Prognostic Health Management(PHM) primariy aims to predict the remaining useful life(RUL) of a degrading component/system utilizing monitoring data(prognostics). The availability of monitoring data from run-to-failure experiments for certain components/systems allows for data-driven prognostic solutions, aiding toward prevalence of the data-driven predictive maintenence(PDM) paradigm. In this

context, we here propose a metric for assessing data-driven prognostic algorithms based on their impact

on downstream PdM decisions. The metric is defined in association with a given decision setting and a corresponding PdM policy, which is employed for determining the maintenance actions. We consider two typical PdM decision settings: i) component replacement planning and ii) component ordering- replacement planning, for which we investigate PdM policies that are commonly utilized in the literature. All policies are evaluated via the data-driven estimation of the long-run expected maintenance cost per unit time; for this estimation we rely on available run-to-failure datasets. The policy evaluation forms the basis for estimation of the proposed metric. we employ differenet version of transformer NN architecture as data-driven prognostic algorithms on an application case study related to turbofan engine degradation, and investigate the joint effect of prognostic algorithm and PdM policy on the metric. These two are the following PDM policies, PDM policy1: simple heuristic Pdm policy for preventive replacement and PDM policy3: modified Pdm policy for preventive replacement on the basis of the full RUL distribution would be used along with prognostic alogorithms(Transformer) such as Informer, Spacetimeformer and Autoformer. The reason behind choosing choosing all three transformer achitecture is becuase they are open sourced. Informer introduces the ProbSparse attention, the ProbSparse attention locates the most dominant queries and only allows keys to attend to these (based on the Kullback-Leilbler (KL) divergence against a uniform distribution). In essence, this method of introducing sparsity should not overlook potentially important time steps, which might be the case for fixed sparsity patterns. The Autoformer replaces the original scaled dot-product attention mechanism with the Auto-Correlation module (!). This operation uses keys and queries to decide on the most important time-delay similarities through autocorrelation and time-delay aggregation. This again, is expected to perform better for many time series applications, as it is specifically designed for time series similarity. Spacetimeformer proposes a new way to represent inputs. Temporal attention models like Informer represent the value of multiple variables per time step in a single input token, which fails to consider spatial relationships between features. Spacetimeformer combines both temporal and spatial attention methods, creating an input token to represent the value of a single feature at a given time. This helps the model understand more about the relationship between space, time, and value information.