In Situ Prediction Driven Feature Analysis in Jet Engine Simulations

Soumya Dutta Han-Wei Shen Jen-Ping Chen

The Ohio State University *

ABSTRACT

Efficient feature exploration in large-scale data sets using traditional post-hoc analysis approaches is becoming prohibitive due to the bottleneck stemming from I/O and output data sizes. This problem becomes more challenging when an ensemble of simulations are required to run for studying the influence of input parameters on the model output. As a result, scientists are inclining more towards analyzing the data *in situ* while it resides in the memory. *In situ* analysis aims at minimizing expensive data movement while maximizing the resource utilization for extraction of important information from the data. In this work, we study the evolution of rotating stall in jet engines using data generated from a large-scale flow simulation under various input conditions. Since the features of interest lack a precise descriptor, we adopt a fuzzy rule-based machine learning algorithm for efficient and robust extraction of such features. For scalable exploration, we advocate for an off-line learning and *in situ* prediction driven strategy that facilitates in-depth study of the stall. Task-specific information estimated *in situ* is visualized interactively during the post-hoc analysis revealing important details about the inception and evolution of stall. We verify and validate our method through comprehensive expert evaluation demonstrating the efficacy of our approach.

Keywords: *In situ* analysis, machine learning in visualization, fuzzy rule-based system, multivariate analysis, rotating stall, big data analytics.

Index Terms: I.5 [PATTERN RECOGNITION]: Feature evaluation and selection—Fuzzy set models; I.3 [COMPUTER GRAPHICS]: Display algorithms—Applications; J.2 [PHYSICAL SCIENCES AND ENGINEERING]: Aerospace.

1 INTRODUCTION

Recent advancements in computing power have enabled scientists to model various physical phenomena with great precision. Such simulations are typically run on supercomputers and result in large-scale data sets. Traditional post-hoc analysis of such data is prohibitive due to the bottlenecks arising from I/O and output data set sizes [13]. Furthermore, to obtain detailed knowledge about the behavior of the models under different conditions, experts study the model output under various parameter configurations requiring them to run an ensemble of simulations. Note that, each of these simulations would produce a large-scale data set and storing all the raw data for post-hoc exploration is not a viable option.

While studying complex events such as the evolution of stall in jet engines, formation of viscous fingers in a fluid etc., scientists generally rely upon domain-specific information derived from the simulation. Efficient extraction and exploration of such information require (a) Understanding about the target features; (b) Effective methods for robust and scalable detection of such features; and (c) Summarization and interactive exploration capability through appropriate visual encodings. Several previous works in this context have acknowledged that defining features with precise descriptors is becoming increasingly difficult due to the intricate nature of the simulated phenomenon [7, 14]. So, experts carefully study and employ visual analysis techniques to study their region of interest. However, as pointed out by Ma [27], manual exploration of large-scale data sets is not desirable.

In this work, we embrace the pathway of *in situ* analysis to tame the large-scale data challenges. Working with a flow simulation, TURBO [10], as a specific use case, we study the evolution of *rotating stall* in jet engines under different parameter settings. Rotating stall is characterized by local airflow disturbances, which grow rapidly and can become destructive to the engine. So, for a stable engine operation, experts employ a safe throttle setting which is modeled by the parameter *corrected mass flow rate (CMF)* in TURBO. By changing it, the performance of the engine can be greatly improved by minimizing the aerodynamic loss. Currently, the best possible operating range of CMF is unknown. Hence, the experts want to study the role of CMF in stall inception. Note that, a traditional post-hoc analysis with raw data [9] will not scale, because just a single run of TURBO generates data in the order of tens of TBs. Besides this, the feature of interest, i.e., the stall cell lacks precise descriptor making the problem even more challenging. Therefore, unlike previous works [9, 10] that used a single variable to identify such features, the expert wants to expand the identification of stall in the multivariate domain.

To address the aforementioned issues, we introduce an analysis technique that enables *in situ* characterization of imprecisely defined features using a knowledge-base constructed in an off-line learning phase. The expert initially investigates a known simulation data to highlight stalled and stable regions. By employing a fuzzy pattern learning algorithm, we capture the multivariate relationship of several variables from the expert-labeled sample points. The relationship is modeled using a fuzzy rule-based system allowing us to translate the knowledge of the expert into the knowledge-base of an intelligent system. After learning, prediction of stall under different parameter settings takes place *in situ* using the inference scheme of the fuzzy system. Besides this, since the stalled regions act as blockages to the normal airflow, it is hypothesized that the stalled passages will demonstrate lower mass flow rate compared to the healthy passages. However, as the traditional global mass flow rate is a single aggregated value, it can only detect whether the stall has happened, and the indication of stall comes quite late. So, along with the fuzzy system guided stall detection, we also conduct a novel passage-wise local mass flow rate based stall analysis. We show that, with our proposed strategy of off-line learning and *in situ* feature prediction, the expert can study the influence of parameters like CMF on the output in a timely manner with minimal effort. Through *in situ* analysis, we bypass the expensive I/O and output only derived task-specific data which allows interactive post-hoc analysis for gaining insights about the stall phenomenon. So, our contributions are threefold:

