

A Fine-grained Semi-supervised Anomaly Detection Framework for Predictive Maintenance of Industrial Assets

Xiaorui Tong¹, Wee Quan Jung¹ and Jeremy Frimpong Banning¹

¹*Raytheon Technologies, Enterprise Data Services, Brooklyn, NY, 11201, USA*

xiaorui.tong@rtx.com
weequan.jung@rtx.com
jeremy.banning@rtx.com

ABSTRACT

Reliable operation of industrial assets is of high priority for businesses where productivity determines the ability to deliver safety-critical products of high quality in a timely manner. The aerospace industry leads the demand for predictive maintenance (PdM). In the manufacturing space, unscheduled down time causes production delay and increases operational costs while introducing potential risks in product quality and on-time delivery. In field application of these products, unexpected breakdown of critical components can result in safety-critical events. Failure events are, therefore, extremely rare in industrial settings. Diverse operating conditions in the manufacturing environment and field applications contribute to the heterogeneous nature of data collected from these assets. This work presents an anomaly detection framework for PdM of industrial assets to address the practical challenges of scarce failure data sources and heterogeneous data across assets. We introduce a fine-grained modeling approach that efficiently accounts for individual asset differences in a semi-supervised fashion which requires only normal operation data for model training. The framework is demonstrated with an end-to-end industrial use case. Vibration sensor data from pumps in one of our manufacturing facilities is ingested to enable PdM with 2 weeks lead time using the proposed framework. This transforms unexpected breakdown time to scheduled maintenance, thereby reducing cost of delays and operation interruptions. The systematic implementation of the framework in the case study covers the practical production aspects including data quality evaluation, model training, optimization and daily serving of predictions. Furthermore, implementation challenges and recommendations are discussed based on the end-to-end solution implementation

experiences.

1. INTRODUCTION

With the advancement of data collection, transmission, storage, streaming, sensor monitoring technologies, and increased awareness of valuable data insights, industries are striving to leverage data analytics and computational models for business value proposition at ever-increasing speeds. Among the plethora of data types, time-series data is the most common type, where data records are timestamped and meaningfully interpreted within the sequence. While abundant information can be accessed via time-series data analysis, anomaly detection is of high interest due to common business needs of staying proactive and informed of unexpected events; whether it is to avoid hazardous events and productivity loss manifested by anomalies, or to detect changes in sales patterns informing product popularity. For example, a milling machine with embedded voltage and pressure sensors as well as externally installed rotational speed and vibration sensors can communicate time-series data records to downstream models for condition based monitoring (Cavalieri & Salafia, 2020). The Numenta Anomaly Benchmark (NAB) dataset (Ahmad, Lavin, Purdy, and Agha, 2017) provides a variety of time-series streaming data sources ranging from server network utilization to medical sensors accessing patient health. Anomaly detection algorithms are tested with these data sources to achieve meaningful insights from uncovering a bottleneck in an IT network to alerting patients under stress in a real-time streaming manner.

Anomalies can be categorized in general into 3 categories for time-series data. (1) Point anomalies refers to individual data points that are out of range compared to the rest of the data (Teng, Lin & Wen, 2017). These anomalies are usually easier to be detected with threshold-based approaches. (2) Contextual anomalies are anomalies that deviate from data points under similar contexts or conditions. For example,

Xiaorui Tong 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.

vibration sensor readings collected when rotational speed is at 5000 rpm with misalignment present in the system can manifest as abnormal when compared with readings collected under the same rpm with no misalignment. These same anomalies may appear to be normal when comparing with readings collected at higher rotational speeds. (3) Collective anomalies, when assessed together as a collection of data points, deviate from the rest of the data. Trending anomalies are a common type of collective anomalies, where the data points can be within a normal range, but the sequence displays abnormal trend when evaluated collectively (Shaukat, Alam, Luo, Shabbir, Hameed, Li, Abbas & Javed, 2021).

While there is no proven anomaly detection approach that works the best for different datasets against all categories of anomalies, there are ways to adapt the anomaly detection workflows to expand the types of anomalies that can be detected. This adaptation can happen at the feature engineering stage. For example, when adding the context associated with target anomalies into the feature dimension either through normalizing existing features according to context or using context features as additional dimensions (Foorthuis, 2021), methods suitable for detecting point anomalies and collective anomalies can work for contextual anomalies. When features are aggregated considering sequence, collective anomalies can be transformed to point anomalies. These adaptations do introduce challenges in the feature engineering space in terms of knowing what context to include, the proper window for sequence construction, and identifying the underlying correlations for normalizing features based on context etc.

Existing research efforts in the realm of anomaly detection cover supervised, semi-supervised (Jiang, Kao & Li, 2021) and unsupervised (Audibert, Michiardi, Guyard, Marti, and Zuluaga, 2020) approaches. Labeled normal and abnormal data are prerequisites for supervised methods, also known as 2-class classification. This type of method has the advantage of higher accuracy since more insights can be deduced from both classes but loses advantages in industrial applications where abnormal data is scarce. Semi-supervised methods only require data from one class, usually the normal class, to train the underlying model and classify new data based on its similarity to the normal class. This approach is ideal for industrial applications where normal data is abundant. Unsupervised methods share the same advantage with semi-supervised methods since there is no requirement on data labels. These categories of methods learn the characteristics of normal and abnormal data through training. Its application is more effective when a representative data population with both normal and abnormal records are present.

