured information regarding maintenance activities in a factory or other industrial setting. Analysis of these MWOs could provide valuable insights regarding the many facets of reliability, maintenance, and planning. Information such as which maintenance activities consume the most work hours, identification of problem machines, and spare parts needs can all be inferred to some degree from well-documented MWOs. However, before one can derive insights, it is first necessary to transform the data in the MWOs (generally some form of natural language) into something more suitable for computer analysis. The National Institute of Standards and Technology (NIST) previously developed a computer aided tagging system that allows for the quick identification of key concepts within the natural language of the MWOs, and a protocol for categorizing these concepts as *problems*, *solutions*, or *items*. Using this annotation method, this paper investigates machine learning methods to gain insights about work hours needed for various maintenance activities. Through these techniques, it is possible to explain the factors captured in the MWOs that have the strongest relationship with the duration of maintenance actions. The workflow of this research is to first build strong data-driven models to classify the duration of any maintenance activity based on the language and concepts gathered from the associated MWO. Sensitivity analysis of the inputs to these classifiers can then be used to determine relationships and factors influencing maintenance activities. This paper investigates two machine learning models - a neural network classifier and a decision tree classifier. Input features for the classifier were the annotated concept tags for solutions, problems and items derived from MWOs of an actual manufacturer. This process for gaining insights can be generalized to various applications in the maintenance and Prognostics and Health Management (PHM) communities.

Madhusudanan Navinchandran et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1. Introduction and Background

Optimizing maintenance activities is important throughout nearly every industrial setting, such as, manufacturing, chemical plants, process engineering (e.g., nuclear plants), and operations of field equipment (e.g., construction and mining equipment). Large scale implementations of maintenance have evolved from primarily reactive maintenance, to regular preventive maintenance scheduling (Sherwin, 2000), then on to more condition based and predictive maintenance practices (Camci, 2015; Helu & Weiss, 2016). Although the dependence and role of the human practitioners has evolved as well, much of the importance of workers has remained the same. The maintenance technician remains an important sensing tool and decision maker in observing symptoms, diagnosing issues, as well as prescribing and enacting maintenance activities. Such specialized, experience-based knowledge is often termed as *tribal* knowledge (Allen, 2013). Some of this knowledge is captured as the natural language put into Maintenance Work Orders (MWOs). Typically, information in an MWO is manually populated by an observing technician when a problem is faced at an asset, conveying details about how the issue was diagnosed and how it was resolved at each stage of the maintenance process. Such MWOs represent a wealth of useful knowledge about the sequence, description, causality, and timing of events with respect to the problem and its resolution. However, given the heavy involvement of manual free-form natural language, this data is often very inconsistent and inadequate for direct computer based analysis. The nature of natural language is such that variations will be

present that require some form of cleaning, translation, and consolidation in preparation for computer based analytics. This paper utilizes a previously developed method to clean and prepare textual data from MWOs (Sexton, Brundage, Hoffman, & Morris, 2017), and then proceeds with its analysis.

By providing methods to analyze this data, knowledge in MWOs can be captured and used to address potential trouble spots (e.g., most problematic machines) and prioritize actions, such as scheduling of operations including maintenance. Particularly in context of this work, models of maintenance action duration derived from MWOs can assist in maintenance action scheduling and determination of anomalous instances that may highlight additional underlying problems.

This paper relates concepts and ideas discovered in natural language field entries of MWOs to the expected duration of the action set associated with that work order. Often the language used to describe an action can give some indication of the relative severity or commitment of resources needed to complete that action. For example, *replacing* a part may tend to consume more time than a simple *adjustment* of a corresponding part. However, following the nature of natural language, there are no absolutes in regard to single words or qualifying concepts. Instead, concepts and ideas are treated as 'soft' influencing factors that may result in different durations given context. Consider that *replacing* lubricant oil consumes less time than *replacing* a motor - hence each solution might have a unique distribution of associated durations. To model the relations between MWO terms and maintenance time, machine learning models are studied and suitably adapted in this paper.

The objectives for this study are to,

- categorize various maintenance actions into meaningful groups based on activity type and duration
- identify the most influential features of maintenance, such as important *problems*, *solutions* and physical *items*
- identify outlier MWO instances to trigger deeper root cause analysis investigations.

The rest of the paper is structured as follows. The state of the art for natural language analysis in manufacturing and current time metrics in maintenance is covered in Section 2. Research challenges and overall methodology for carrying out the study is described in Section 3. A use case showing the implementation of the methodology on a manufacturing MWO dataset is demonstrated in Section 4. Results and issues from the analysis, and scope for extending this research are discussed in depth in Section 5, and conclusions are presented in Section 6.

2. CURRENT STATE OF THE ART

The relationship between maintenance and time related data is often discussed in the domains of reliability and performance monitoring. For example, literature discusses modeling for maintenance intervals with respect to time, by using distributions such as Gaussian, Weibull or Gamma (Locks, 1973). Additionally, there are time-related metrics that focus more directly on the equipment quality and performance, such as the Mean Time Between Failures (MTBF) (Gulati & Smith, 2009). Some research even investigates comparing performance of time-based (calendar-based maintenance) versus condition-based maintenance techniques using only failure time data sets (Ahmad & Kamaruddin, 2012). However, rarely in literature is the time taken (duration) for specific maintenance actions investigated as it relates to those activities, and in particular utilizing information gained from MWOs.