1. We introduce an off-line learning and *in situ* prediction based analysis strategy which provides a machine learning based solution for exploring features in very large-scale simulations.
2. We successfully demonstrate an approach of *in situ* detection of imprecisely defined features and show its applicability us-

*e-mail: {dutta.33, shen.94, chen.1210}@osu.edu

2165-8773/18/$31.00 ©2018 IEEE
DOI 10.1109/PacificVis.2018.00017

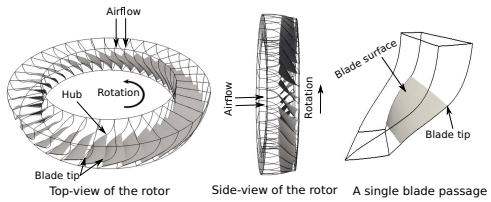


Figure 1: A schematic diagram of the rotor structure of the compressor stage.

ing a real-world flow simulation TURBO for studying the evolution of rotating stall in transonic jet engines under varying input parameter configurations.

3. We develop visual-analytics systems, which allow experts to interactively specify their region of interest directly from the data, and furthermore, facilitate effective investigation of the feature-specific data, derived *in situ*, for simulation profiling.

2 Related Works

Significant contributions have been done by the visualization researchers for enabling analysis of complex flow features. Chen et al. [10] studied the pre-stall behavior of turbine engines in their work. Shaffi et al. [32] proposed techniques which can analyze flow features near vortex-turbine intersections in wind farms. Chen et al. used statistical anomalies for stall detection in [9]. A comprehensive report of visualization works in analyzing features in different material science and computational fluid dynamics (CFD) applications can be found in [21].

***In situ* analysis in visualization.** The necessity of *in situ* analysis has become prominent for its advantages over post-hoc techniques [37]. Fabian et al. added *in situ* capability in ParaView with CATALYST library [16]. Run-time visualization in VisIt using LibSim was introduced by Whitlock et al. [36]. Another *in situ* framework ADIOS was developed by Lofstead et al. [26]. Enhancement of simulation time exploration was proposed by Vishwanath et al. by using GLEAN [34]. Woodring et al. demonstrated *in situ* eddy analysis in high-resolution ocean simulation models [40]. For efficient *in situ* data processing, Woodring et al. introduced a zero copy data structure [39]. *In situ* volume visualization was highlighted in the work of Yu et al. [41]. Recently, *in situ* data triage schemes have been followed by researchers to extract only the task specific information for enabling scalable post-hoc analysis [11,24]. Ahrens et al. utilized an *in situ* image-based approach [1] for exploration during post-hoc analysis. Sampling-based analysis of cosmology data was done by Woodring et al. in [38]. Other research works have also showed distribution-based data summarization for flexible post-hoc analysis of large-scale data sets [13, 15]. A state-of-the-art report on *in situ* methods in visualization can be found in [3].

Predictive feature exploration in visualization. Prediction-based techniques for feature exploration and visualization under uncertainty have been shown quite effective in many previous visualization works. A predictor-corrector based approach for detecting feature correspondence was proposed in [28]. Interactive predictive parameter space analysis was demonstrated in the works of Berger et al. [4] where the technique provided guidance to the users for locating interesting parameter regions. Prediction of uncertainty in volume visualization pipeline was obtained using a possibility-based approach [17]. Prediction of uncertain features using a feature-aware classification field was used in [14]. Similar to our approach, various rule-based techniques were used by many researchers for performing predictive feature analysis. A Rule guided feature tracking using inductive learning was demonstrated by Banerjee et al. [2]. A framework for hypothesis generation and verification using a fuzzy logic based learning approach was introduced by Fuchs et al. [18]. A fuzzy rule-based efficient approach for tracking molecular particles was shown by Jiang et al. [22]. A fuzzy set based possibilistic marching cubes algorithm was proposed by He et al. in [20]. Automatic error controlling during volume rendering was achieved using a fuzzy controller [25]. In this work, we use a fuzzy rule-based approach for characterizing and predicting features which do not have a precise descriptor. Our approach learns the target feature properties in an off-line mode using the expert's guidance and predicts in an *in situ* environment for facilitating scalable feature exploration of very large-scale data.

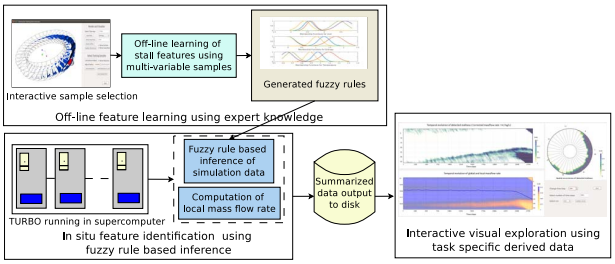


Figure 2: A schematic diagram of the proposed analysis method. Our off-line learning and *in situ* prediction based analysis strategy allows the experts to study the evolution of features in large-scale data sets in an effective and timely manner.