In the PdM space, there are oftentimes abundant data under normal conditions and either limited or no data associated with that of abnormal conditions. Semi-supervised and unsupervised methods with no requirement on availability of

failure data are therefore practical choices (Sgueglia, Sorbo, Visaggio & Canfora, 2022). Moreover, methods with less assumptions on data distribution, time dependencies and correlations, generalize better to a wide variety of applications and should be prioritized for implementation (Shaukat et al., 2021). Explainability is another critical aspect as it closely relates to the success of user adoption. Solutions that can assist end users in understanding the underlying causes of prediction results are preferred over black box solutions.

In this work, we present a semi-supervised time-series anomaly detection framework that is designed to incorporate a plethora of anomaly detection algorithms in a fine-grained manner for industrial predictive maintenance applications, to achieve the following objectives:

1. Generalize to industrial applications where normal operation data is abundant but failure data is unavailable or limited.
2. Perform under known conditions of sensor noise, heterogeneous data across assets (Cho, May, Tourkogiorgis, Perez, Lazaro, Maza & Kiritsis, 2018), and diverse asset operation profiles and conditions.
3. Offer efficient and flexible model management options when deployed to production as an end-to-end application (from data ingestion to insight consumption by end-users).
4. Provide explainable evidence into model prediction results for end users in order to enhance user confidence and information that can assist Subject Matter Expert (SME) investigation.

2. APPROACH

2.1 The anomaly detection framework

The proposed anomaly detection framework is presented in Figure 1. Data quality check and qualification are built in for both training and prediction to ensure accurate downstream results. For example, qualifying training data corroborates proper establishment of normal baseline profile. Feature engineering is the key step that extracts features potentially capable of manifesting discriminative signatures under abnormal conditions. This process calls for a collaborative effort among SMEs and data analytics practitioners in exploring useful features based on prior knowledge, experience, or physics-based modelling. We also recommend keeping this process expandable so that more features can be included into the production system as practitioners gain knowledge on feature correlations with abnormal symptoms from analyzing abnormal events that happen later. The feature engineering process can also handle the task of converting one category of anomalies to another. Specifically, contextual anomalies can effectively be

converted to point anomalies or collective anomalies with context incorporated numerically into the feature space. Our case study targets both point anomalies and collective anomalies. As machine degradation tend to happen over varied time periods, PdM practitioners are interested in abnormal data trends. When data is aggregated in timeframes into features, the collective signature can inherently be included with consideration of how data changes over the defined timeframe. For example, when data trend is a consideration for collective anomalies, features including slope and trendline etc. per aggregation period can transform collective behavior into point-wise behaviors.

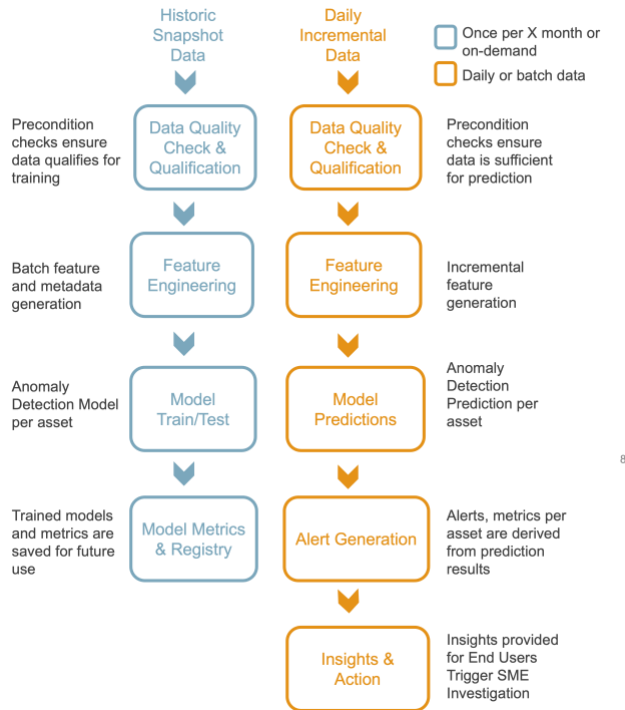


Figure 1. The fine-grained semi-supervised anomaly detection framework

In the next stage, a semi-supervised anomaly detection algorithm can be selected for training considering performance metrics. In this work, we utilized an anomaly detection model class developed in house that builds baseline profiles for normal data and measures deviations against newly collected data.

2.2 The Fine-Grained Modeling Approach

Although it is improbable for a single model to universally address the anomaly detection requirements for all types of assets, we recommend exploring expanding categories of models to accommodate various fields. During the early stages of model development, one can assess multiple types of anomaly detection models and down-select specific

models based on the performance metrics they exhibit. Meanwhile, for a given model type, there can be multiple model instances fit for individual differences among the population (product, device, equipment etc.); we refer to this as the fine-grained modeling approach. Essentially, the multiple model instances (spanning anywhere from tens to thousands) all belong to the same model type, with each having its unique model parameters and configurations.

