Building a Scalable Intelligent System to Advise Predictive Maintenance Operations in a Steel Mill

Digital technologies are transforming industry at all levels. Steel has the opportunity to lead all heavy industries as an early adopter of specific digital technologies to improve our sustainability and competitiveness. This column is part of AIST's strategy to become the epicenter for steel's digital transformation, by providing a variety of platforms to showcase and disseminate Industry 4.0 knowledge specific for steel manufacturing, from big-picture concepts to specific processes.

Authors

Anshuman Sahu
Noodle.ai, San Francisco, Calif., USA

Ravikant Chahar
Noodle.ai, Bengaluru, KA, India
ravikant.chahar@noodle.ai

Santiago Olivar
Noodle.ai, San Francisco, Calif., USA

Ravi Balasubramanian
Noodle.ai, Bengaluru, KA, India

Atin Gupta
Noodle.ai, San Francisco, Calif., USA

Hyungil Ahn
Noodle.ai, San Francisco, Calif., USA
hyungil.ahn@noodle.ai

The continual increase in demand for high-quality products has spurred the growth of modern factories. One of the key requirements to meet this rapid demand is to optimize operational efficiency through reduced downtime. This is evident from a study by the International Society of Automation,[6] which has shown that unplanned downtime across all industry segments is estimated to cost US$647 billion per year. As a conservative estimate, an hour of downtime in a steel mill could lead to ~US$50,000 in lost revenues and a typical steel mill has unplanned downtime upwards of 500 hours in a year. One major driver behind this is related to the maintenance of industrial equipment. The traditional approaches to maintenance are mostly reactive or periodic in nature based on component usage. The reactive maintenance approach performs maintenance operations only when equipment fails. While this doesn't entail intermittent equipment downtime, it can lead to expensive capital expenditure and unexpected downtime. The scheduled (or periodic) maintenance approach performs maintenance operations at a regular cadence. This can contribute to extra costs in terms of frequent maintenance interventions that may not be relevant.

The advancement of sensing and data acquisition technologies along with their robust performance in extreme conditions has led to their widespread adoption in manufacturing shop floors.[1,2] This, coupled with innovation in machine learning, has fueled a newer paradigm of predictive maintenance, which provides an opportunity to overcome the inefficiencies of the traditional approaches. The predictive maintenance approach implements smart, dynamic and scalable strategies informed through monitoring the health and usage of equipment. This has led to a growing number of organizations investing resources toward modernizing their current maintenance strategies. Based on a recent study by PricewaterhouseCoopers,[7] 95% of the surveyed companies shared that the adoption of these strategies contributed to improving key performance drivers. 60% of the companies reported an average improvement of at least 9% equipment uptime and other benefits in terms of improving equipment lifetime, reducing health, safety and environmental risks.

The asset-intensive steel industry has also been adopting predictive maintenance strategies as part of modernization and key competitive advantages. There is a need to build automated systems that can learn from the operational data to proactively guide maintenance teams. This paper will share key challenges faced in building and deploying such a system now running live on multiple critical equipment within a steel mill in one of the largest steel manufacturers in the world.

Background

Monitoring and predicting the health status of critical equipment is an essential ingredient of the predictive maintenance strategy in the context of smart manufacturing. With the rapid evolution of information processing both at the edge and the cloud, it enables the delivery of results at near real operational time. Recent advances in machine learning and deep learning have demonstrated a growing number of successful algorithms in machine health prognosis. Instead of manually encoded rules, these algorithms learn the evolution of machine conditions to generate predictions.[3,4] Interested readers are encouraged to refer to References 3 and 4 for a comprehensive review of machine learning and deep learning methods that can be applied.

