



Predictive Maintenance

Building Robust AI Models with Sparse Training Data

Are your AI models struggling to meet your predictive maintenance goals due to a lack of failure data for model training? Are your process engineers not trusting AI models' predictions because of a lack of explainability?

If your answer is yes to either, DeepIQ has the technology that can be a game changer for you. With DeepIQ's patent-pending Hybrid Knowledge-AI approach, you can build maintenance models that are:

- High quality

- Interpretable
- Predictive

Predictive maintenance is possible even for assets without significant failure data.

This whitepaper will explain this modeling approach using a bearings predictive health use case. We will conclude with some lessons for maximizing the value of your predictive maintenance program based on our implementations at some of the world's largest enterprises.

Background

Traditionally, industrial companies have relied on condition-based maintenance-monitoring equipment performance with visual inspections and scheduled tests to determine the most cost-efficient time to perform maintenance. Since condition-based maintenance focuses on lagging indicators of failure, an asset may have already incurred severe damage by the time of maintenance.

Enterprises that move to a predictive maintenance strategy focus on leading indicators of failure. This approach hopes to eliminate or avoid expensive downtime and repair costs.

Recently there has been significant improvement in sensor technologies across multiple dimensions, including cost, data richness, and ease of implementation, making predictive maintenance a viable alternative.

However, one significant challenge remains. Building high-quality predictive health models using AI requires a significant amount of data per failure mode. For most equipment, high-quality training data sets are not available. As a result, predictive health models might have considerable performance issues regarding missed detections and false alarms. Combined with the fact that AI models are black box approaches, a lack of the ability to explain the generated alerts has negatively impacted the adoption of this technology.

DeeplQ Introduction

DeeplQ's DataStudio software is a self-service {Data + AI} Ops application built for the industrial world. DataStudio simplifies industrial analytics by automating the following three tasks:

- Ingesting operational and geospatial data at scale into your cloud platform.
- Implementing sophisticated time series and geospatial data engineering workflows; and
- Building state of the art ML models using these datasets.

DeeplQ offers a patent-pending capability that revolutionizes the application of AI to data-starved problems. This capability relies on fusing two essential modeling techniques:



- Knowledge Engineering
- AI

To explain this in more detail, we will begin by introducing the concept of knowledge models.

Knowledge Models with Fuzzy Inference Nets

In this section, we will briefly introduce the concept of knowledge modeling with fuzzy inference nets. Readers interested in more details about DeepIQ's DataStudio support for fuzzy modeling can contact info@deepiq.com for more detail.

Fuzzy inference nets are a simple yet powerful way of capturing SME knowledge into a computational model. Fuzzy models are based on the Fuzzy set theory, where elements have degrees of membership, compared to traditional sets, where components have binary membership (they either belong to a set or not). For example, consider the problem assigning the equipment temperature as hot or not hot. Using traditional set theory, we can classify each element set into two distinct categories-hot or not hot, as shown in the crisp set in Figure 1.

However, fuzzy sets can classify each element into several linguistic categories such as somewhat hot, extremely hot, and not hot.