Machine learning in visualization. Use of machine learning techniques in visual analytics applications has emerged as one of the most promising paths forward for the researchers [27]. A majority consensus-based vortex analysis framework was proposed by Biswas et al. [7]. Zhang et al. [42] used AdaBoost algorithm for improving vortex extraction. Kaumpf et al. visualized the confidence in the clustering process while analyzing data in their work [23]. A detailed study on the effects of dimensionality reduction and clustering on high dimensional data was done by Wenskovitch et al. [35]. For sophisticated higher-dimensional classification of volume data sets, Tzeng et al. [33] utilized neural network based techniques. Comparison of different labeling schemes with visual analytics guided labeling for machine learning applications was demonstrated by Bernard et al. [5]. In our work, the expert selected the labeled samples for training by directly interacting with the data. After that, we followed an off-line learning and *in situ* feature prediction for scalable classification of large-scale data.

3 Background, Domain Problem, and Overview

3.1 Application Background

Understanding the evolution of rotating stall is critical for the aerodynamics scientists to prevent permanent engine failure. To gain insights about rotating stall, a numerical CFD simulation code TURBO has been developed at NASA which can model stall with high accuracy. In Figure 1 the rotor structure and different components of TURBO simulation are shown. Using data generated from TURBO, experts want to investigate various conditions which may initiate stall. This mandates a comprehensive ensemble study within a parameter space consisting of the various flow controlling and blade designing parameters. Among them, engine's throttle setting, modeled by the parameter CMF, is a prime candidate for investigation. By setting a proper CMF value the performance of a compressor can be boosted with increased fuel efficiency as well as improved reliability. Unfortunately, certain CMF values can also trigger instability. Hence, a detailed understanding of the impact of CMF in stall inception will allow scientists to push the performance limit while still maintaining the engine safety. The benefits of improved understanding of stall could yield compressor designs that can operate closer to their maximum efficiency.

Existing visual analytics guided stall exploration strategies [9, 10] utilize raw data and follow a post-hoc approach which is not applicable in our scenario as they do not scale due to: (a) I/O bottleneck; and (b) the large size of the output data. So, running TURBO multiple times with different CMF values will produce data sets

which ideally can only be analyzed cost-effectively in an *in situ* environment. A recent work along this line [13] proposed *in situ* data reduction using Gaussian mixtures and performed post-hoc exploration using the reduced data. The approach worked well while using a single variable for stall analysis. However, extending it to the multivariate domain for allowing correlation preserving exploration will face *in situ* computational challenges since high dimensional Gaussian mixture estimation is significantly more expensive compared to the univariate one. Also, the spatial anomaly-based analysis in the previous works [9, 13], if done *in situ*, will require a large amount of additional data communication slowing down the simulation.

3.2 Domain Specific Requirements

Given the aforementioned limitations of existing methods under the context of our current study, we have identified the following requirements from the expert which are needed to be addressed:

1. The expert first wants to know if the current value of CMF has led the simulation to a stalled condition. If it has resulted in a stall then what time step ranges have started showing signs of a stall and how the phenomenon has evolved over time?
2. Which part of the rotor is affected by the stall and how the stalled regions span in the spatial domain?
3. How can multiple variables together be used instead of a single variable to detect the stalled regions more reliably?
4. Can we analyze and visualize the derived information about the simulation and make decisions such as whether to run the simulation for a longer time? This will facilitate efficient use of the CPU cycles when the study needs to be performed under a constrained resource budget.

3.3 Overview of the Proposed Approach

Figure 2 shows a schematic view of the proposed analysis pathway. Acknowledging the fact that the features of interest are non-trivial to be defined by specific descriptors, we provide a visual-analytics tool through which the expert can explore the data and directly select their target region. By collecting the sample points marked by the expert, we construct a sample set where for each sample point we measure three variable values. We employ a fuzzy rule-based pattern learning algorithm to learn the multivariate pattern from the sample set. The fuzzy rule base is then employed *in situ* for predicting the stall when the simulation is run with unknown parameter settings. The fuzzy system assigns a classification score to each point which represents the *stallness* of that point. Besides this, we also compute the local mass flow rate of each blade passage. The task-specific information about stallness and local mass flow rate are stored into disks which are significantly smaller in size compared to the raw data. Finally, the *in situ* derived information is explored post-hoc using an interactive visualization that enables profiling the simulation and reveals important details about stall inception.

4 Interactive Generation of Training Data

In the absence of precise/hard feature descriptors, selection of a region of interest directly from the data was advocated by researchers [14]. The usefulness of the labeled samples, collected directly from the data by the expert, for efficient classification of vortices was demonstrated in previous studies [7,42]. We follow the similar strategy and allow the expert to locate their features directly in the data. As hard classification of features is not possible, the expert labels a confidence value reflecting the degree of stallness for the selected points by specifying a value between 0 and 1, where 1 means definitely stall, and 0 indicates the stable condition. This strategy ensures a tight coupling of the expert-knowledge into our system [42].