The work presented in this paper combines unsupervised and supervised machine-learning algorithms. The time-series sequence data that is fed to these algorithms consists of sensor features (independent variables) and a target label (dependent variable). The set of independent variables can be extended to include other relevant process information. Unsupervised algorithms operate only on the set of independent variables to often identify interesting regions spanned by a subset of those variables. These interesting regions are determined by their relationship with the process state in order to ensure interpretability. On the other hand, supervised algorithms require a target label such as the failure event time stamp in this case. The goal then is to predict the target label or some function of the target label from the space spanned by the independent variables. It is emphasized that the right choices between unsupervised and supervised algorithms are often dictated by the complexity of failure dynamics as well as some critical modeling challenges that will be discussed in the next section.

Modeling Challenges in Predictive Maintenance

Designing and operationalizing an effective solution for maintenance teams requires several considerations. While building a data-driven approach, the following challenges are encountered from modeling standpoint, which will be described subsequently:

- There is a need to analyze streaming data from multiple sensors in near real time.
- Sensor data in production environment is noisy and shifts under different operational regimes.
- It is critical to characterize the breakdown modes of components in sufficient resolution.
- It is important to predict the breakdown event in advance so that the maintenance team can act.
- A limited number of failure labels or downtime events play a role in choosing the right modeling approach and determine the overall performance of the models.
- The stakeholders need insights into the behavior of key sensors contributing to the breakdown.

Multi-Stream Sensors — For any component, multiple sensors are monitored since the relevant failure mechanism (mode) often manifests as a multi-variate pattern. This implies the model should consider the dependencies between the sensors as opposed to treating them independently. These dependencies are challenging when one considers the fact that sensor values are sampled at an extremely high frequency. For instance, the sensor data used are sampled at high frequencies (for instance, 10 milliseconds). As models are built across multiple failure modes, this accentuates the need for a scalable system to train and deploy these models.

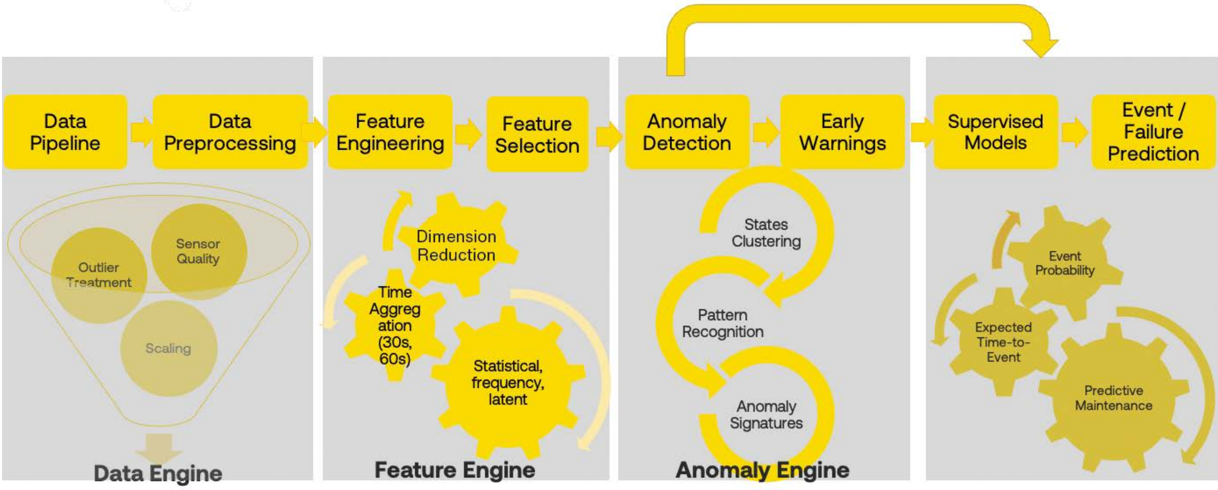
Sensor Behavior During Production Runs — Steel mill production runs entail multiple batches manufactured every day. As a result, the sensor time-series readings are essentially non-stationary and can vary significantly between runs. Thus, the operational context, including but not limited to setpoints, heat/product characteristics and other operating variables, needs to be encoded along with the sensor data.