Mukherjee and Chakraborty (2007) discuss work towards understanding the text of maintenance logs to gain insights, such as diagnostic fault trees, from these logs. As mentioned in Section 1, the process of extracting this type of information from MWOs data is not straightforward. The difficulty in processing the language is often due to the free-text entry by maintenance technicians and a lack of constraints on spelling, grammar, or vocabulary. These difficulties are highlighted in (Devaney, Ram, Qiu, & Lee, 2005; Brundage et al., 2019). Maintenance personnel often prefer to enter data quickly, as their job depends on performing maintenance actions efficiently and correctly, rather than entering elaborate descriptions that adhere to formal language descriptions. Though methods have been proposed for identifying manufacturing issues from text (e.g., assembly issues (Madhusudanan, Gurumoorthy, & Chakrabarti, 2017)), this paper focuses on relations between natural language of MWOs and maintenance times.

Once properly processed and prepared, MWO data can be used to compare various possible maintenance actions in regards to expected outcome, duration, cost, or other resource investments. For example, Bokinsky et al. (2013) indicated the need to compare the actual performed maintenance action versus what was listed in a manual to ensure best practices are being upheld. This conclusion followed from a natural language analysis of Maintenance Action Forms for aircraft to reduce the time it is out of service. The timeliness of maintenance is also stressed in (Parida & Kumar, 2006), such as the role of downtime, that is directly related to the maintenance activity being undertaken. In the domain of medical devices, Sipos et al. (2014) built predictive models for equipment failures from billions of event logs.

A need exists to further analyze the rich yet under-explored knowledge of MWOs. In particular, there is a need to capitalize on the potential usefulness of capturing and understanding

time-durations of maintenance activities. This paper, as described in the next section, discusses two potential methods to analyze MWO datasets.

3. METHODOLOGY

This work coalesces free-form information found in MWOs into annotated concepts that are designated into the categories *problems*, *solutions* and *items* using Nestor, a previously developed tool (Sexton et al., 2017). In this context, *problems* are faults, failures, symptoms, or other motives inspiring the MWO. *Items* are the location or equipment where the *problem* is observed or work is performed, and *solutions* are the maintenance action taken. This work studies the *solutions* and *items* that are most important to time spent on maintenance activities. Firstly, it is necessary to clean and identify the various features of MWOs, such as the various *problem* features (e.g. *fault, leak*), *solution* features (e.g., *debur, reset*) and *item* features (e.g., *cylinder, flywheel*). Then, predictive models are built with these features as predictor variables. Based on the models, the most important features are then identified by studying how these features influence the overall maintenance time duration.

3.1. Text cleaning using tagging

A combination of human annotation and computer assistance in the form of Nestor is used to identify the various *problem*, *solution* and *item* features contained in the MWOs. The produced tags, or ideological concepts, are clean representations of noisy unstructured MWO text. For example, *Replace*, which is an alias for all related indications (e.g., Repalce <=> Replace <=> Replacing), is tagged as a *SOLUTION*. Use of tagged words lowers the variations in morphological forms and spellings as compared to raw MWO text, since a human has clarified their correct spelling with the correct alias. For example, in the dataset described in this paper (see Section 4.1), action words (verbs) extracted by Natural Language Processing resulted in 577 *solution* words, whereas tagging resulted in only 65.

3.2. Pipeline for Predictive Model

A pipeline of machine learning algorithms is used to build models for predicting maintenance durations using MWO knowledge. Instead of trying to predict exact times, explainable time classes were used as target variables. The use of time classes has two advantages. Firstly, it provides better performance than when trying to predict exact time values. Secondly, is more intuitive to understand and useful for maintenance managers since the classes give relatable quantities of task duration that provide reasonable expectations of accuracy.

Two separate variations of predictive models were investigated, neural networks and decision trees, and compared for ease of construction, flexibility, accuracy, and intuitive interpretation. Each model pipeline is structured to capitalize on the individual strengths of their respective base models.

3.2.1. Neural Network Model

The first step in the neural network driven pipeline is a set of neural networks called autoencoders. An autoencoder is '*a neural model where output units are directly connected with or identical to input units*' (Li, Luong, & Jurafsky, 2015). Autoencoders enable abstracting input features into more condensed and consistent representations. These were selected due to their ease of setup and ability to efficiently condense related concepts using only the most relevant data. Next a binary classifier is used to obtain a broad prediction of time classes. Finally, each broad class is processed using classifiers to get a finer time class for maintenance.

3.2.2. Decision Tree Model

A decision tree classifier was also investigated and compared to the neural network model. Decision trees were selected both for their intuitive structure in relating input importance, and their speed and strength for classification (this will be described in more detail in section 4.4). With the solutions, problems and items as features, and various time classes as outputs, a decision tree classifier would result in a model to predict any one of the time classes.