To achieve this, we provide a visual-analytics system which the expert uses day-to-day for exploring the regions of interest. Since

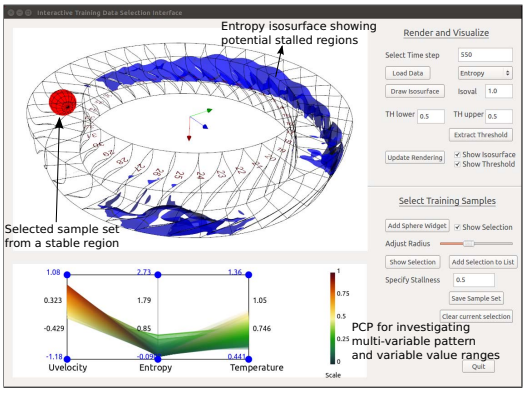


Figure 3: Interactive interface for selecting the region of interest from data.

stall cells are associated with flow separation characterized by reverse axial flow, the expert is interested in locations where Uvelocity (Uvel) is negative. However, near the blade tip, a small amount of reverse flow always exists which does not indicate stall. Only when the amount of reverse flow grows significantly and blocks normal airflow, it can lead to a stall. Hence, instead of relying on a single variable, the expert also wants to explore entropy and temperature variables for locating stalled regions. This is because, based on the experts' domain knowledge, the stalled regions generally show high entropy and temperature values, i.e., they have a direct correlation, and Uvel values appear to be negative and low. Hence, Uvel is expected to have a negative correlation with the other two variables. The goal is to identify a sample set from which the above multivariate relationship among the three variables can be estimated using a set of fuzzy rules, such that the rules can be applied to infer the degree of stallness for unknown data set. Using our interface, the expert analyzes potential regions in the data using isosurfaces and thresholding techniques on multiple important variables and uses domain knowledge to iteratively refine the regions which best represents the stall cells. Our tool provides an adjustable 3D sphere widget for highlighting such regions. In Figure 3, we depict the interface where the red points show a selection using the sphere widget. The interactive Parallel Coordinates Plot (PCP) allows the expert to study the multivariate pattern among the selected variable values where the line colors reflect their scalar values.

A detailed investigation of several time steps from a known data set reveals that the stalled regions as mentioned above indeed demonstrate high entropy and temperature values, whereas, an almost opposite relationship is found in the stable regions where the Uvel is positive and high, and entropy and temperature values are low. The left image in Figure 4 shows the case where the approximate stalled region is highlighted by an isosurface of a high entropy value of 0.9 and the selected sample points are marked in red coming from a stable region. The multivariate relationship of these points is revealed in the PCP below showing the correlation among the three variables. Observe that, the Uvel is high and positive and both entropy and temperature demonstrate low values. Hence, the stallness value of these sets of points is set to 0.1 by the expert indicating a low chance of a stall. In contrast, the right image in Figure 4, we see sets of points selected from a stalled region that is enclosed by the entropy isosurface. Here, the pattern is quite different as seen in the PCP below. The Uvel is negative indicating reverse flow, and entropy and temperature values are high. So, the degree of stallness of these sets of points are set to 0.9 reflecting high confidence that the selected region is stalled. Following this approach, the expert can investigate several time steps and mark regions that are either stalled or stable and also specify the degree of stallness based on their domain knowledge. All these points are collected for constructing the fuzzy rule base.

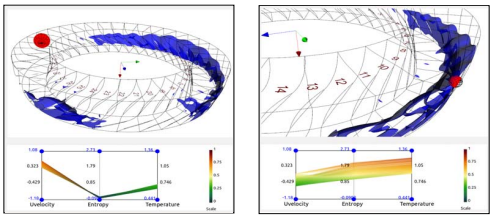


Figure 4: Interactive selection of samples for training where the stalled regions are roughly shown using a high entropy value isosurface. In the left image, the sample points highlighted (within the sphere in red) are selected from a stable region, whereas, in the right image the samples are picked from a stalled region. The differences in patterns among the three selected variable values shown in the PCP are notable.

5 Off-line Learning For Fuzzy Rule Generation

In this work, we deal with features that cannot be defined precisely. As a result, the expert usually employs visual analytics methods and finally roughly locates the features with a certain degree of confidence. However, manual identification of such features from thousands of time steps is undesired, especially when the data size is very large. So, robust and intelligent feature detection algorithms are essential for automatic classification. Also, since our target environment for feature detection is *in situ*, we require algorithms which offer fast classification capability with a low memory footprint. A fuzzy rule-based system (FRBS) in this case presents a suitable choice. For robust detection of imprecisely defined stall cells, we use an FRBS consisting of a knowledge-base and an inference engine. Use of an *in situ* friendly FRBS allows us to compactly transfer the domain knowledge of the expert into an intelligent system and automate feature classification.