Failure Mode Characterization — Components can fail through multiple mechanisms or failure modes. It is important to prioritize the failure modes that are critical for the operation of a component. As discussed previously, this is facilitated by understanding how to characterize the occurrence of failure in terms of sensor data. This ensures consistency of failure mode labeling in a component and across components.

Breakdown Events and Number of Failures — The historical component failures are accompanied by recorded events where an operator took actions. These action sequences are helpful in identifying the relevant failure modes and understanding when the failure events occurred. This, combined with the failure mode characterization, is of significance in practice as it impacts the accuracy of the failure labels, given that the number of failure events is relatively small across component(s). It is critical to predict these events sufficiently in advance (hours/days as opposed to minutes before the event) so that the maintenance team can proactively act on the prediction.

Model Interpretability — During the model training phase, the system learns from the sensor behavior leading to past failure events. During the inference phase (live runs) in near real time, it predicts whether any anomalous patterns are occurring, and the expected time for when the failure event would happen. For the maintenance teams to act on these early warnings, it is important to identify and share with them the key contributing sensors that are relevant to the predicted failure event. This helps them better understand the system output and builds trust.

Figure 1



High-level overview of asset health modeling pipeline.

Intelligent Asset Health Application

Given the challenges described in the previous section and considering the large number of assets that need to be monitored, a systematic approach was adopted for building and deploying the models. The model-building process was decomposed into several steps involving data pre-processing, feature generation and model development, as shown in Fig. 1. The system architecture allows for each step to be configured for an asset and the steps stitched together into a pipeline for training and deploying the models. The data pre-processing step considers the input data quality in terms of sensor scale, noise, outliers and special conditions corresponding to the physical process. The feature generation step considers feature engineering and feature selection. Both the data pre-processing step and the feature generation step allow for handling of challenges from "sensor behavior during production run." The model-building step consists of two phases: the first phase covers Noodle.ai's anomaly detection model (FlowOps Sentinel) that captures the onset of anomalies specific to the failure mode in the asset; the second phase covers Noodle.ai's prediction model for expected time to failure (FlowOps Precog). The models can capture multi-variate interaction between different sensors as well as within each sensor at different time resolutions. Additionally, the prediction model can also use outputs from the anomaly detection model. The pipeline construct also allows one to choose model hyperparameters that provide best performance.

Model Results on Different Use Cases

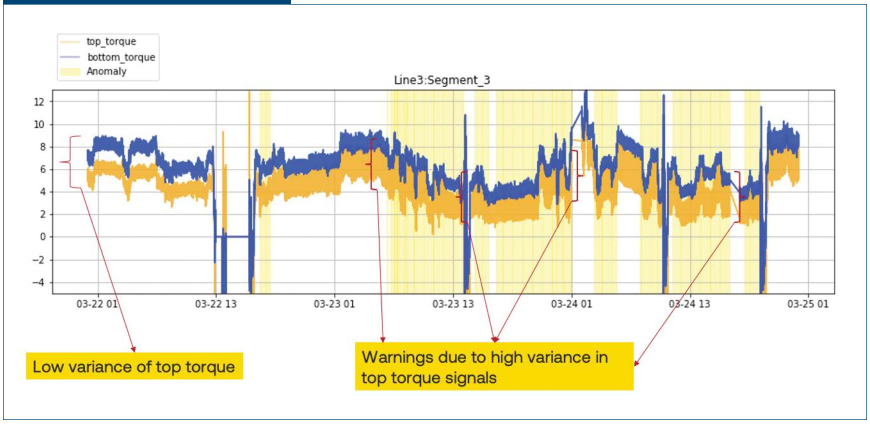
Here applications of the model to a few real-life use cases in the steel mill are shared. For confidentiality purposes, the signal names have been made anonymous in the plots.