When a specific anomaly detection model class is selected, there are two options for applying the model for a class of assets: 1) a unified model for assessing all assets and 2) a fine-grained model structure where each asset gets its own model instance.

Option 1 requires exploration and development efforts in data preprocessing and feature engineering to ensure that model inputs are scaled to be consistent across assets without losing information. This is possible in cases where established data standards are available or data from multiple assets can be scaled and conditioned to conform to a unified distribution. This assumption, however, may not suffice in all cases and can easily make the models vulnerable to changes in the field. For example, when maintenance is performed on a subset of assets, or sensors are replaced for assets in one cell but not all, data drift may be present for certain assets only (requiring flexibility to make asset-specific adjustment instead of population-wise model adjustment). Additionally, the interpretability of features with physical meaning may be lost during data scaling.

Option 2 is selected due to the benefits of flexibility during field usage and reduced effort to characterize data inconsistency across assets. Data quality is an important consideration for any modeling effort. In the industrial setting, additional logistics and complexity regarding data quality is involved due to challenges in data collection (sensor selection, mounting and degradation over time) and additional settings unique to this field (signal conditioning and scaling at the hardware/software layer etc.). A practical aspect of sensor selection is that oftentimes, businesses can go with cost-effective sensor options that might not yield the best data quality but still serve the application purpose (revealing anomalous patterns despite data noise). Moreover, not all assets may receive the same sensor settings/configurations and thus different noise attributes can be introduced to raw data. In this case, investing effort into making data uniform across assets can be futile. Consequently, a flexible approach to minimizing the effects of those individual differences may be more effective and risk adverse.

2.3 Performance metrics

Classical performance metrics for anomaly detection models include precision, recall and F-measure which is a weighted combination of precision and recall. In the predictive maintenance space, it is the anomalous predictions that trigger action, which means time and capital investment. Actions taken in accordance to true anomalies contribute to avoidance of unscheduled downtime and potentially hazardous events. Actions taken to investigate false anomalies result in periods of unproductivity, and in the long run, reduced confidence in deployed models. Moreover, businesses have preferences on what metrics are of higher importance in specific scenarios. Productionizing the framework takes collaborative efforts and understandings from not only data science and analytics personnel, but also stakeholders who fund and support such initiatives and end users who take actions on the prediction results to realize true business value and cost savings.

From our experiences, it is helpful to revisit the basics in regards to performance metrics in an effort to connect specific performance metrics to business related metrics, make it understandable across multiple personas, and highlight direct connection to business purposes. Meanwhile, those metrics need to have specific contexts based on the prediction goal. For example, an anomalous prediction produced with enough lead time allows for proper investigation, but late predictions can be unfruitful regardless of accuracy.

For PdM, the true positive metric (TP) and false positive metric (FP) can be evaluated for model validation. TP and FP need to be obtained from datasets of 2 conditions (prior-to-failure and normal operating condition). For semi-supervised approaches, we need to work with the fact that oftentimes there is no data around failure to evaluate true positive rate. The goal is therefore mainly to minimize the false positive rate (quantifiable) and at the same time maintain the detection sensitivity to unseen anomalies (validated upon data availability).

For the limited failure instances that are available, this data can be leveraged to evaluate the true positive performance metric for models combined with the false positive metric obtained from normal data evaluation. This process validates models of choice and provides evidence for model adoption when communicating to stakeholders and end users. In our case study, this validation stage is the checkpoint that determines whether the developed models qualify for deployment.

2.3.1 True Positive (TP) and False Positive (FP)

This metric is not available for assets with no failure records. If available, TP is the top priority metric for

evaluation as it represents detection of failure incidents. How to determine positive labels when they are not explicitly indicated?

Assumption: failure due to asset degradation happens over time for most industrial assets. We assume a proper failure developing period for failure progression. Therefore, data collected during this time prior to machine breakdown (according to plant maintenance records) is considered abnormal/faulty data. This assumption contains a failure developing period that can vary depending on failure modes. In our case study, we have combined SME (Subject Matter Expert) knowledge and historical data evidence to arrive at a reasonable assumption for labeling data prior to failure.

Given a failure event, during the defined failure developing period prior to the failure occurrence, if N anomalous predictions are made, on the premise that the earliest prediction provides a sufficient time window for remedy actions, the event is successfully captured. The higher TP the better, but it is still of value for end-users even if only one aggregated alert was generated to trigger action in time. For example, after applying the business logic of requiring 2 consecutive anomalous predictions to generate an alert, TP score larger than 7.69% (1 out of 13 days prior to failure) is a good metric if the first alert is generated in time (enough lead time for PdM team to take action), meaning that the event is captured prior to maintenance.

This metric needs additional data sources (inputs indicative of failure records or failure data directly as input) aside from training data.

FP is a direct measure of false alarms. This is a must-have metric for model evaluation, registry, and optimization.