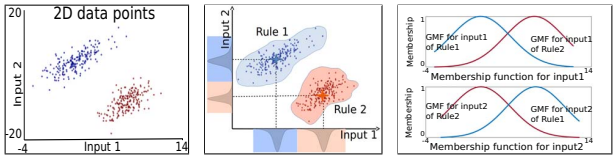
5.1 Fuzzy Clustering Guided Rule Generation

In order to extract the relationship between the input values and their corresponding output values, the expert specified sample points are first clustered using a fuzzy-c-means (FCM) [6] clustering algorithm. Note that, each point in our training set is a 4-tuple: {Uvel, entropy, temperature, stallness}, where the first three values are observations of variable values and are treated as input to the fuzzy system, and the fourth component is the output fuzzy classification score. By clustering the input and output values together, the generated clusters group input points which also have similar output. Efficacy of the FCM algorithm in extracting cluster structures from high dimensional data has been demonstrated previously in [29] and has been adopted in many cluster based rule generation techniques. The algorithm generates a membership matrix along with the cluster centers. The membership matrix contains the membership of each point to all the clusters. The sum of membership values across all the clusters for each point is 1. So, given $X = \{x_1, x_2, .., x_n\} \in \mathbb{R}^p$ as the input to the FCM, the algorithm produces a set of centroids $V = \{v_1, v_2, .., v_c\}$ and a membership matrix M by minimizing the objective function:

$$\psi_r(M,V) = \sum_{k=1}^{n}\sum_{i=1}^{c} m_{ik}^r \|x_k - v_i\|^2 \quad (1)$$

where r is a weighting exponent (typically = 2), n is the number of points in the training set, and c is the number of clusters. The dimension of the output membership matrix M is $c \times n$ and the element m_{ik} represents the membership of k^{th} point in i^{th} cluster.

As each cluster encapsulates a specific relationship among the grouped input-output sample points, each such cluster is formalized into a *IF-THEN* predicate-based fuzzy rule with a standard form: *IF (antecedent) THEN (consequent)*. The antecedent clause of each rule consists of several atomic sub-clauses and are connected using fuzzy T-norm conjunction operators that aim to estimate the degree of "closeness" of the given input component to the corresponding



(a) (b) (c)

Figure 5: Figure 5a: A synthetic bivariate training data generated from two 2D multivariate Gaussian distributions centered at (2,8) and (8,2) respectively; Figure 5c: Trained Gaussian membership functions (GMF) for the sample bivariate data; Figure 5b: Conceptual scheme of fuzzy clustering based rule identification.

sub-clauses of the rule. We use the fuzzy operator "AND" in this work as the fuzzy conjunction operator. To quantify this degree of "closeness", each sub-clause in the rule is modeled by a fuzzy membership function. Note that, different types of membership functions can be used here such as triangular, trapezoidal, Gaussian etc. In this work, we have used Gaussian membership function (GMF) because it is differentiable everywhere and has only two parameters which can be tuned efficiently [29]. Therefore, the degree of closeness in each sub-clause is estimated by evaluating the GMF associated with it. This process is also known as the "fuzzification", which changes a real scalar value into a fuzzy value. Formally a GMF is defined as:

$$gmf(x) = exp^{-\frac{(x-\bar{x})^2}{2\sigma^2}} \quad (2)$$

where $\bar{x}$ is the mean and σ is the standard deviation. The estimated centroid of each cluster then becomes the suitable choice for mean values of the corresponding GMFs and the standard deviation of each GMF is computed as:

$$\sigma_i^j = \sum_{u=1}^{n} \frac{-(x_i^u - v_i^j)^2}{2 \cdot log(m_i^u)} \quad (3)$$

where σ_i^j is the standard deviation of the i^{th} clause of j^{th} rule, m_i^u is membership of u^{th} sample point obtained from the membership matrix M, and n is the number of points in the training set.

To demonstrate this concept of rule generation, we created a synthetic 2D data set shown in Figure 5a which was obtained by randomly sampling points from two 2D multivariate Gaussian distributions centered at (2,8) and (8,2) respectively. It can be observed that the 2D point set has two clusters. The output value for points coming from the first Gaussian was set to 0 and for the points sampled from second Gaussian was set to 1 for experimentation. Hence, the training set, in this case, is a set of 3-tuples. Since each cluster is translated to a fuzzy rule, the rule for the first cluster can be written in the form: "IF (($input_1$ is close to $input_1$ of cluster $center_1$) AND ($input_2$ is close to $input_2$ of cluster $center_1$)) THEN the point has output value close to 0". Similarly, the second rule for the other cluster will be: "IF (($input_1$ is close to $input_1$ of cluster $center_2$) AND ($input_2$ is close to $input_2$ of cluster $center_2$)) THEN the point has output value close to 1". In Figure 5b, we show the schematic diagram using the 2D synthetic data where the mean values of the membership functions are estimated by the cluster centroids and each cluster is interpreted as a fuzzy rule. The estimated GMFs are depicted in Figure 5c. It can be observed that each rule with two sub-clauses has two membership functions (denoted by blue for rule 1 and orange for rule 2). With the estimated GMFs, we can construct the antecedent part of the rules. Next, we demonstrate how the estimation of the parameters for the output prediction function is achieved which is the consequent part of the rule.