The first use case is regarding Cardan shaft (decoupling) failures as shown in Fig. 2. There are 26 such components within the continuous caster, and more than 20 signals per component. Using the anomaly detection pipeline, it was identified that the torque signals exhibited very high variance relative to normal operation. Figs. 3 and 4 show time-series plots that capture the warnings and highlight the abnormal behavior. Based on advanced warnings, the

Figure 2

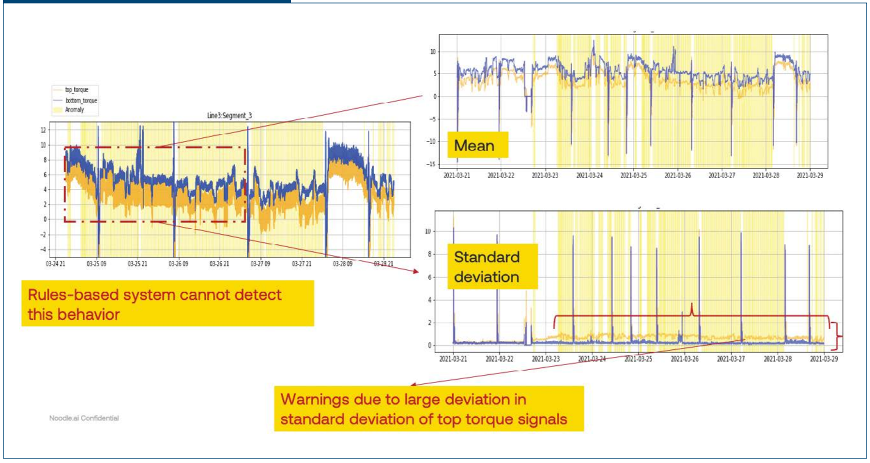
Cardan shaft assembly and a broken shaft.

Figure 3



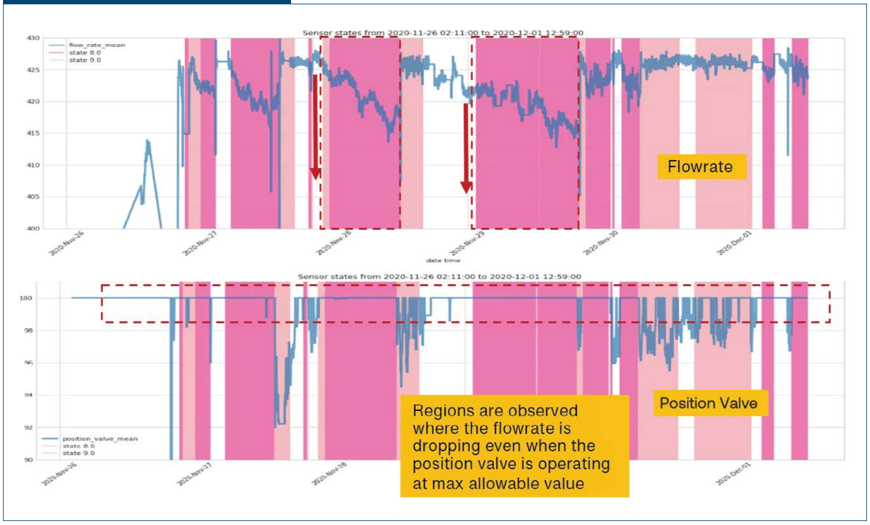
Warnings related to decoupling failures in a particular component.

Figure 4



Warnings related to decoupling failures that highlights the large variance behavior.

Figure 5



Warnings related to mold plate cooling loop blockage.

maintenance team took proactive action by disengaging the gearbox and moving away the roller to avoid the component failure before a planned inspection stop. On several occasions, warnings came a week before the planned stop, giving the maintenance team advance notification to have the part replacement readily available.