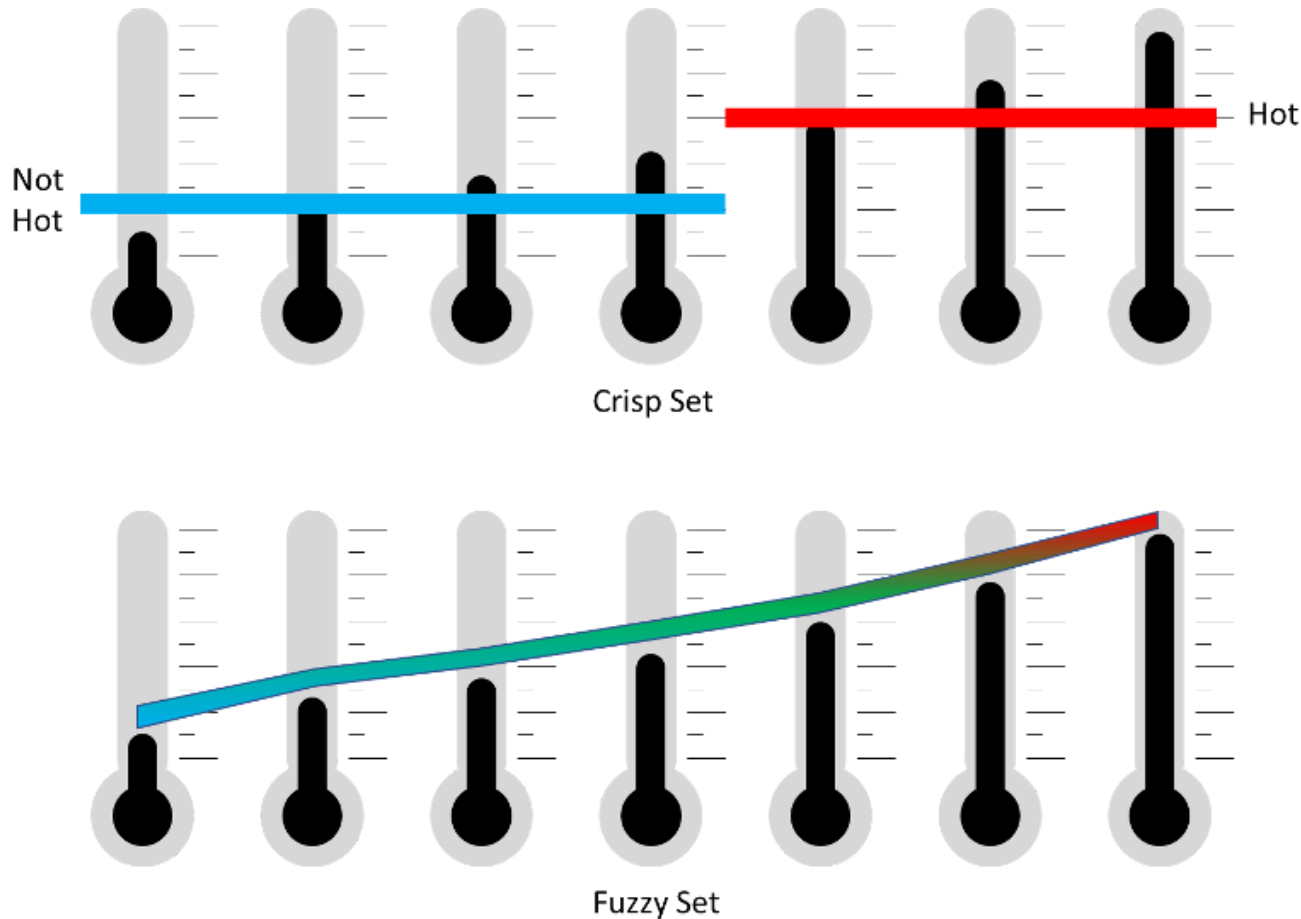


Figure 1: Crisp vs Fuzzy Datasets

Based on these fuzzy set concepts, subject matter experts can build a knowledge model using the following step process:

- First, features are converted to fuzzy sets using a technique known as fuzzification
- Then, a sequence of fuzzy rules is used to map input to output
- Finally, the output is defuzzied to generate the prediction

The result is a hierarchical decision-making process that converts raw features into a decision. This uses a reasoning system that mimics the expert's decision-making process.