5.2 Estimation of Parameters for the Output Function

Since the inference algorithm of the fuzzy system will be run *in situ*, we model the output variable using a linear function allowing com-

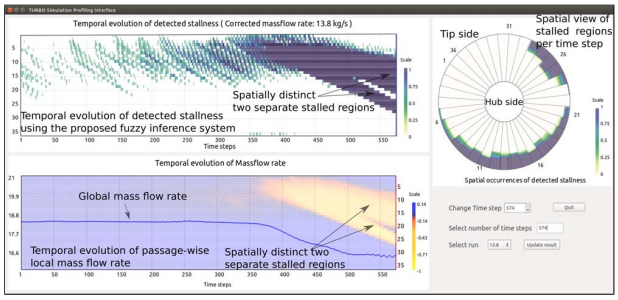


Figure 6: Visual analytics interface to study the stall features (CMF = 13.8 kg/s) estimated *in situ* through fuzzy rule-based system and local mass flow rate computation.

putationally efficient inferencing. We adopt the well-known *Takagi-Sugeno fuzzy rule-based system* (TS-FRBS) which has been shown effective in modeling various dynamic systems in the past [30, 31]. The output in a TS-FRBS is represented as a linear function of input variables. The value of this output indicates the confidence of the system for the input tested. Now, assuming the output variable y^j is a linear function of the input variables for the j^{th} rule, the output function $g(\cdot)$ can be represented as:

$$y^j = g(x_1^j, ..., x_q^j) = p_0^j + p_1^j \cdot x_1^j + ... + p_q^j \cdot x_q^j \tag{4}$$

where $p_0^j, p_1^j ... p_q^j$ are the coefficients of the function $g(\cdot)$, and q is the number of input dimensions. Given the estimated GMFs and the sample training data, the output parameters $p_0^j, p_1^j ..., p_q^j$ are computed by optimization with respect to the training data and this optimization reduces to a linear least square estimation problem as described in [29]. By solving this least square problem we obtain the parameters for output function, i.e., the consequent part of the rule. Therefore, with the GMFs, the fuzzy rules, and the model for the output variable learned from the training data, characterization of the fuzzy system is complete. Next, we describe how the fuzzy rules are combined to infer output values for unknown input data.

5.3 Fuzzy Rule-Based Feature Classification

The output response of a TS-FRBS is represented as a linear function of input variables. Formally, given a specific input $(x_1, ..., x_q)$, and a set of fuzzy rules R^j, $(j = 1, 2, .., c)$, the value of output y^j is inferred as follows. First the input is evaluated through each of the rules and a degree of match is computed which is called the *firing strength* of that rule for the input. The firing strength α^j of j^{th} rule is computed as:

$$\alpha^j = (gmf_1^j(x_1^j) \ \wedge \ gmf_2^j(x_2^j) \ \ ... \ \ \wedge \ \ gmf_q^j(x_q^j)) \tag{5}$$

where gmf_1^j, gmf_2^j ... gmf_q^j are the membership functions of the form described in Equation 2, $\wedge$ is a fuzzy conjunction operator. We have used the "AND" as the fuzzy conjunction operator since all the sub-clauses in the rule need to satisfy simultaneously. This fuzzy "AND" operator is also known as the "Product t-norm" and is typically realized by algebraic product of the membership values. Intuitively, the firing strength estimates the degree of match of rule R^j for the given input by conjunction of the membership contributions coming from the evaluation of the sub-clauses using the Product t-norm. Hence, if most of the sub-clauses of a rule have satisfied strongly for a given input, then the firing strength of that rule for the input will be high. Note that, for any given input, the fuzzy rule base produces j output values and the final output response y for that input is computed as the average of all y^j values weighted by their firing strengths as:

$$y = \frac{\sum_{j=1}^{c} \alpha^j \cdot y^j}{\sum_{j=1}^{c} \alpha^j} \tag{6}$$

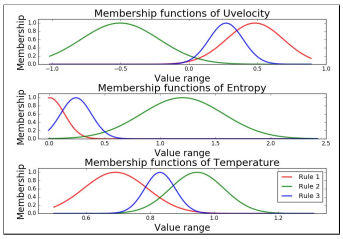


Figure 7: The Gaussian membership functions (GMF) generated using sample data collected from a simulation run with CMF=13.8 kg/s. Each color indicates membership functions of a rule in the image.

