International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

Machine Learning Applications for Industry 4.0

Predictive Maintenance and High Conformity Quality Control

Aravind Potturu[a]*
[a]Enterprise Business Architect, Smart Factory,
Enterprise Performance, Deloitte LLP,
USA

Abstract:- Industry 4.0 is empowering manufacturing plants transform into Smart Factories by leveraging intelligent technologies. Artificial Intelligence and Machine Learning combined with IoT are the key enablers of this digital manufacturing transformation. This study outlines how Artificial Intelligence powered maintenance and quality control can help manufacturers to achieve operational efficiency.

keywords: Artificial Intelligence; Machine Learning; Industry 4.0; Predictive Maintenance; Quality Control; Computer Vision

I. INTRODUCTION

The term Artificial Intelligence (AI) is coined to any form of intelligence that is exhibited by non-humans. Nowadays, such form of intelligence – usually captured in computer software – can be obtained by utilizing Machine Learning (ML) algorithms. These algorithms are capable of processing large amounts of data, resulting in the detection of relevant patterns and insights which reside within the data. Given enough qualitative data, these algorithms can be trained to perform specific tasks that could previously only be carried out with the help of human intelligence.

Many industries have started to adopt intelligent algorithms to automate and enhance prediction making processes for a wide range of business applications. In the manufacturing industry – for example – such intelligent algorithms may provide additional value in a businesses' maintenance and quality control strategy. This paper will provide a series of technical implementations regarding the implementation of machine learning software for predictive maintenance and quality control support within the manufacturing industry.

II. PREDICTIVE MAINTENANCE

To reduce mechanical component failure, most manufacturing companies rely on the concept of *preventive maintenance* [1]: maintenance programs – including routine work, component change, reparation of detected components, and regular inspections – after a pre-defined duration of operations. However, classic maintenance schedules like preventive maintenance have some inherent shortcomings:

- **Over-Maintenance**: More than often, the inspected component is still in perfect condition, resulting in unnecessary maintenance procedures and component handling.
- **Reduced Manufacturing Capacity and Revenue Loss**: Preventive maintenance requires manufacturing machines to be taken out of service, irrevocably resulting in reduced manufacturing capacity and revenue losses.
- **Incidental Damages**: Regular maintenance requires invasive actions - such as disconnecting electric cables, loosening and tightening bolts, and removing mechanical components - which may cause incidental damaging of secondary components.

However, recent technological advances such as Machine Learning have provided manufacturing companies with a powerful tool, allowing them to implement innovative maintenance strategies such as *predictive maintenance* [1].

In *predictive maintenance*, advanced statistics and predictive models are used to estimate the condition – and possibly predict the future failure – of mechanical components. This poses advantages when compared to classic *preventive maintenance* schemes:

- **Cost Reduction:** Mechanical components are only repaired or replaced when indicated by the algorithm. This reduces both the labor cost (maintenance staff) and the cost required for spare parts (inventory cost and actual component cost).
- **Reduction in Secondary Damages:** By identifying potential component failures, manufacturing companies can reduce the financial losses incurred by secondary damages when component failure occurs. This because failed components have the tendency to damage other - usually nearby – components, resulting in an aggregated cost in the case of component failure.
- **Reduced Outage Time**: Predictive maintenance results in a reduced number of maintenance interventions, thereby reducing the outage time and increasing the manufacturing capacity. The expected annual benefits β ($ per year), can be estimated by:

$$\beta = P(s) \cdot (\beta_f + \beta_s) \tag{1}$$

With β the total annual benefits, $P(s)$ the probability of successful of early component failure detection, β_f the benefit due to the decrease in outage time, and β_s the benefit due to forced outage time becoming scheduled outage time.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

In what follows, I will discuss the concept of Survival analysis – which is a commonly used machine learning tool that allows the prediction of mechanical component degradation and component failure, and therefore the implementation of data-driven maintenance schemes.

Survival Analysis for component failure detection

Survival Analysis is the branch within statistical modeling and machine learning which – unlike regular machine learning models that focus on regression and classification problems – aims at predicting the expected duration until the occurrence of a certain event of interest.

Historically, survival analysis has been used extensively within the field of medicine to obtain insights into which physiological factors influence the lifespan of certain populations. Due to improved digital storage capacity and increased efforts in keeping detailed logs about executed maintenance and component wear – survival analysis has become increasingly more popular for determining the time until mechanical component failure within high maintenance industries.

In this way survival analysis – commonly referred to as reliability analysis or reliability theory when dealing with mechanical components – provides valuable insights with respect to:

The component's survival function: the probability of a mechanical component failure after time t (or equivalently, the probability of component survival until time t), which can be denoted by the following function:

$$S(t)_{\text{component}} = p\,(T > t) \qquad (2)$$

Mathematically, this function represents a curve starting at $p = 1$ (i.e., 100% chance of survival at time $t = 0$) which approaches zero probability at $t = \infty$.