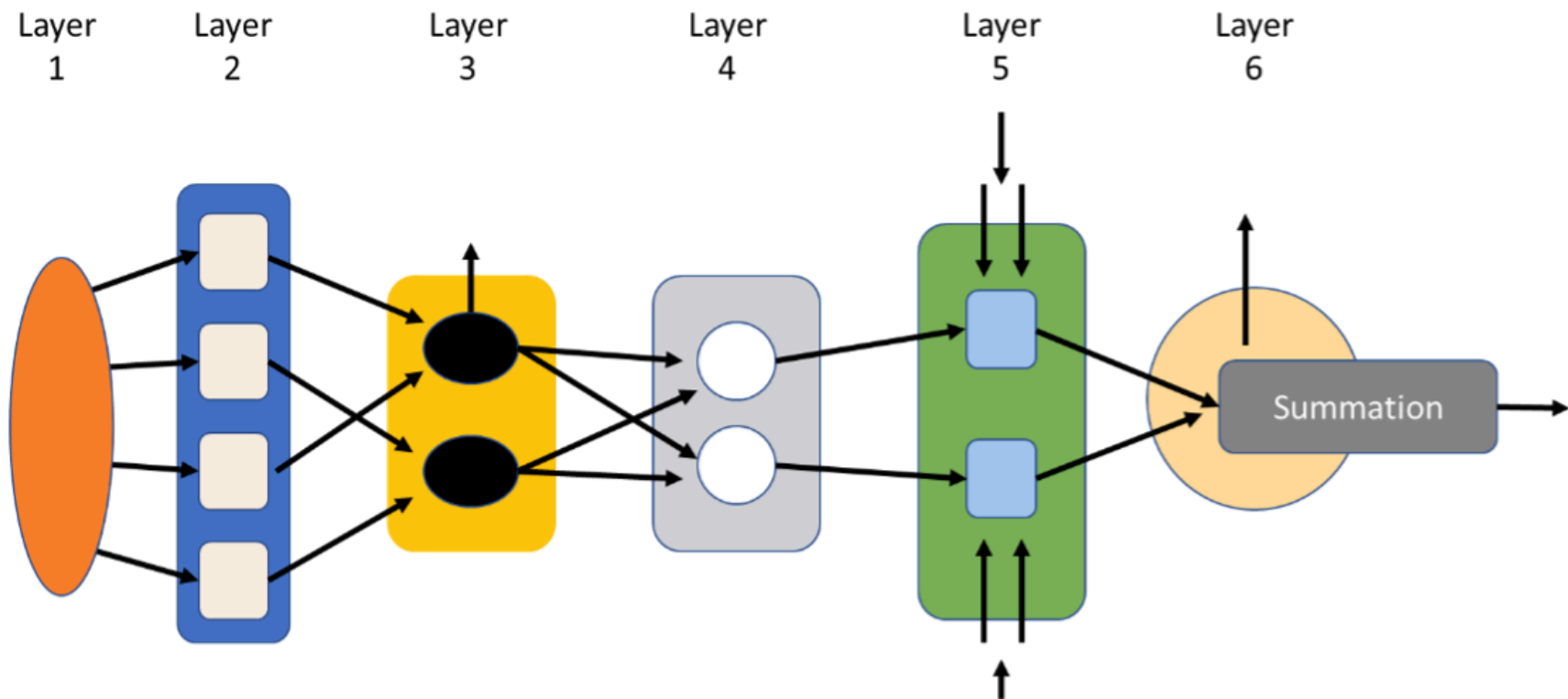


Figure 2: Fuzzy Modelling

With DataStudio, building inference nets is a simple click and drag operation that does not require specialized training.

Knowledge Models for Predictive Health

To build knowledge models for predictive health, maintenance experts identify features critical to predicting failures. Next, they fuzzify these inputs into a number between 0 and 1. Then, the experts use fuzzy aggregation rules that capture feature interactions to model how they impact failures. The aggregations generate a fuzzy set for each failure mode that indicates the belief that the equipment has that failure condition. For example, a high current is abnormal during the steady state processes but might be considered normal during rush conditions.

As the last step, experts manually tweak the parameters to optimize the historic data performance resulting in a model that mimics the process of maintenance experts to predict equipment health. However, there is a nuance to this process that limits this approach's



reach. Assume your knowledge model has fifty parameters. Even if each parameter is discretized to have a maximum of ten different values, searching a 10^{50} search space for the best possible model is impossible for a knowledge engineer. This is where AI can help.

AI Based Optimization

With DeepIQ's AI algorithm, you can avoid the laborious task of manually tweaking the knowledge network to optimize the knowledge model parameters. With this approach, you are not learning the model from scratch, but effectively capturing the expert's knowledge, and then using AI as an efficiency multiplier.