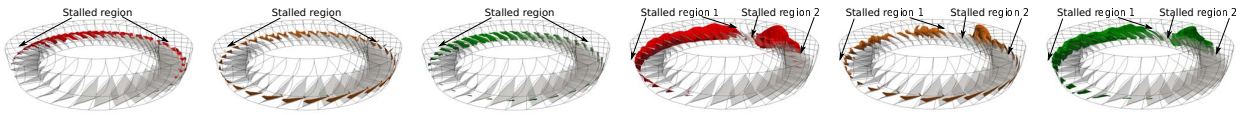
To explain the above inference algorithm, the example 2D synthetic data are shown in Figure 5a is used. The estimated GMFs presented in Figure 5c. The blue GMFs represent rule 1 and the orange GMFs represent rule 2. Given an unknown input point $(1.7, 7.8)$, as it is closer to blue cluster, it will generate high firing strength for rule 1 and low firing strength for rule 2. Furthermore, we also expect the final output value, i.e., the fuzzy classification score to be very close to zero since the points centered at $(2, 8)$ in the training set was assigned output value zero. By applying the inference algorithm presented above, we obtain the output value as 0.0043, which is very close to zero as expected. Similarly, when another input point $(7.1, 2.6)$ is tested, the output is 0.8540 which is close to 1 and can be considered as part of a cluster 2. Finally, when testing an input point $(5, 5)$ which cannot be classified strongly to any of the clusters, the fuzzy system produces an output of 0.4921 indicating its fuzzy classification. For conditions like these, when a hard classification is not possible, use of a fuzzy system helps us to perform a confidence driven classification considering the uncertainty.

6 In Situ Feature Detection for Stall Analysis

We employ the aforementioned fuzzy rule-based system for the classification of stall impacted regions. We construct the rule base using an expert selected training set containing observations from both stalled and stable regions as discussed in Section 4 and 5. The training is first done off-line using a pre-generated simulation data set which contains the stalled condition. After all the parameters of the fuzzy system and the rules are estimated, we employ the inference algorithm in the *in situ* environment for other unknown simulation data sets. Note that, we directly apply the learned system from one pre-generated simulation data to other cases without repeating the learning again. We also estimate the local mass flow rate for each passage. It is hypothesized that, as the stall cells block the normal airflow through passages, the mass flow rate of the stalled passages will be lower compared to the stable ones. So, the analysis of the local mass flow rate over time can be used as a complementary source of information to that of the fuzzy system based stall detection for enhancing the overall effectiveness and robustness of our stall analysis. Below we discuss how we estimate these two measures during the *in situ* run and enable flexible stall exploration in detail.

6.1 Fuzzy Rule-Based *In Situ* Stall Prediction

The fuzzy inference system works on each data point by assigning a value which reflects its "stallness". The input to the fuzzy inference system for each point is a 3-tuple: {Uvel, entropy, temperature}. After the application of the inference algorithm on all the points, we can construct a new classification field where the scalar value at each point will reflect the stallness of the point. Higher values of stallness indicate higher chances of being part of a stall cell. As stated in Section 3.2, one of the primary goals of the expert is to study the evolution of stall impacted regions over time, and furthermore, the expert wants to know whether these stalled regions are detected near the blade tip or close to the hub. To answer these questions, we summarize the classified stallness field in two specific



(a) Isosurface of Entropy = 1.0 at T = 375. **(b)** Isosurface of Uvel = -0.2 at T = 375. **(c)** Isosurface of Temperature = 1.0 at T = 375. **(d)** Isosurface of Entropy = 1.0 at T = 560. **(e)** Isosurface of Uvel = -0.2 at T = 560. **(f)** Isosurface of Temperature = 1.0 at T = 560.

Figure 8: Visualization of entropy, Uvel, and temperature isosurfaces for locating the stalled regions in the CMF=13.8 kg/s data set. Figure 8a, 8b, and 8c show the isosurfaces at T = 375 when the global mass flow rate drops as shown in the bottom left panel of Figure 6 indicating stall inception. Figure 8d, 8e, and 8f depict the isosurfaces of the same variables at later time, T = 560 when the stall is well developed. It can be seen that two separate stalled regions are formed as marked in the images.

ways. First, we aggregate the points that have stallness greater than an expert specified stallness threshold for a passage which reflects the stallness of the whole passage. Secondly, to estimate their relative spatial extent inside each passage, we aggregate all the points along the axis spanning from hub to tip by counting the number of points for each segment along this direction using the same stallness threshold used above. These two types of aggregated information for each passage are stored in the disk and visualized during post-hoc analysis for studying the evolution of stall phenomenon.

6.2 Local Mass Flow Rate Estimation

Besides classification of data using their predicted stallness, we further estimate the local mass flow rate for each passage during the simulation run. **Mass flow rate** is a measure of the air mass passing through the compressor stage per unit time [19] and can be computed as $\dot{m} = \rho\vec{v}\cdot\vec{A}$, where ρ is the density, $\vec{A}$ is the area vector of a flow path cross-section of the inlet or exit of the compressor, and $\vec{v}$ is the flow velocity. The global value of mass flow rate in stable condition remains consistent over time, however, when the simulation generates stall, it drops rapidly. While the indication of stall measured by global mass flow rate is discernible, it appears too late to apply any stall preventive measures. Hence, we estimate per passage local mass flow rate for each time step. The scientist hypothesizes that by analyzing the local mass flow rate evolution among blade passages, the indications, and development of stall will be more effective compared to the global mass flow.