Variables of interest and Covariate relations: In the context of survival analysis, it is possible to detect the variables of interest – i.e. the variables which have a significant influence on the survival function and therefore the time until failure of the mechanical component.

In addition, survival analysis allows the observation of interactions between different covariates, possibly resulting in valuable insights regarding 'perfect storm'-scenarios in which the aggregation of a series of harmless variable conditions may lead to critical component failure.

In what follows, I present a step-by-step framework which allows the creation of a data-driven model to aid manufacturing companies in implementing a predictive maintenance strategy for mechanical systems [2][3]:

Step 1: Data collection

Collect data from the degradation process of the component of interest. This data includes component wear after certain usage time, average component failure time, and ambient component characteristics during operation.

Step 2: Determine degradation type

Component degradation may follow a linear, exponential, power, or logarithmic degradation process with respect to component usage. This degradation type needs to be determined and may depend on ambient conditions and material properties.

Step 3: Machine Learning regression model

Build a regression model (using machine learning methodologies), by taking into consideration the obtained data and specified component degradation characteristics.

Simplified degradation processes (e.g. linear processes) can be approximated by simple machine learning regressors like random forest regression or linear regression.

More sophisticated component degradation processes (as is usually the case) can be approximated by Neural Network models, Support Vector Regressors or Simple Deep Learning models. Select the optimal model based on prediction performance (using standard train – validate – test frameworks in combination with exhaustive parameter tuning techniques).

Step 4: Use obtained models for inference

The obtained model (one model per component or system) can be used for inference about the degradation process of similar components – i.e., identical components operating in similar ambient characteristics – and detect future component failure and wear.

Step 5: Use Inference results for Predictive Maintenance Decision Making

Inferred stages of component degradation (obtained in step 4) are used as an input to the predictive maintenance program of manufacturing companies. Taking into consideration required safety regulations, industry-specific safety factors, and machine availability, optimized maintenance schedules for specific components (as well as entire systems) can be determined.

Step 6: Iterate, Update Data, and Improve Inference

As time passes, more degradation data and component failure data become available. By establishing a constant data-pipeline from the data collection (step 1) to the degradation type determination (step 2) and machine learning model training (step 3), the results from the model inference (step 4) can be improved, allowing manufacturing companies to optimize their predictive maintenance strategy (step 5) in a continuous way.

III. HIGH CONFORMITY QUALITY CONTROL

An important aspect for guaranteeing a manufacturing company's trustworthiness is the production of durable, defect-free components which have been produced within very strict

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 11, November-2020

safety tolerances. To this end, manufacturing companies have implemented accurate quality control programs in order to deliver defect-free products to be used in high conformance environments.

Driven by an exponential increase in computing power and far-evolved data-storage capacities, quality control systems have made a lot of progress during recent years, with highly-specialized manufacturing environments generating only a few defective components per one million units [4].

In what follows, I will discuss the use-case in which imagery data of the manufactured component can be used in combination with advanced computer vision techniques for the purpose of component defect detection - thereby aiding high conformance manufacturing environments to automate quality control, reduce costs and improve defection rates.

Component Defect Detection using Computer Vision

Most component defects are caused by irregularities in manufacturing systems such as overheating, excessive machine loads, machine component wear, and faulty calibrations. Component defects resulting from such abnormal operating conditions can – usually – be detected by machine learning classifiers which have been trained on aggregated quantitative data obtained during the production process.

Some component defects – however – may not be captured in the collected data, resulting in a portion of the damaged components being left unidentified. In such cases, high conformance production environments rely on a secondary security step which involves visual inspection to guarantee a production output with minimal component defects.

However, adding a layer of visual defect inspection has some inherent shortcomings when opting to work with manual inspectors:

- Manual inspection is inconvenient and cumbersome, resulting in premature inspector fatigue and possible inspection flaws.
- Limited visual capacity only allows the detection of large production flaws, in comparison to the strict tolerances that are required within a high conformance production environment.
- Detected flaws may be misinterpreted or misclassified due to the subjective nature of the inspector’s judgment.

However, recent developments within the field of computer vision – along with drastic progress in the domain of artificial neural networks and deep learning – has resulted in the emergence of a type of artificial neural network which is capable of effectively processing and analyzing imagery data: Convolutional Neural Networks (CNNs) [5].