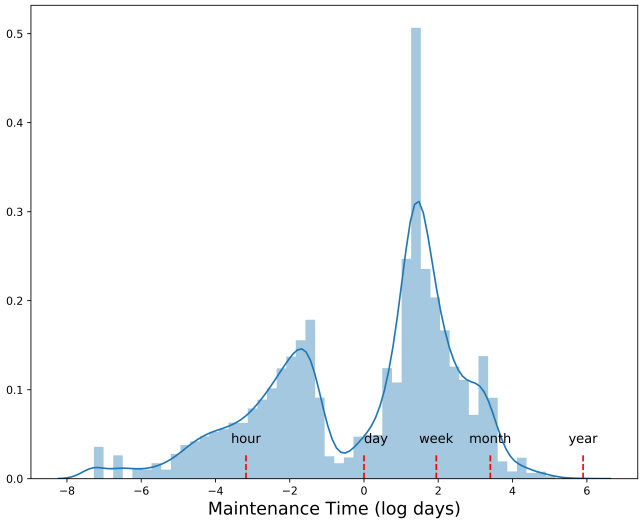
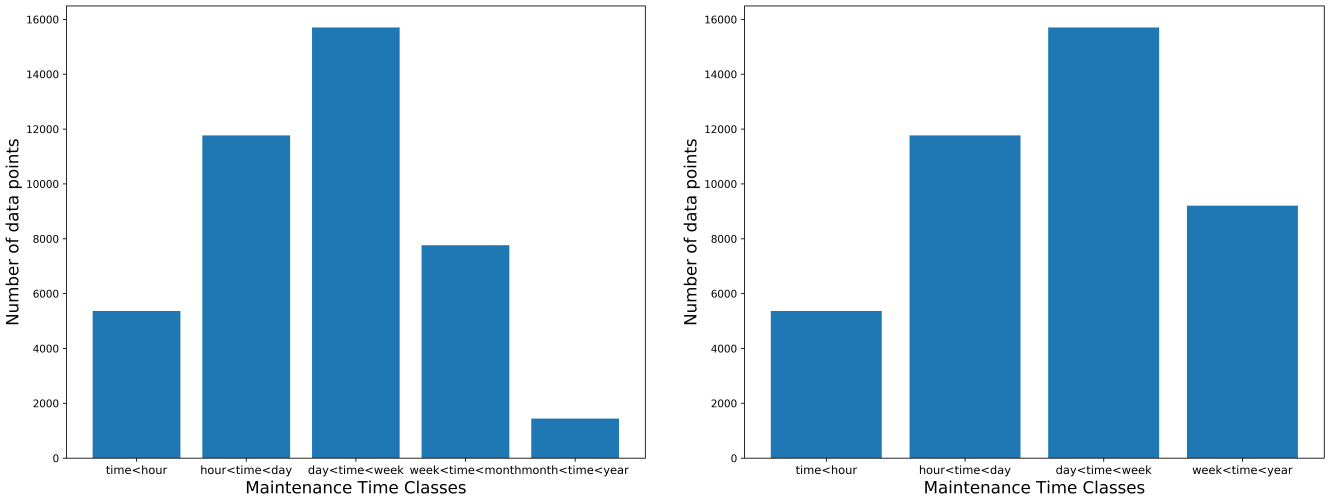


Figure 1. Distribution of times for the dataset. Various timescales are indicated using markers. (The y-axis represents the kernel density estimate for the time values).

3.3. Analysis and Model Interpretation

Once classification models are built for the maintenance features, the structure and behavior of the models themselves be-



(a) Five Time Classes (Average Count = 8409)

(b) Four Time Classes (Average Count = 10511)

Figure 2. Distribution of workorder samples across different time classes. The distribution varies widely with some classes having low number of samples, which are balanced after resampling to the average number of samples.

come the subject of analysis and can be used to determine the most influential input features found in the original MWOs. In order to help determine the most important features, a sensitivity analysis was performed by monitoring the individual models performance when each feature is excluded during training, one at a time. This method is used since it reveals the most influential features on model performance during the construction phase. This is not the only indicator of importance to the relationship between maintenance action duration and the captured concept features but it is a good estimation to help guide further investigations. There are some other methods for sensitivity and importance analysis that are useful - these are discussed in Section 5. For the decision tree, additional information regarding the most influential features is found by looking at the importance of features in the decision tree structure itself.

With this generic methodology in place, we now describe a case study that illustrates the application of the methodology to analyze a real manufacturing dataset.

4. PREDICTIVE MODELS - A CASE STUDY

This section illustrates the application of predictive models described in the previous section on a manufacturing dataset and presents the performances of the models.

4.1. Dataset used

The MWO dataset used for this study was sourced from a real automotive manufacturer and consisted of 47 798 MWOs. The dataset is in a spreadsheet format, and some of the fields are *Workorder Number*, *Status*, *Actual Start Date and Time*, *Actual Finish Date and Time*, *Asset Number*, Textual *Description*, *Location* and *Reported By*. More details about information contained in MWOs can be found in (Brundage, Morris, Sexton, Moccozet, & Hoffman, 2018). The most important fields from these MWOs are the actual start and finish times and the two text description fields about the maintenance activity.

4.1.1. Data Quality Challenges

In preparation for the analysis portion of this work, the non-uniformity of the dataset presented some unique challenges that are likely to be common in real world applications. Every MWO dataset has its own characteristic fields, such as *Asset Number, Workorder Number, Problem Description, Requested By, Solved By, Opening Time* and *Cost Incurred*. For this research, the text descriptions of the problems and actions taken and time-related fields are of interest. The text descriptions were annotated with tags using the methods described in (Sexton et al., 2017).

MWOs contain dates and times in different formats and hence must be preprocessed to get maintenance time durations. To improve consistency, all time data is converted to days with a range across the data set spanning from zero to hundreds of days. The dataset used for this paper had only the starting and ending times for the workorders, not the actual task work hours. Hence, it was not possible to ascertain the actual duration of maintenance solutions. The assumed duration is deemed to be the end time minus the start time listed on the MWO. These time calculations do not always accurately reflect the actual time taken for maintenance, but are instead rough estimates due to inconsistencies and variations