The next use case is regarding blockage in a cooling loop that could lead to melting of mold plates and catastrophic failure in line. There are eight cooling loops for mold plates within the continuous caster with more than 24 signals per loop. Pieces of loose metal inside the cooling loop can lead to a valve blockage. Depending on their position, these loose metal pieces could lock onto the stem and prevent the valve from closing. This severely impacts the ability to cool the mold plates, leading to catastrophic failure. Using the anomaly detection pipeline, it was identified that the valve position and flowrate signals deviated from their correlations relative to normal operation. Fig. 5 shows a time-series plot that captures the warnings. The warnings can be further characterized in terms of states (color-coded in different shades) where each state captures different intersensor relationships. The maintenance team acted by stopping operations, and then inspected the cooling loops, flushed the loops and found a significant amount of loose particles as shown, in Fig. 6. With proactive maintenance, they avoided a catastrophic event and were able to bring operations back on-line quickly.

In addition to generating anomalies, additional diagnostics are provided to clarify the anomalies. These include highlighting the relevant sensors that contribute to the anomalous patterns and the states that capture these intersensor relationships. Also, a criticality factor is assigned for prioritizing the maintenance action. Fig. 7 shows a sample output in tabular format (can be consumed for display via dashboard) for some of the use cases that are running live.

Once the system identifies the anomalies, it also provides an expected duration in which the component is likely to fail. This prediction is generated by combining the sensor patterns and anomaly characteristics (state, criticality, duration, etc.). Based on operational requirements, metrics (both on-line and post-event) are used to measure how useful and actionable these predictions are for the maintenance team. The on-line metric measures the overall deviation of a subsequent prediction from the previous prediction whereas the post-event metric measures whether the predictions align along the prediction horizon cone. Fig. 8 shows the generated predictions for the decoupling use case. The predictions are generated hourly (can be configured for user specified periodicity).

One of the challenges faced is the noise in recording the failure label. In order to understand the impact of this noise, delays in recording the failure label for the publicly available NASA engine failure data set were simulated.[5] One data set was randomly chosen and split into train, validation and test groups. Train and validation data sizes were changed to account for scenarios when only a subset of relevant failure data is available. For the train and validation data, varying amounts of perturbation were added to remaining time to event values to account for not accurately capturing the failure time. Two settings were chosen for this: small and large

Figure 6

Pieces of loose particles recovered by maintenance team upon flushing the cooling loop.

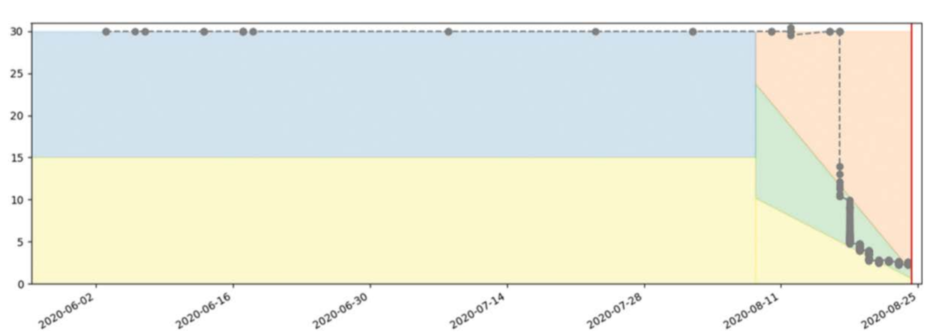
Figure 7

#	Time_start	Time_end	Asset	Use_case	Line (L3/L4)	Component (3-12)*	Type	Criticality	Sensor Contribution							
									Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
1	10-Aug-2020 00:00:00	10-Aug-2020 00:00:59	Continuous Caster 2	Segment Decoupling	L3	3	1	3	0%	50%	10%	0%	40%	N/A	N/A	N/A
2	10-Aug-2020 00:01:00	10-Aug-2020 00:01:59	Continuous Caster 2	Segment Gap	L4	7	2	3	N/A	N/A	N/A	N/A	N/A	20%	80%	0%
3	10-Aug-2020 00:03:00	10-Aug-2020 00:03:59	Continuous Caster 2	Segment Decoupling	L3	12	2	2	30%	0%	15%	45%	10%	N/A	N/A	N/A
4	10-Aug-2020 00:03:00	10-Aug-2020 00:03:59	Continuous Caster 2	Segment Gap	L3	11	1	1	N/A	N/A	N/A	N/A	N/A	50%	10%	40%

Anomaly Duration
Asset Hierarchy
Anomaly state
Warnings Criticality
Interpretability - Signal Importance

Sample output showing the key pieces of information related to anomalous behavior.