Sample Problem

Let us look at the example of one of the key rotating parts in mechanical equipment – bearings. Bearings are often considered the most vulnerable parts of rotating equipment and any failure can lead to catastrophic damage to the entire asset. Vibration data analysis has been the most used technique for detecting defects in bearings. Spectral or frequency analysis of vibration signals is the classical method to diagnose these issues.

However, the required fault information is often in the noise, making it difficult to identify the defect. On the other hand, time domain analysis provides simpler and faster calculations. This is insensitive to initial defects making it hard to predict the problem earlier.

Here, we present a strategy that uses both spectral and time domain information and make use of fuzzy logic inference nets to categorize the various bearing faults.

As a first step, we use DataStudio's built-in components to ingest the vibration signals data. To demonstrate this capability, we use the IMS bearing dataset. We created two workflows to generate the feature dataset for this problem as shown in Figure 3 and Figure 4. The features are the various time energy indicators such as standard deviation (SD), entropy (EN), variance (VAR) and kurtosis (KU). Please refer to the above reference for more details on the data set.

As mentioned earlier, the failure examples are very sparse. There are eight examples of a healthy signal, one example each of an inner race, an outer race, and a rolling element failure. The first workflow ingests time domain vibration signal recorded over one second and computes various statistical features. We used a procedure outlined in open literature , where we split the vibration signal into slices so that appropriate bearing defect frequencies are properly captured in the feature set.

Further, singular value decomposition (SVD) is used for noise reduction and to extract the features of highest importance. The second workflow loops over all the vibration signal samples using DataStudio's loop component. Note that these workflows can handle exceptionally large datasets (terabytes in size) due to the DataStudio's distributed computing capability.