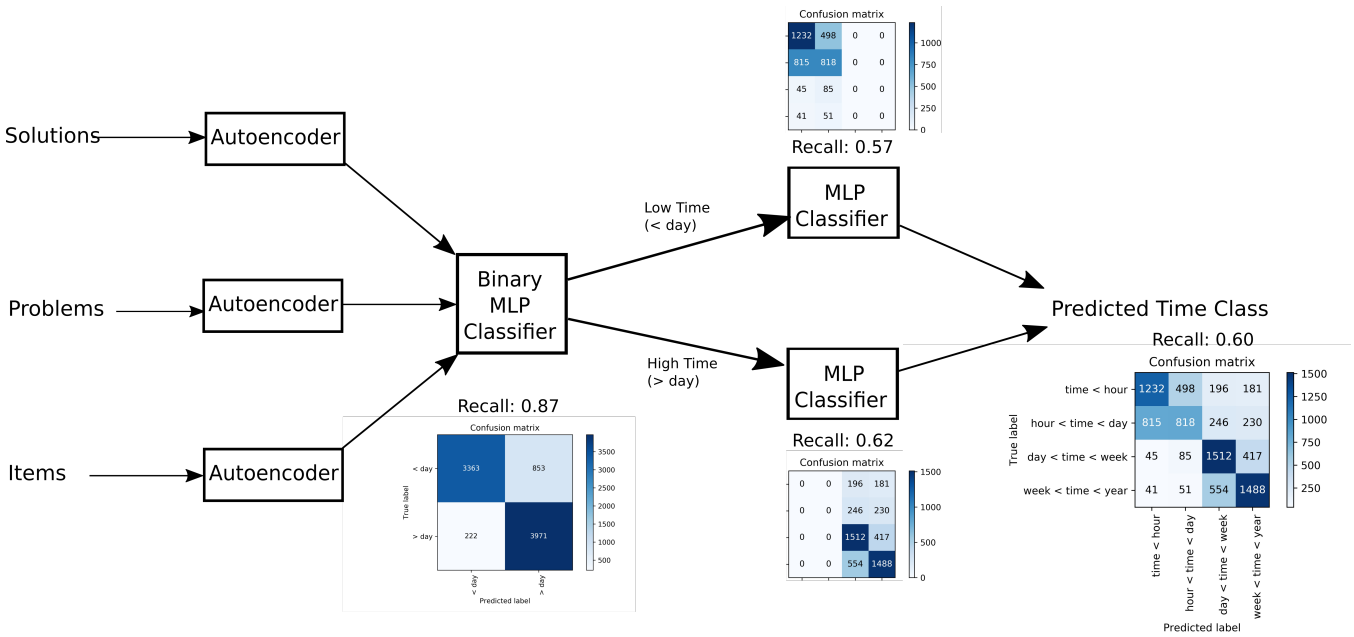


Figure 3. The schematic of the architecture of the classifier model used. Recall scores at each classification step, along with the confusion matrices are also shown.

in recording times. Often, missing time entries exist for some maintenance activities. For this case study, there are 4 914 entries with incorrect formats and missing time entries. There are a further 843 entries with negative total duration. These are excluded for the purpose of analysis. More fine-tuned, and correct time recordings are needed to improve the accuracy of these resulting models. More details about this aspect are presented in Section 5.

4.2. Time distributions and time classes

The intended output for the predictive models is a categorical window of the maintenance duration. Though maintenance times in MWOs are real-valued, predicting the output times as precise real-valued numbers is far less useful than practical task assignment windows because jobs are typically scheduled into some window of an expected duration of the task. In other words, short tasks may be scheduled in 5 min blocks, but longer tasks are more commonly blocked off in terms of hours. For this data set, some concessions of the designation of the duration categories is also fed from the small number of data points and low accuracy of predictions found in initial studies with the time data. With larger volumes of more accurate data this could be overcome. Despite these concessions, the conclusions and methods developed in this work could easily be extended to the level of granularity most useful and feasible for any target use case.

The designation of the classes in this work was largely led by expert intuition and observations of the data itself. The distribution of times shown in Figure 1 makes it clear that there is a large split in amount of actions requiring less than a day, and another smaller cluster break between the week and the month markers. Additionally, practical and relatable demarcations such as *hours*, *days*, *weeks*, *months* and *years* are more explainable and help foster understandings of the ensuing analysis than simply putting across numerical ranges of times.

Figure 2(a) shows the distribution of frequencies of samples across five classes (*hour, day, week, month,* and *year*). The data points were resampled across all classes to match the average number of samples across all classes. Since there are a relatively small number of values in the fifth class (month<time<year), it can also be combined into the fourth class and represented as a single time class (week<time<year). Figure 2(b) shows the same distributions if there are only four classes, corresponding to *hour, day, week* and *year*. The four classes consist of:

- Less than an hour ($<$ 1 h)
- Greater than an hour but less than a day (1 - 24 h)
- Greater than a day but less than a week (24 - 168 h)
- Greater than a week ($>$ 168 h).

4.3. Application of pipelines to manufacturing dataset

With the *solution, problem* and *item* tags as features and time classes as outputs, classifiers are built to predict the time classes for maintenance. A schematic for the entire pipeline is shown in Figure 3. The first step is to preprocess the input features using a set of autoencoders. These autoencoders

are used to compress these features into approximately half the original number of inputs. The autoencoders are used to both focus the classifiers and help to remove tangential information contained in the original feature set. The number of *problems* reduces from 51 to 32, *solutions* from 65 to 32, and *items* from 271 to 128.

The first stage binary classifier was set to delineate at the largest gap in the original data distribution, the less than one day mark. The purpose of this classifier was to roughly judge if the jobs were 'short' or 'long' based on the language found in the MWO. The classifier implemented is a Multi-Layer Perceptron (MLP) with three hidden layers. MLP is a neural network with an input layer, an output layer and at least one hidden layer. The performance of a classifier is judged using the *recall* score, which is the fraction of all correct time classes correctly identified by the classifier. It calculated as the ratio of the number of *true positives* to the sum of *true positives* and *false negatives* (with an averaging method for multiclass classification). The split between training and testing sets was 80 % to 20 % respectively. The binary classifier performs with a recall score of 0.87 (For five classes, recall=0.88).

In the next step, each of the long time and short time classes are separately classified using two separate additional MLP classifiers. These additional classifiers respectively further classify each MWO input into the *hour* vs. *day* categories if the MWO was found to be a 'short' job or the *week* vs. *year* categories if it was designated a 'long' job.

The 'long job' (high time) MLP classifier (recall = 0.62) was better during testing than the 'short job' (low time) MLP classifier (recall = 0.57). When the predictions for all the classes are combined, the overall recall is 0.60 (For five classes, overall recall = 0.59).