The general understanding is that lower FP rates are more desirable. However, this only applies to scenarios when there is a path to validate other metrics such as TP such that there is confidence in detection sensitivity of the model. Having a FP close to 0 is risky given no evaluation of TP because the model then runs the risk of low sensitivity to anomalies.

2.3.2 Business Logic for Alert Consistency

In practical scenarios where anomalous artifacts manifest themselves in the feature space within a certain period before failure occurs, applying business logic to integrate alert consistency can improve robustness of alert predictions and decrease FP. Here we adopt the business logic of accumulating N days of continuous anomalous predictions before the system generates an alert that calls for remediation.

With business logic included, here we have 2 sets of TPs and FPs. TP_{raw} and FP_{raw} are the metrics before applying business

logic; TP_{logic} and FP_{logic} are the metrics after applying business logic.

With the above 4 metrics, performance evaluation can now focus on TP_{logic} and FP_{logic} as alerts for anomalies in actual applications are generated after applying business logic.

FP_{raw} can be leveraged to resolve the previously mentioned risks of having models that generate minimum or even 0 false positive rates but are potentially inefficient in detecting anomalies.

2.4 Model Optimization Metric

In this section, we construct an objective function to represent the model optimization metric. This objective function is the loss function that the optimization process attempts to minimize. During the training stage, with access to only normal data, we can obtain FP_{raw} and FP_{logic} introduced in the last section. The objective function can be defined as:

$$f_{min} = \sum_{i=1}^n (FP_{logic} - FP_{raw}) \quad (1)$$

Where i is the evaluation period and n is the total number of evaluation periods.

With the above definition, the false positive rate after applying business logic is minimized which translates to a reduced number of false alarms for end users. At the same time, it blends in the requirement to increase the false positive rate for raw prediction, causing the model to increase the probability of classifying data as abnormal, which equals to increased sensitivity to anomalies. Therefore, a combined effect in the objective function will allow the optimized model to achieve high classification performance in terms of minimized false alarms while also remaining sensitive to anomalies. To avoid optimization effect resulting in high FP rate, we found it effective to combine the optimized parameter set with empirical threshold of the risk score.

Weights can be assigned to each component of the f_{min} metric if there is a desire to adjust current effects. It is also worth noting that this metric is designed for collective anomalies that are of major interest in our applications.

f_{min} is used as the optimization metric for a Bayesian optimization approach implemented in Hyperopt (Bergstra, Yamins & Cox, 2013). Hyperopt is a distributed asynchronous hyper parameter optimization library. A configuration space for all model parameters can be specified and properly described (uniform, log-uniform, quantized log-uniform and categorical). The Tree-structured Parzen estimator Approach (TPE) is leveraged for optimization which can improve upon prior parameter evaluations and the corresponding loss function during iterative model evaluation. The TPE algorithm scales linearly in the number of variables and the optimization efficiency is greatly improved based on past observations.

3. CASE STUDY

3.1 Systematic Implementation in Predictive Maintenance of Cooling Pumps

In this case study, we implement the proposed framework for predictive maintenance of pumps supplying cooling and circulation fluid for machine tools on the shopfloor. Those pumps are of relatively low cost but represent a large portion of assets on our shop floor, accumulatively resulting in expensive unscheduled downtime and operation interruptions. Since we have 500+ of these pumps at one site alone, enabling predictive maintenance capability for this large population of assets can contribute to avoiding a great number of unscheduled downtime and reducing manual hours around issue tracking & investigation. More importantly, the developed framework is extendable to other type of assets where corresponding feature engineering techniques can be applied to reveal degradation signatures for individual assets. Therefore, a generalizable framework with modular software design, extendable to include new types of models, are built to satisfy the business needs. Figure 2 represents the production implementation of the proposed framework in Figure 1.

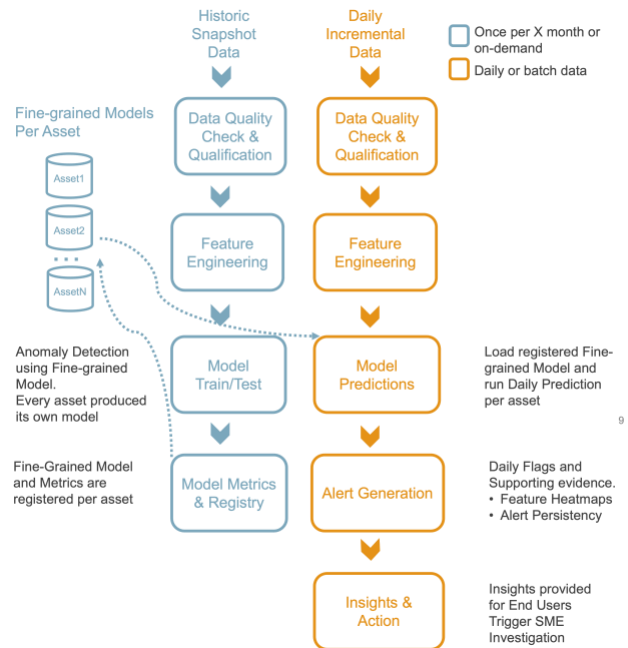


Figure 2. Production System Flow of the Framework

The semi-supervised anomaly detection framework is applied to historical data to evaluate performances, especially before maintenance events. The ingested data includes velocity and acceleration of 2 axes. Data quality criteria are established to account for the expected signal value range, the number of missing records, percentiles of signal values as well as data drift measurements. Furthermore, data qualification measures are incorporated to ensure that training data is