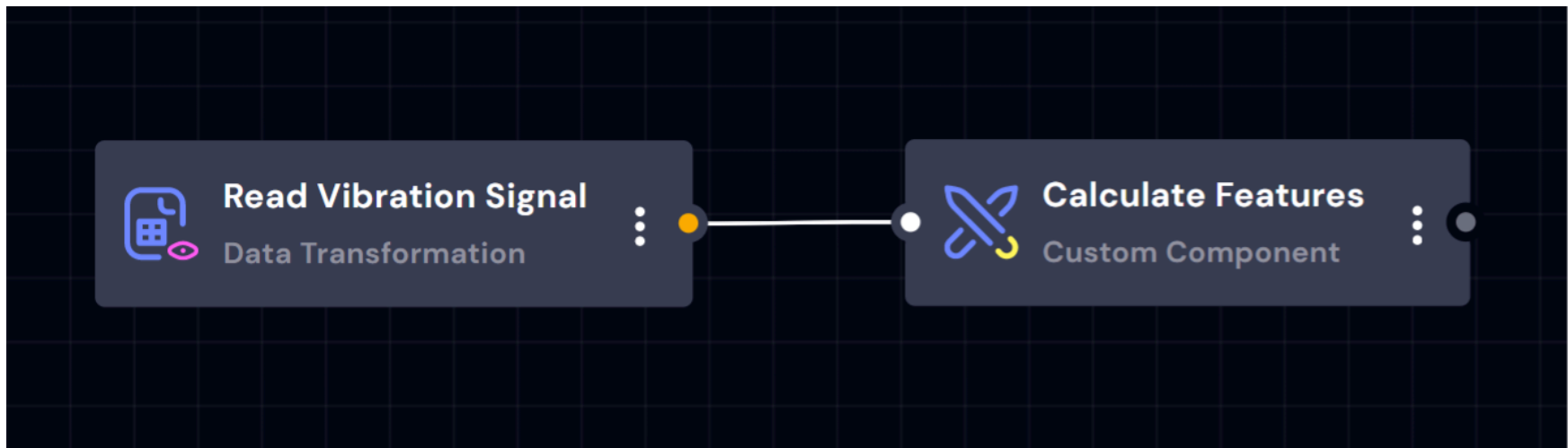


Figure 3: Workflow to ingest multiple vibration sensor samples from a folder

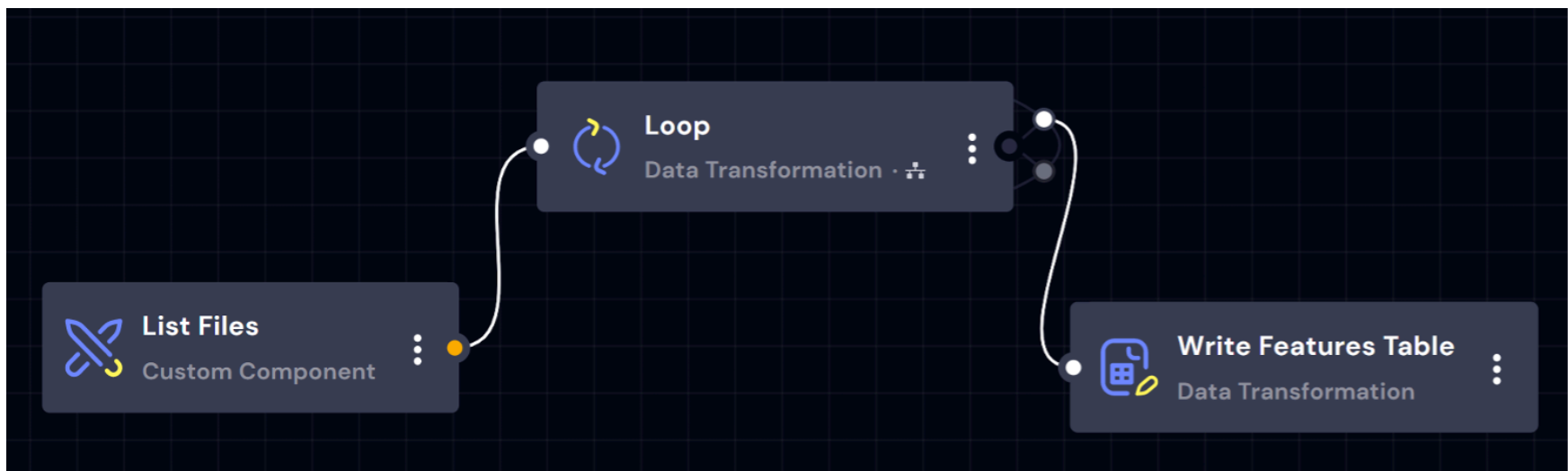


Figure 4: Workflow to compute statistics for each vibration sample

These computed features form the input to a Fuzzy Inference System (FIS). As explained earlier in the paper, in FIS the input features are first fuzzified to values between 0 and 1. To generate these values, DeepIQ's DataStudio provides multiple membership options



(large, small, trapezoidal, triangular) as simple drag and drop components. Next, multiple fuzzy aggregations rules are applied on the fuzzified data to capture the different failures in the dataset and classify the dataset accordingly. An example of these rules is:

If fuzzified entropy lies between 0.4 and 0.6, and fuzzified standard deviation is more than 0.8, then classify the data as an inner race failure. The outcome will be a fuzzy set for each of these bearing failures. These rules are primarily set by the experts by tweaking the parameters based on their knowledge. Several fuzzy aggregations are readily available in DataStudio app for experts to use. The complete inference system is created as a single workflow as shown in Figure 5.