4.4. Decision Tree Classification

The design and implementation of the previous pipeline of classifiers inspired a separate step-by-step classification approach, a decision tree classifier. Similar to the previous pipeline, the target output for the decision tree is the time classification of each MWO based on its captured language. Decision trees were built both with and without the use of auto encoders as preprocessing nodes to compare if there was any trade off between end analysis results vs model accuracy. The performance of decision tree classifiers (using only input features) is shown in Figure 4. The time classes corresponding to within an hour, week and year are more correctly classified than the in-between classes (day and month). In general, the decision tree classifier had better performance than MLP classifiers, with recall scores of

- 0.66 (only input features and 4 time classes)
- 0.67 (using autoencoders and 4 time classes)
- 0.64 (only input features and 5 time classes)
- 0.65 (using autoencoders and 5 time classes)

The use of autoencoders to preprocess the features improves the performance of the classifier slightly, but also adds a layer of obfuscation to the final sensitivity analysis that may outweigh the accuracy gain in many cases.

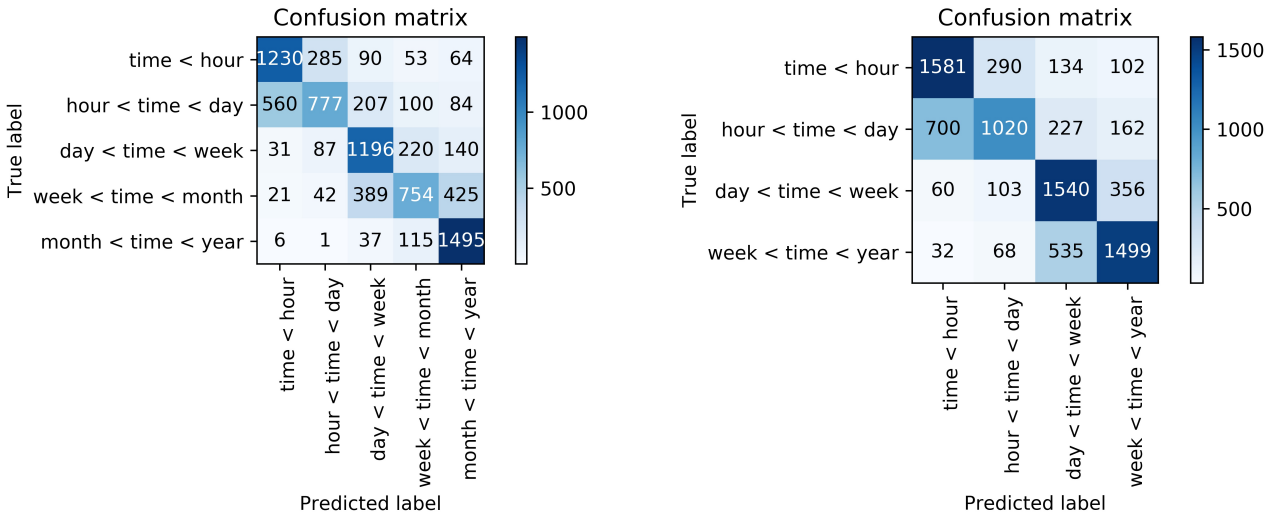
4.5. Most important features

The models described earlier have targeted the prediction of estimated time for a maintenance activity. However, it is also useful to inform the maintenance manager about the most important features that influence the amount of time taken. There are two methods described here to decide the most important features. These methods are demonstrated on the decision tree classifier method described in the previous subsection.

In the first method, the decision tree classifier (with autoencoders for preprocessing) is supplied with all features, except one. This is repeated for all features, and the performance of the classifier is recorded. Figure 5 shows how the recall varies for each feature removed, for 387 features. The important features can be inferred from low recall values when the feature is absent.

As an alternate method, the important features for the decision tree classifier (without autoencoders) are obtained from their Gini importance (Shouman, Turner, & Stocker, 2011). This method did not use autoencoders since the output features from autoencoder are not uniquely identifiable. The various *problems*, *solutions* and *items* are shown in Figure 6(a). Since there are more items due to it being the largest category, the features are shown again in Figure 6(b) by dividing with the number of features in that category.

Between the two different methods there are some common features that are shown as important. Common solutions were *completed* and *clean*; some common problems are *fault* and *dirty* and some items are *beam, conveyor, hmi, primary, linebore* and *clamp*. This list is useful to a maintenance manager to help identify which solutions have the greatest influence on maintenance time. It is also useful to identify items that are anomalous with respect to time i.e. the maintenance for these items takes too long or are unusually quicker than expected. Deeper analysis for determining the root cause may then be undertaken. For example, the words *dirty* and *clean* have large importance - this could mean that depending on whether something was dirty and cleaned would strongly determine the amount of time for maintenance. Common sense tells us that cleaning is not a very time consuming activity as compared to, say, replacing an entire part. Similarly, items such as *conveyor* and *linebore* are consistently found to influence time durations. Such sensitivity analysis can be coupled with other models such as regression models to determine the



(a) Five Time Classes

(b) Four Time Classes

Figure 4. Performance of Decision Tree Classifiers for five and four time classes.

nature of influence of important features - whether they contribute to increased (or reduced) time durations. This analysis could eventually lead to better planning and resource management by identifying and quantifying reasonable expectations regarding various maintenance tasks within a facility.

5. DISCUSSION AND FUTURE WORK

This paper has discussed the use of machine learning models to predict time classes for maintenance duration. Numerous factors influenced the performance and results of the feature importance analysis. Some of the notable observations regarding that are listed here.