representative of normal operation and daily incremental sensor data qualifies to enter the prediction pipeline. Feature engineering is performed with the raw vibration data to obtain statistical features as well as features reflecting signal trend over pre-defined time intervals. In this way, potential collective anomalies can be transformed into point anomalies. The anomaly detection approach used in this case study considers the statistical distribution shift between baseline and test data. Fine-grained models are trained and registered per asset, which provides an advantage for flexible and low-cost retraining per asset. In the events that can potentially lead to data drift, including asset maintenance and replacement, model retraining should be triggered. The alert generation module outputs feature heatmaps (example in Figure 5) and alert persistency metrics. These prediction results are the data-driven evidences for insights and action.

Figure 3 shows the predicted risk score for an asset* (time axis is shifted for data privacy consideration). Daily risk score is a direct model output in the range of 0 to 1, with a higher value indicating a higher risk of being classified as abnormal. As can be observed, 2 weeks before asset failure, there were continuous records of high risk scores, consistently indicating a potential issue. These values are distinctly higher than other periods of time in history when there were no reported maintenance events according to plant maintenance records.

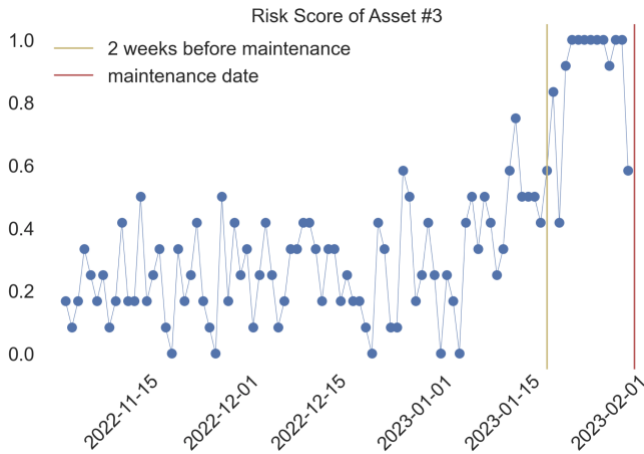


Figure 3. Predicted Risk Score of Asset #3

Here we assume a failure development period of 2 weeks and apply the business logic of requiring 2 continuous days of risk scores crossing thresholds to activate an alert. The resulted alert history is presented in Figure 4.

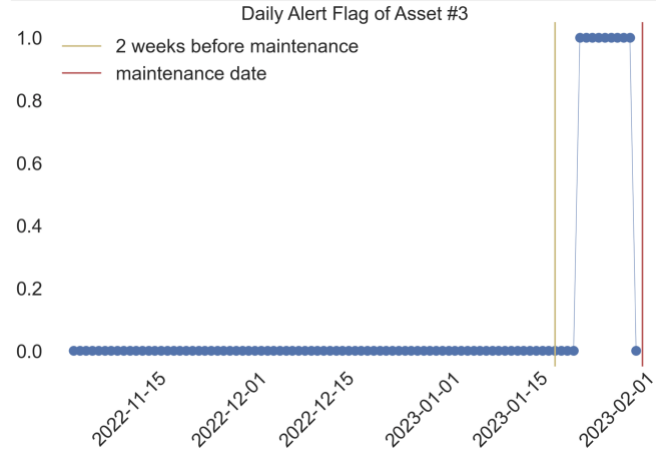


Figure 4. Predicted Daily Alert Flags of Asset #3

For Asset #3, an extended period of time before maintenance is used to evaluate FP and 2 weeks before maintenance is used to validate TP. Outside of the assumed failure development period, there were 0 days with risk scores over the threshold (0.8), therefore FP_{raw} is 0. Within the assumed failure development period, there were 11 days out of 14 days where predicted risk scores exceeded the threshold. After applying the business logic, there were 9 days out of the 13 days before the event where alerts were generated. There were no alerts generated for the normal operation period under study. Table 1 illustrates the metrics for the cited event of this asset. The event was successfully predicted with sufficient lead time (>1 week) to trigger action.

Table 1. Performance Metrics of Event for Asset #3

Performance Metric	TP_{raw}	TP_{logic}	FP_{raw}	FP_{logic}
Value	78.5%	69.2%	0	0

It is worth noting that once a failure is developing, it might not always manifest in data (e.g., abnormal vibration for pumps), due to effects of other operating parameters (e.g., feedrate for machine tools, cutting depth etc.). Therefore, we expect to see inconsistent alerts prior to failure and consider any TP_{logic} larger than 7.69% to be a meaningful prediction that can effectively trigger intervention action and generate business value. The bottom line is as long as the developing issue manifests at certain times during the assumed failure development period, the method can identify the failure and generate alerts.