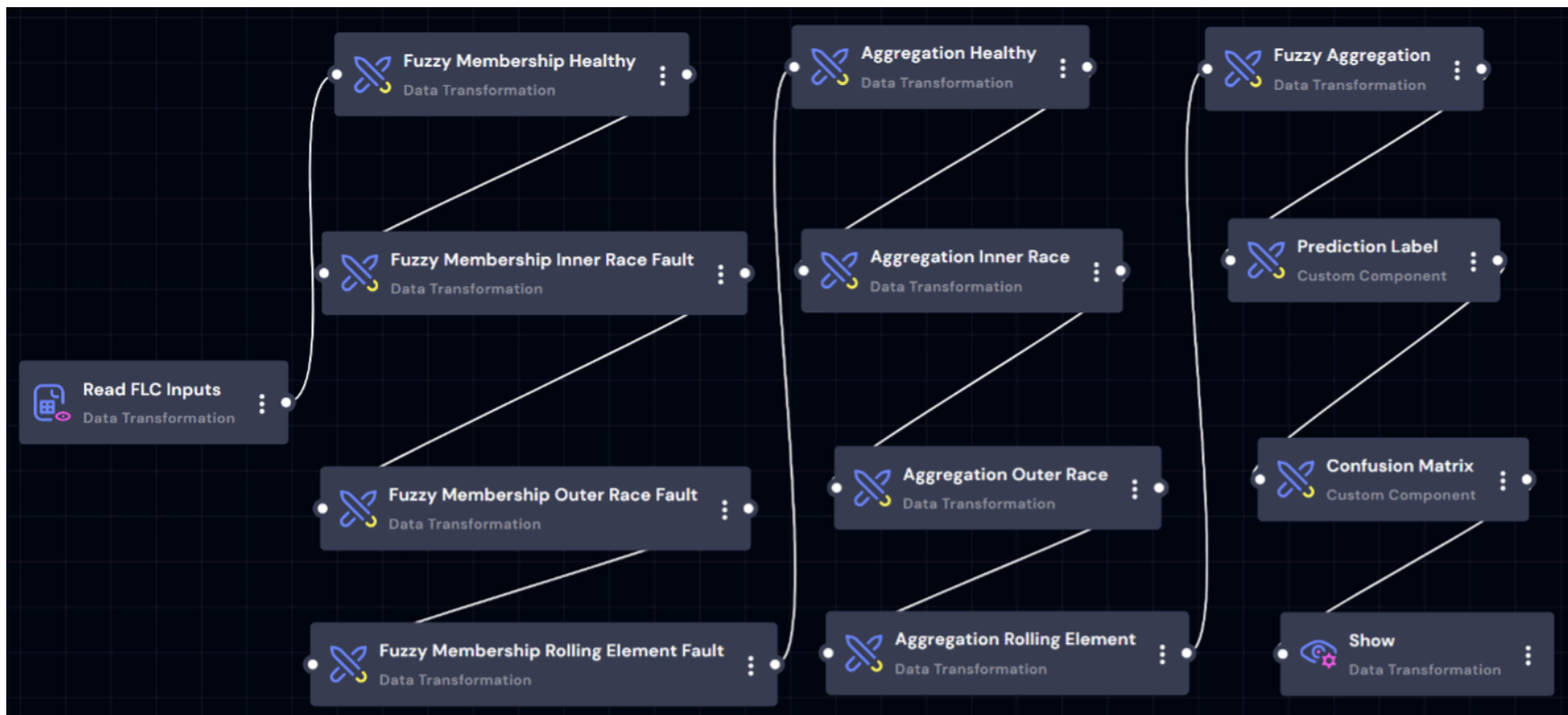


Figure 5: Fuzzy inference system

As mentioned earlier, the network has many tuning parameters in the network and manually tweaking these parameters to get the best performance is not possible. DataStudio offers AI capabilities to optimize the parameters by maximizing the fit of the knowledge model to historic data. After employing the optimizer on the Fuzzy Inference Model, we observed significant improvements in the performance of the classifier as shown in Figure 6.

Baseline

AI Optimized

DeepIQ is on a mission to transform industrial processes by digitizing industrial expertise. Our vision is to drive end-to-end automation, enabling systems such as self-running power plants or drilling rigs using generative AI as the higher order reasoning layer operating over existing industrial automation technology stack.

[Privacy Policy](#) [Terms of Services](#)



FEATURES

Extract
Engineer
Explore

INDUSTRIES

Upstream
Midstream
Downstream & Chemical
Mining

SOLUTIONS

IT-OT Contextualization
Well Construction
Optimization
P & ID Digitization
Predictive Maintenance
Production Optimization
Route Optimization
Sustainability

PARTNERS

AWS
Azure
Cloudera
Google
Databricks
OSIsoft(AVEVA)
Snowflake

COMPANY

[About Us](#)

[Resources](#)

[News](#)

[DataStudio Deployment
Guide](#)

[Career](#)

[Contact Us](#)