Assignment of Duration Categories: During the course of the investigation, various demarcations of time duration categories were explored to best describe the data. For the MLP classifier, the binary classifier performance decreased by reducing number of classes from 5 to 4. For the overall classifier, as well as decision tree classifier, reducing the number of classes led to improved performance. Also, within the multi-step classification of the MLP classifier, the performance is noticeably better for the binary classifier than for the second classification step for four/five classes. These specific results are somewhat dataset dependant, but the general trend of searching for classes that are adjacent to each other are largely expected to to improve performance regardless of the dataset.

Use of Tagged Data: The analysis illustrated the value of using tagged data, since the number of features to be used reduces significantly. Apart from clarifying the terms used, tagged data makes the analysis computationally feasible in terms of having to use lesser features. It also makes the results more explainable and coherent.

Quality of Time Data: Unusable data entries are a major issue. These are either missing time entries or incorrect entries, such as closing times for workorders that are earlier than opening times. MWOs for which there were missing dates and times were entirely ignored but this reduces the amount of correct data available to train the models. Such issues emphasize the importance of collecting accurate time related data during maintenance.

Maintenance Data Collection: The time data available in the dataset were only the actual start and finish times. There is no more specific time information e.g., when the maintenance technician arrived, or when the workorder was opened. Such finer time data would lead to improved inferences about the actual duration of maintenance. Further details of what other time data are useful can be found in (Brundage et al., 2018). Efficient data collection strategies are needed for better maintenance time data capture.

5.1. Application to maintenance management

The general procedure for analysis is an outcome of this work. To derive maintenance management insights starting from workorder data, the following procedure is a recommended workflow:

Clean the data by using Nestor to tag the data. This preprocessing results in a representation of data that is possible to be analyzed.

Calculate number of time classes for maintenance time du-

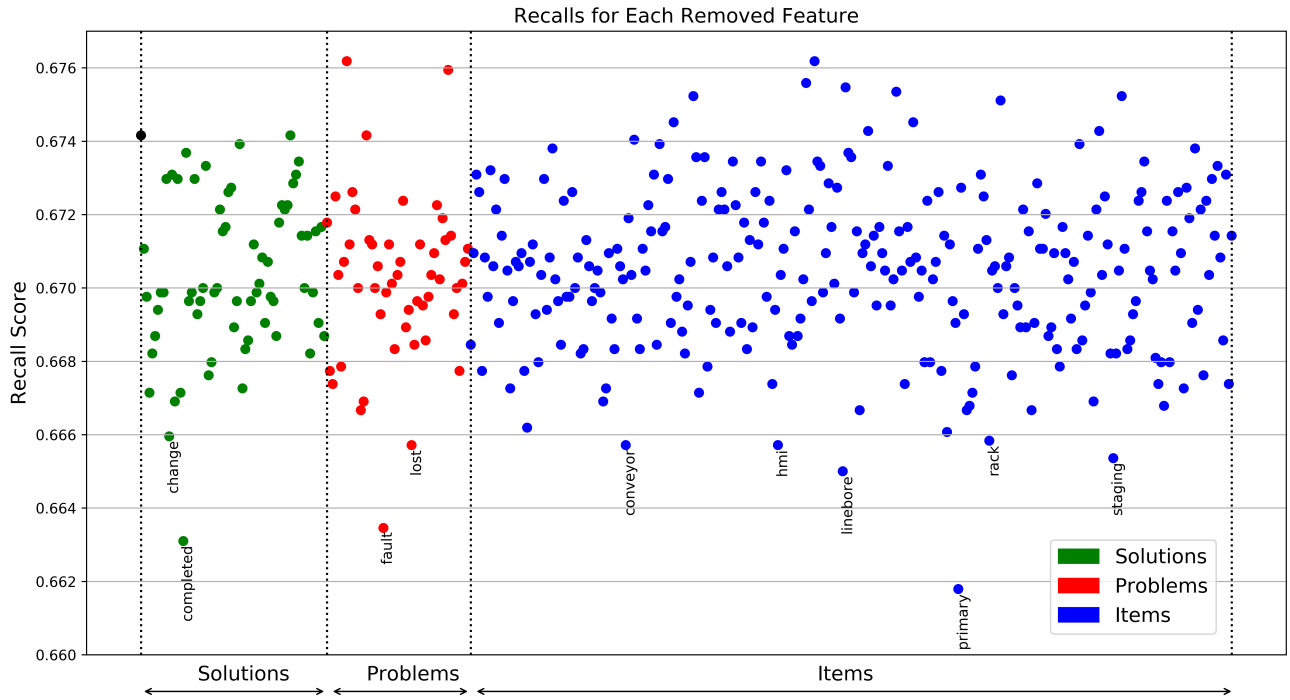


Figure 5. Performance of the decision tree classifier (with autoencoders for preprocessing) by removing one feature at a time. Some of the low recalls are labelled with the feature that is removed at that point.

rations. An appropriate number is chosen by looking at the distribution of times such that the number of entries in each class is comparable.

Choose a machine learning classifier depending on computing resources, dataset size and expected performance levels. A couple of examples have been discussed here, but there are many more options available. The use of machine learning models also involves splitting data for training and testing (For this paper the split was 80 % to 20 % respectively).

List out important features using methods such as feature importances for decision tree classifier. This would help maintenance managers identify hotspots such as certain items that have been contributing heavily to maintenance duration, even when it is not apparent without the analysis.

Identification of important features such as *problem* features, *solution* features or *item* features will help to address specific points in maintenance.

5.2. Future work

The machine learning models can be improved and more generalized observations can be derived. This is possible with larger and varied datasets and more tagging on datasets (such as greater time spent on tagging and tagging of bigram phrases). It could lead to generalized observations of tags that are indicative of maintenance domain. For example, some terms might relate to expensive or time consuming solutions, such as needed a *replacement* part. This could potentially contribute to language standards for MWO recording practices in maintenance.