3.2 Supporting Evidence

Easy-to-digest supporting evidence is provided for end users to gain insights into why certain predictions were made, especially when receiving anomalous predictions. Figure 5

shows an example of supporting evidence for the predicted anomalous event in the previous section. This is a time-series feature heatmap indicating which features have deviated from the baseline as determined by the optimized model. Features with high deviation are highlighted while features conforming to the baseline are black. As can be observed, there are continuously high feature deviations from baseline before anomalous predictions and end users know from this visualization which features have contributed as well as how long the underlying risk has been present. Moreover, all features generated here have physical meanings corresponding to the asset operations which end users can interpret. These insights equip practitioners with the ability to make informed decisions on remediation actions and can contribute to the long-term adoption of production systems.

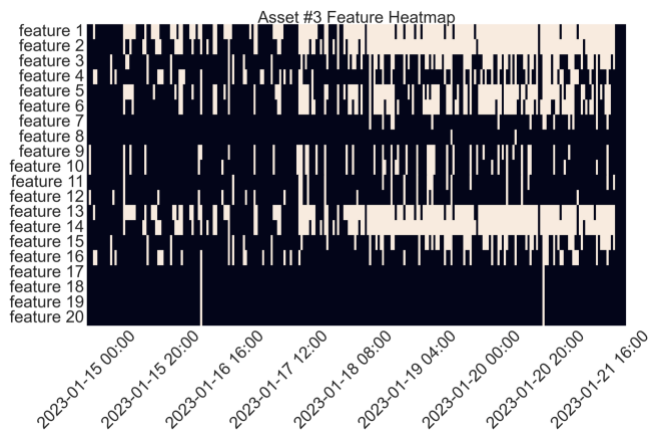


Figure 5. Feature heatmap of Asset #3

4. IMPLEMENTATION CHALLENGES AND RECOMMENDATIONS

4.1 User Journey to Value

There is a consensus in the understanding of the value predictive maintenance can bring for businesses when data is transformed into insights. It is equally, if not more, important to emphasize that the final value realization is achieved through users taking actions on the insights. For example, when a shop floor maintenance team receives an asset alert and takes action to schedule maintenance in a timely manner instead of letting the asset run to failure, business value is created by preventing unscheduled downtime. Therefore, user adoption of the deployed system is a critical aspect of the implementation.

Here we walk through the user journey of the consumption of model results for specific key scenarios citing the case study and provide our recommendations.

4.1.1 False Alarm

Model prediction: false positive

Key scenario 1:

Action: Asset check-up is performed, confirmed no issue with the asset and the asset continued to run with no issue for >1 month afterwards.

Personas: predictive maintenance technician (PdM), machine maintenance tool service team (MTS), operator

Cost estimation: about 3 hrs. combined from PdM and MTS

This is normally performed during scheduled downtime (instead of unscheduled downtime), so no extra cost for operation interruption.

Recommendation: Fine-grained model for this asset should be retrained to improve accuracy and ensure that the re-trained model performs better than current version. This requires the implementation to provide a model retraining option. Investigation should be performed if model retraining doesn't solve the issue. In this case, consider including a different model type to enhance detection.

4.1.2 Success prediction of An Event

Model prediction: true positive

Key scenario 2:

Action: Asset degradation is known via alerts. The asset continued to run until failure.

Business value: No value added if there are no operational adjustments.

Key scenario 3:

Action: After receiving alerts, critical manufacturing tasks are scheduled for this asset.

Personas: production team

Business value: avoided possible product delivery delay and key operation interruptions.

Time to re-arrange for production: can be a shift time (4-8 hours depending on complexity of set-up). This time is invested in order to minimize the risk of running critical manufacturing tasks on a machine with an alert status.

Key scenario 4 (Recommended):

Action: Given an alert for asset degradation, an asset check-up is performed, confirming an issue with the asset. The asset is then scheduled for maintenance/replacement at a convenient time.

Personas: PdM, MTS, operator, production team

Business value: Avoided unscheduled downtime and operation interruptions enabled by model (varies depending on asset type, estimated >8 hrs.)

4.1.3 Missed Event

If the deployed model fails to detect an event, we recommend recording this for model improvements. It is expected that

deployed models may have missed events and false alarms and these field-application responses should be incorporated for continuous development and continuous integration for a live production system to grow and mature. Given a missed event, model retraining can be triggered to produce another model version with better performance metrics based on historical data. This can be another fine-grained model of the same type or of a different model type. Experiences here can also contribute to improved model retraining guidelines such that future model retraining can be triggered automatically in the presence of performance drift, thus preventing missed events with an enhanced version of the model.

4.2 Metrics Capture for Production System

4.2.1 Business Metrics

We recommend defining the format and mechanism to capture user feedback automatically for evaluating value creation and model improvement. In an ideal case with seamless connection to Enterprise Resource Planning (ERP) system that records plant maintenance details, maintenance events can be automatically pulled from these systems at a certain cadence, combined with model performance evaluation to update model performance metrics. Business metrics for predictive maintenance can cover TP, FP and FN for each event (one metric value per event). ERP system records delay estimates for similar types of events. These can be combined with TP to quantify realized business value. FP and FN are measures for analytics teams to assess negative costs, make model improvements, or explore other model types for better accuracy. This is a key step that brings the business value enabled by predictive maintenance from estimation to realization.