Figure 8



Time-to-event output predictions for decoupling use case in a particular component.

perturbation with seven and 25 cycles, respectively. Train and validation data were used for model training and hyperparameter tuning, respectively. The mean absolute percentage error between the predicted and actual value on hold-out test data (20 engines) was computed, which is not affected by the scenarios described here. Table 1 shows the results of the experiment. It was observed that for a limited number of failure label data, the increased noise in recording the failures results in higher prediction error. In order to have acceptable error bounds, this needs to be addressed.

Table 1

Experiment Results on NASA Engine Data Set

	No perturbation	Small perturbation	Large perturbation
Training-validation data	MAPE	MAPE	MAPE
Train data: 1 engine Validation data: 1 engine	67.13	86.75	127.28
Train data: 1 engine Validation data: 2 engines	65.08	81.53	125.01
Train data: 5 engines Validation data: 2 engines	43.27	59.95	97.41
Train data: 15 engines Validation data: 2 engines	32.49	56.97	101.48

Conclusions

Machine learning can guide maintenance operations across multiple failure modes in steel manufacturing processes. Building a scalable system for live production runs is challenging due to multi-stream sensors, noisy data and multiple operation modes. This paper described a system that addresses these challenges by learning dependencies within multi-variate sensors in unsupervised fashion to generate early warnings. The system learns temporal degradation patterns to predict the expected time to next breakdown. Patterns relevant to a failure mode are displayed by computing sensor contributions. Results of a real implementation in a steel mill were discussed, along with how this helps maintenance planning with proactive guidance.

As part of ongoing improvements, the authors are investigating different ways to improve the results. It should be highlighted again that the quality of failure labels is extremely important for these models. A key focus would be to improve the current process around capturing, recording and attributing failure events to the respective components. In addition, the authors are currently working on ingesting and utilizing other relevant information such as parts replacement events.

Acknowledgment

The authors would like to thank the maintenance teams and plant managers who provided valuable context for understanding the use cases. Their feedback was extremely useful in designing such a system. The authors appreciate their ongoing support and involvement as they strive to improve their results and create meaningful business impact.

References

1. Yang, H.; Kumara, S.; Bukkapatnam, S.; and Tsung, F., "The Internet of Things for Smart Manufacturing: A Review," *IISE Trans.,* Vol. 51, 2019, pp. 1190–1216.
2. Menezes, B.; Kelly, J.; Leal, A.; and Le Roux, G., "Predictive, Prescriptive and Detective Analytics for Smart Manufacturing in the Information Age," *IFAC,* Vol. 52, 2019, pp. 568–573.
3. Vogl, G.W.; Weiss, B.A.; and Helu, M., "A Review of Diagnostic and Prognostic Capabilities and Best Practices for Manufacturing," *J. Intell. Manuf.,* Vol. 30, 2019, pp. 79–95.
4. Biggio, L., and Kastanis, I., "Prognostics and Health Management of Industrial Assets: Current Progress and Road Ahead," *Frontiers in Artificial Intelligence,* Vol. 3, 2020.
5. Saxena, A.; Goebel, K.; Simon, D.; and Eklund, N., "Damage Propagation Modeling for Aircraft Engine Run-to-Failure Simulation," *Proceedings of the 1st International Conference on Prognostics and Health Management,* October 2008.
6. https://www.isa.org/standards-publications/isa-publications/intech-magazine/2013/feb/automation-it-predictive-maintenance-embraces-analytics.
7. https://www.pwc.be/en/documents/20180926-pdm40-beyond-the-hype-report.pdf.

This paper was presented at AISTech 2021 — The Iron & Steel Technology Conference and Exposition, Nashville, Tenn., USA, and published in the AISTech 2021 Conference Proceedings.