With regards to the sensitivity of the models to input features, it is also planned to use the method of leaving out one feature at a time on the Neural Network Model. Also, there are other methods that could be used, for example, one could change the values of only one feature at a time to know the effect on predictions. Another method is to use partial dependence plots for visualizing importance of a given feature, one or two at a time. Use of such multiple methods might help to generalize the list of important features. These will all be addressed in future work.

This work utilized a dataset from the domain of manufacturing maintenance. Similar analyses can be performed on MWOs from other domains, such as aerospace, shipping, heating ventilation and cooling (HVAC), to identify common and domain specific parts of language that relate to the maintenance duration.

Wherever data is available, similar models can be built and studied for cost of maintenance activity.

5.3. Other applications

Beyond identifying important features, predictions of maintenance time could be useful for maintenance and production

scheduling. Time windows of maintenance obtained from the model can be used as input to decide how long a resource may be unavailable.

Based on a problem condition at a machine, a maintenance manager could search through MWOs for previous solutions. However, there could be many solutions, and no way to prioritize which solution to perform. The solutions to a problem may be about merely lubricating a part, or entirely replacing a component. Modeling the relation between the solutions, problems and items provides a means of ordering these solutions by the amount of time to be likely consumed. Prioritizing might lead to potential savings in overall time.

There are potential uses from this work with respect to managing the inventory of spares. If there are items that need frequent replacement which influence maintenance times, these items could be stocked in spares to shorten the duration.

The distributions of solutions, problems, items and times could be used to build simulation models of higher fidelity that have multiple failure modes and treatments. Thus, simulations of maintenance scheduling might be more well informed.

6. CONCLUSIONS

This paper discussed the analysis of MWO data to help estimate maintenance duration and identify important *problems*, *solutions* and *items* features. These features are used to build machine learning models to predict estimated time duration of maintenance activities. From the machine learning models, it is also possible to infer important features that influence maintenance time. The methodology for using MWO data to infer important features involves cleaning the text data, deciding on the time classes, fitting predictive models and listing most important features from the models.

ACKNOWLEDGMENT

The authors wish to acknowledge the inputs and support of Sascha Moccozet in the initial phases of this work.

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

Ahmad, R., & Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & Industrial Engineering*, *63*(1), 135–149.

Allen, R. (2013). Tribal knowledge. *Quality*, *52*(1), 54–59.

Bokinsky, H., McKenzie, A., Bayoumi, A., McCaslin, R., Patterson, A., Matthews, M., ... 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.*

Brundage, M. P., 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* (pp. V003T02A027–V003T02A027).

Brundage, M. P., Sexton, T., Hodkiewicz, M., Morris, K., Arinez, J., Ameri, F., ... Xiao, G. (2019). Where do we start? guidance for technology implementation in maintenance management for manufacturing. *Journal of Manufacturing Science and Engineering*, 1–24.

Camci, F. (2015). Maintenance scheduling of geographically distributed assets with prognostics information. *European Journal of Operational Research*, *245*(2), 506–516.

Devaney, M., Ram, A., Qiu, H., & Lee, J. (2005). Preventing failures by mining maintenance logs with case-based reasoning. In *Proceedings of the 59th meeting of the society for machinery failure prevention technology (mfpt-59)*.

Gulati, R., & Smith, R. (2009). *Maintenance and reliability best practices*. Industrial Press Inc.

Helu, M., & Weiss, B. (2016). The current state of sensing, health management, and control for small-to-medium-sized manufacturers. In *Asme 2016 11th international manufacturing science and engineering conference* (pp. V002T04A007–V002T04A007).

Li, J., Luong, T., & Jurafsky, D. (2015). A hierarchical neural autoencoder for paragraphs and documents. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)* (Vol. 1, pp. 1106–1115).

Locks, M. O. (1973). *Reliability, maintainability and availability assessment*. Hayden.

Madhusudanan, N., Gurumoorthy, B., & Chakrabarti, A. (2017). Automatic expert knowledge acquisition from text for closing the knowledge loop in plm. *International Journal of Product Lifecycle Management (IJPLM)*, *10*(4).

Mukherjee, S., & Chakraborty, A. (2007). Automated fault tree generation: bridging reliability with text mining. In *Reliability and maintainability symposium, 2007. rams'07. annual* (pp. 83–88).

Parida, A., & Kumar, U. (2006). Maintenance performance measurement (mpm): issues and challenges. *Journal of Quality in Maintenance Engineering*, *12*(3), 239–251.