Below is a brief explanation of how each metric corresponds to field operations:

TP: Captured event, confirmed a maintenance needed after receiving alert and going through a pre-defined list of actions.

FP: False alarms, confirmed no need for maintenance after receiving alert and taking action to investigate.

FN: Missed event, there was a major event on the monitored component that is not human-error related, but users never received an alert.

4.2.2 Production System Metrics

These metrics provide insights on how well the implemented production system runs and if it meets business expectations (e.g., predictions need to arrive in time for users to take action). Consider time of input, time of output, errors, and resources used for the operation. To enable drilldown, all metrics are enriched with metadata such as function, steps, and job so they can be used for tracing details. Table 2 lists some of the metrics used in our implementation:

Table 2. Examples of Production System Metrics

Production System Metric	Indication
Input On Time	Measures if all input is available at the time of run
Results On Time	Measures if the prediction or alert is provided on time
Count of Error/Warning/Retries	Measures the health status of system execution
Execution Time	Wall time of every step and job. Can increase CPU or optimize code to lower execution time if needed.
Total Input/Output Rows & Columns	Measures the amount of data to be processed
Total IO Read/Write	Measures the efficiency of data movement for a given input size
Average CPU Utilization	Measures usage of CPU resources. Low CPU utilization indicates non-efficient usage.

4.2.3 System Troubleshooting Pattern

After capturing historic System Metrics, trend can be used to help troubleshooting. One approach is to check for the deviation from moving averages of sample metrics. To enhance traceability, metrics are tagged with metadata such as names of workflow steps, functions and jobs etc. This provides the ability to aggregate and filter metrics following by scoring/grading for system health (normal – within threshold, warning - works but above average error and retry, failed with errors).

For example, for each step/functionality below, it would be beneficial to stop the process from continuing if a step is unhealthy. Otherwise, downstream results will be unreliable and a waste of execution costs.

- Data Ingestion part (ETL) – There should be sufficient data landing on time, otherwise, downstream features will be calculated as nan or zero or even error. When data does become available with delayed arrival, the system can re-run, but will it meet business needs (e.g., late prediction may not allow enough time for PdM to take action)?
- Feature engineering – The number of rows, columns written should be consistent per model. Anything out of the ordinary should trigger a data audit. Predictions should not occur for models with missing features unless accounted for in deployed models and proven to still yield valid prediction results.

- Model prediction – Check if the number of predictions generated is consistent with trend. Did it meet business expectations?

With the above metrics and the corresponding historical trends for reference, personnel in charge of troubleshooting can effectively identify root causes and bring in the right resource for corrective action.

5. DISCUSSIONS AND FUTURE WORK

This work demonstrates the end-to-end implementation for predictive maintenance in the industrial 4.0 space. As mentioned in previous work discussing program implementation for scalability across manufacturing sites (Tong, Bakhshi, & Prabhu, 2022) or product domain (e.g., predictive maintenance for key components of aircraft), scalability calls for systematic implementation of all stages including data collection, analytics algorithms, deployment tools and operational adoption. After a successful pilot implementation at one site, lessons learned can be leveraged as a lighthouse for other sites to follow. Data collection can be enabled both at the manufacturing site (especially for legacy equipment, add-on sensors can be installed for identified needs) or on the equipment manufacturer side as an additional business service (for new equipment which can come with pre-installed sensors ready to connect to customer's data collection instrumentation). With growing technology maturity and established successes in PdM, part of the analytics can be adopted at the edge (especially for fine-grained applications where minimum communication between assets are required), saving substantial cost in cloud data storage and system maintenance.

6. CONCLUSIONS

This paper describes a fine-grained semi-supervised anomaly detection framework that addresses the practical challenges of heterogeneous data across industrial assets and the scarce failure data sources. A case study on predictive maintenance of cooling pumps is presented covering an end-to-end implementation of the proposed framework. Practical implementation challenges and proposed recommendations including user adoption to ensure estimated value is realized, metrics for production system assisting with business value capture, and production system maintenance and troubleshooting are discussed with detailed examples.

REFERENCES

- Ahmad S., Lavin A., Purdy S., Agha Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262, pp. 134-147. doi: doi.org/10.1016/j.neucom.2017.04.070
- Bergstra, J., Yamins, D., Cox, D. D. (2013) Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures.