Like classical neural networks, convolutions neural networks consist of several consequent layers which are connected to each other. In convolutional neural networks – however – these layers consist of a series of different filters which are applied to the imagery input data. In essence, this sequence of filters allows the network to learn and detect increasingly more complex features within the input images. In addition, convolutional neural networks can be trained to search for specific features – which are position and rotation independent – within an image, allowing the detection and classification of certain component defects.

In general, convolutional neural networks require several layers of filters, with the most common layers – usually referred to as ‘the building blocks’ – being the convolutional layer, the max-pooling layer and the fully connected layer [6]

- **Convolutional Layers**

  Convolutional layers consist of a set of cube-shaped filters which are convolved with either the input image, or the output data from the previous layer. These convolutional filters are slid over the input data and are used to compute the dot product between the pixel values within the area of interest and the filter weights – resulting in what is commonly referred to a ‘convolved feature map’[7].

- **Pooling Layers**

  Pooling layers are periodically inserted in between the different layers of convolutional networks and are responsible for reducing the size of the convolved features obtained in previous layers. Their main purpose is to reduce the computational power that is required for processing the imagery input data [7].

- **Fully Connected Layers**

  Fully connected layers are added at the back of the convolutional neural network and take as an input the flattened vector-representation of the previous layers – being the convolutional layers and max-pooling layers. In essence, the fully-connected layers represent a regular neural network classifier which – instead of being trained on quantitative data as is the case with regular neural networks – is trained on the detected features within the component images [7].

- **Network Output**

  The output from the fully connected layer is a one-dimensional vector which contains the probabilities of the input image belonging to a certain class. Usually, the network’s predicted value - i.e., the network’s output - is determined by taking the class – i.e. the type of component defect - that is associated with the highest probability within this one-dimensional probability vector [7].

In what follows, I propose a framework to implement, select, and train a convolutional (deep) neural network to detect production flaws or component defects within high conformance production environments.

- **Step 1: Data collection**

  Collect imagery data - i.e., high-resolution images - from the components of interest. This dataset should include components without defects ($y = 0$), as well as components which were defected during the manufacturing process (i.e., wrong component tolerances, cracks or damages, and surface irregularities) during the production process ($y = 1, 2, .., k$ with $k = \#$ possible defects).

- **Step 2: Determine Network Architecture and Model Training**

  As discussed earlier, an arbitrary number of different layers can be used to construct a convolutional neural network. The layer types and quantity depend on the dimensions of the input image and the network's purpose. However, instead of training every network from zero, one could implement a pre-trained convolutional neural network and re-train this network on the available dataset. This results in a reduced network convergence time and an overall better predictive accuracy.

- **Step 3: Model Implementation (test phase)**

  By installing a high-resolution camera at different stages - or at the end - of the manufacturing process, imagery data of (partly) finished components can be fed into the trained convolutional neural network. During the test phase, this model can be added on top of the manual inspections to test its functionality and acquire more training data.

- **Step 4: Model Implementation and Iteration**

  When confirmed, fully implement the selected model for the automation of visual QC operations.

  Again – as discussed in the previous section - establish a data-pipeline in order to allow continuous updating and training of the selected predictive model. In the case of a defect detection, qualified maintenance personnel are informed and called for closer manual inspection.

REFERENCES

[1] Ahuja, I. P. S., and Khamba, J. S., "Total Productive Maintenance: Literature Review and Directions," International Journal of Quality and Reliability Management, Vol. 25, No

[2] Lin, J., Pulido, J., & Asplund, M. (2015). Reliability analysis for preventive maintenance based on classical and Bayesian semi-parametric degradation approaches using locomotive wheel-sets as a case study. Reliability Engineering & System Safety, 134, 143-156.

[3] Korvesis, P. (2017). Machine Learning for Predictive Maintenance in Aviation (Doctoral dissertation).

[4] Escobar, C. A., & Morales-Menendez, R. (2018). Machine learning techniques for quality control in high conformance manufacturing environment. Advances in Mechanical Engineering, 10(2), 1687814018755519.

[5] Arsenov, A., Ruban, I., Smelyakov, K., & Chupryna, A. (2018, November). Evolution of Convolutional Neural Network Architecture in Image Classification Problems. In Selected Papers of the XVIII International Scientific and Practical Conference on IT and Security (ITS 2018).–CEUR Workshop Processing (pp. 35-45).

[6] Namatēvs, Ivars. (2017). Deep Convolutional Neural Networks: Structure, Feature Extraction and Training. Information Technology and Management Science. 20. 10.1515/itms-2017-0007.

[7] Wu, J. (2017). Introduction to convolutional neural networks. National Key Lab for Novel Software Technology. Nanjing University. China, 5, 23.

DISCLAIMER

This disclaimer informs readers that the views, thoughts, and opinions expressed in the paper belong solely to the author, and not necessarily to the author's employer, organization, committee or other group or individual.