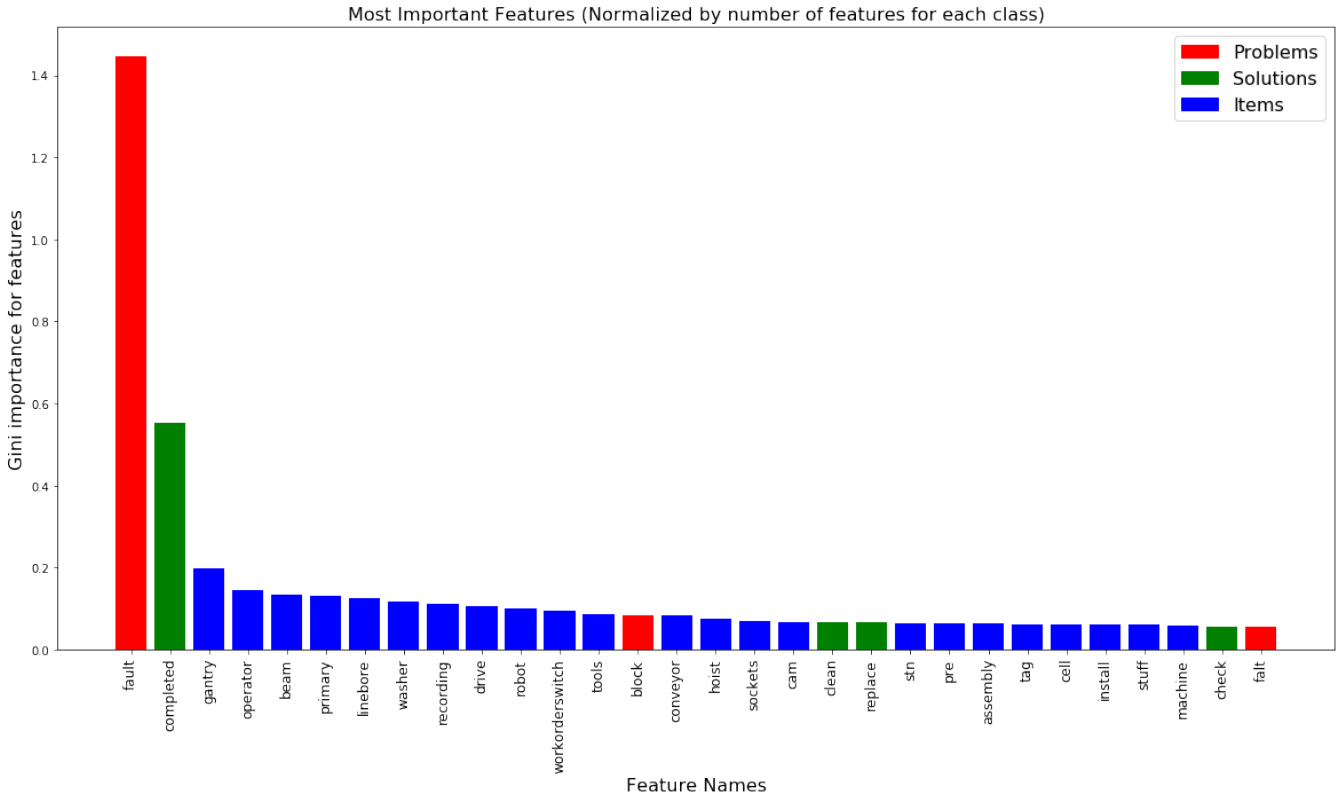
Sexton, T., Brundage, M. P., Hoffman, M., & Morris, K. C.

(2017). Hybrid datafication of maintenance logs from ai-assisted human tags. In *Big data (big data), 2017 ieee international conference on* (pp. 1769–1777).

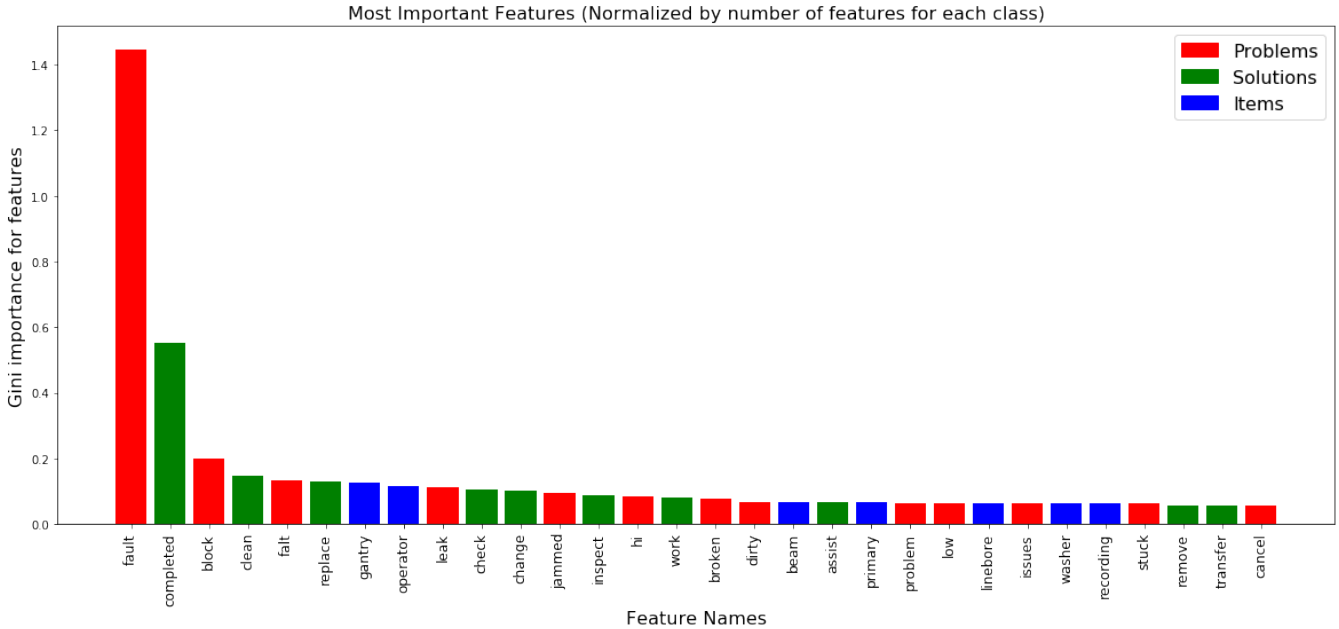
Sherwin, D. (2000). A review of overall models for maintenance management. *Journal of quality in maintenance engineering*, *6*(3), 138–164.

Shouman, M., Turner, T., & Stocker, R. (2011). Using decision tree for diagnosing heart disease patients. In *Proceedings of the ninth australasian data mining conference-volume 121* (pp. 23–30).

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* (pp. 1867–1876).



(a) (without normalizing).



(b) (after normalizing using number of features).

Figure 6. Ordered list of most important features from the decision tree classifier. These features are ranked by their order of Gini importance. In Figure (a), the top 30 features are simply ranked by importance. Since the number of features is maximum for *items*, most of the important features are *items*. Hence, Figure (b) shows an ordered list, where the importance is divided by the number of either *problems*, *solutions* or *items*.