- Proceedings of the 30th International Conference on Machine Learning*. PMLR vol. 28(1), pp. 115-123
- Cavalieri S, Salafia MG. (2020). A Model for Predictive Maintenance Based on Asset Administration Shell. *Sensors* 20, no. 21: 6028. doi: <https://doi.org/10.3390/s20216028>
- Cho, S., May G., Tourkogiorgis I., Perez R., Lazaro O., Maza B.D.L., Kiritsis D. (2018). A Hybrid Machine Learning Approach for Predictive Maintenance in Smart Factories of the Future. *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0*. APMS 2018. vol 536. Springer, Cham. https://doi.org/10.1007/978-3-319-99707-0_39
- Foorhuis, R. (2021). On the nature and types of anomalies: a review of deviations in data. *International Journal of Data Science and Analytics*, vol. 12, pp. 297–331. doi: <https://doi.org/10.1007/s41060-021-00265-1>
- Jiang, J., Kao, J. & Li, Y. (2021). Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning. *Applied Sciences*. vol. 11, no. 15, pp. 6698. <https://doi.org/10.3390/app11156698>
- Audibert, J., Michiardi, P., Guyard, F., Marti S., and Zuluaga, M., A. (2020). USA: UnSupervised Anomaly Detection on Multivariate Time Series. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '20)*. Association for Computing Machinery, New York, NY, USA, pp. 3395–3404. <https://doi.org/10.1145/3394486.3403392>
- Sgueglia A., Sorbo A. D., Visaggio C. A., Canfora G. (2022). A systematic literature review of IoT time series anomaly detection solutions. *Future Generation Computer Systems*. vol. 134, pp.170-186, ISSN 0167-739X, doi: <https://doi.org/10.1016/j.future.2022.04.005>.
- Shaukat, K., Alam, T. M., Luo, S., Shabbir, S., Hameed, I. A., Li, J., Abbas, S. K., & Javed, U. (2021). A Review of Time-Series Anomaly Detection Techniques: A Step to Future Perspectives. *Advances in Information and Communication. FICC 2021. Advances in Intelligent Systems and Computing*, vol 1363. Springer, Cham. https://doi.org/10.1007/978-3-030-73100-7_60
- Tong, X., Bakhshi, R., & Prabhu, C. (2022). Industry 4.0 for Aerospace Manufacturing: Condition Based Maintenance Methodology, Implementation and Challenges. *Annual Conference of the PHM Society*, 14(1). <https://doi.org/10.36001/phmconf.2022.v14i1.3189>
- Teng, X., Lin, Y., & Wen, X. (2017). Anomaly Detection in Dynamic Networks using Multi-view Time-Series Hypersphere Learning. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17)*. Association for Computing Machinery, New York, NY, USA, pp 827–836. <https://doi.org/10.1145/3132847.3132964>

BIOGRAPHIES

Xiaorui Tong was born and raised in Pingliang, Gansu, China. She obtained a B.S. in Mechanical Design, Manufacturing and Automation in 2011 from Central South University in Changsha, Hunan, China, and a M.S. in Mechanical Engineering at the Center for Intelligent Maintenance Systems in 2013 from University of Cincinnati in Cincinnati, Ohio, USA with a research focus on prognostics and health management (PHM). She received her Ph.D. in Material Science and Engineering in 2018 from West Virginia University in Morgantown, West Virginia, USA where her research on building computational models combining machine learning and material science principles for renewable energy applications was featured as the cover article by the Journal of American Ceramic Society. She worked with companies including the Goodyear Tire & Rubber Company and Parker Hannifin Corporation on developing computational models for various industrial systems. She worked at the FORCAM Inc. on enabling early Industry 4.0 capabilities for manufacturing equipment in collaboration with multiple manufacturing facilities. She is currently an Associate Director of Data Science & Analytics in the AI/ML Solutions team within Applied Data at Raytheon Technologies. Her work focuses on leveraging domain knowledge, computational modeling, and artificial intelligence to enable insights for industrial processes and products.

Wee Quan Jung was born and raised in Guangdong, China and attended schools in US. He obtained a B.S. in Computer Science from University of Pennsylvania in Philadelphia, Pennsylvania in 1998, he worked in various applications from Voice Over Internet Protocol (VoIP) before the technology was widely commercialized and big data analytics, mobile chat applications, large-scale

microservices for vendor integration and mobile app gateways. He has built core architectures for telecom services and mobile apps using distributed multi-region applications, reaching 250K+ users. He currently works at Raytheon Technologies in AI/ML solution and has achieved the title of Distinguished Engineer. He has led teams in building large-scale production systems running thousands of models for predictive maintenance of a fleet of engines (1K+). As a technology developer for highly efficient and scalable production software systems, his work focuses on strategic implementation for optimization and simplification of such systems.

Jeremy Frimpong Banning was born and raised in California, United States. He obtained a B.S. in Electrical Engineering in 2005 from the University of California, San Diego and a M.S. in Applied Mathematics at San Diego State University in 2011 with a concentration in non-linear dynamical systems and chaos. He has worked with several companies within the defense and aerospace industry including Leidos, Sikorsky Aircraft, Lockheed Martin, and United Technologies. He is currently an Associate Director of Data Science & Analytics and leads one of the AI/ML Solutions teams within Applied Data at Raytheon Technologies. His primary focus lies in Engine Health Monitoring and predictive maintenance endeavors, where he and his team utilize machine learning and facilitate the widespread adoption of cloud technologies. His aim is to empower teams in developing resilient and scalable solutions across the enterprise.