7 Visual Interface for Studying Stall Evolution

For post-hoc visual analysis of stall, we have developed an interface shown in Figure 6. The goal is to provide insights about the evolution of the simulation by presenting feature specific information derived *in situ* using fuzzy rule system and local mass flow rate. Our interface depicts information through three different views:

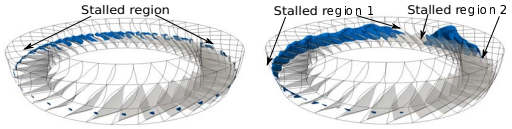
Evolution of temporal stallness. The temporal overview shows the time-varying evolution of stall impacted regions detected by the fuzzy system prediction (top chart in Figure 6). The x-axis represents time and the y-axis is mapped to the blade passages. Each cell in this plot reflects the degree of stallness of the passage at a time step. The primary purpose of this plot is to present the overall temporal trend of the detected stalled regions. This helps the expert to (a) Locate the time step ranges when signs of stall start appearing; (b) The blade passages that get affected by the stall; (c) The overall temporal pattern of the propagation of the detected stalled regions from one passage to another. However, note that, this plot does not reveal the location of the detected regions in spatial domain.

Spatial view of stallness. To show the spatial locations of the detected stalled regions for each time step, a 2D circular plot is presented in the top right panel. Here, the detected regions in each passage are laid out radially center-to-outward, i.e., from hub to the tip of the compressor stage. Presenting information in this intuitive format which has a direct correspondence to the actual physical rotor helps the expert to easily comprehend the spatial organization of the stall cells at each time step. This is why a radial layout was preferred in our work compared to a linear layout for showing this information. Furthermore, a radial layout also offers a compact use of the screen space for creating visualizations. From Figure 6, we observe that a majority of the stalled regions are detected near the blade tip. Note that, this spatial aggregated 2D view shows only the relative spatial extent of the stalled regions from the hub to the tip. In this view, at each level of aggregation from the hub to the tip, the color of each semi-circular segment reflects the number of stalled points. However, it does not distinguish regions within each semi-circular band and uses a single aggregated value to represent the stallness. To show more detailed locations in this spatial view, we can segment each semi-circle into further smaller segments and keep the number of stalled points from each such small segment. This will increase the overall storage of our in situ summarized information while making the visualization more precise.

Evolution of local mass flow rate. The information derived from mass flow rate is shown on the bottom left panel where the x-axis is time steps. The solid blue line indicates the global mass flow rate and the y-axis on the left show its value range. Here, we compute the relative deviation of local mass flow rate with respect to the mean local mass flow rate of the rotor. According to the expert, in a stable condition, the local mass flow for each passage remains identical due to the axisymmetry property of the rotor, so any passage that deviates significantly from the expected (mean) mass flow is abnormal. For each time step, we estimate the relative deviation of local mass flow rate $\dot{m}dev_i$ for the i^{th} passage as: $(\dot{m}_i - \overline{\dot{m}})/\dot{m}_i$, where $\overline{\dot{m}}$ is the mean local mass flow at a given time step, and $\dot{m}_i$ is the mass flow of the i^{th} passage. Note that, the mass flow deviations which are smaller than the mean, i.e., the negative mass flow deviation values are of our interest, since mass flow of the stalled passages drops during the stall. The relative mass flow deviation values computed per passage are shown using a heat map based visualization as seen in Figure 6. Each cell in this heat map reflects the relative mass flow deviation of a passage at a time step. The passage ids are depicted on the right side of the chart. Note that, each cell in this plot has a one-to-one correspondence with the above stallness chart. Laying out the mass flow information in this way was preferred by the expert since the expert could easily investigate the mass flow deviation values of important stalled regions identified from the stallness chart in any time step. Observe that the relative mass flow deviation values convey more information of stall inception where the asymmetry in the mass flow among the passages are clearly seen from the color variation in the heat map.

8 Verification Through Domain Expert Evaluation

We verify the proposed method by applying it to a known data set with corrected mass flow rate (CMF) = 13.8 kg/s. The simulation was run for 4 revolutions and raw data for every 25^{th} time step was stored resulting in 576 time steps. The rotor in TURBO consists of 36 passages with a spatial resolution of $151 \times 71 \times 56$ for each passage. The Gaussian membership functions (GMFs) obtained after the training, are shown in Figure 7. From the GMFs, one can observe that data points coming from stable regions will find strong degree of match with rules 1 and 3 as they will have high firing strength from these rules, and similarly low degree of match with



(a) Detected Stallness at T = 375. **(b)** Detected Stallness at T = 560.

Figure 9: Visualization of isosurfaces of the fuzzy system predicted stallness field. Figure 9a shows isosurface of 0.8 at T = 375, and Figure 9b shows isosurface of 0.8 at T = 560. The detected regions at both of these time steps correspond well with the regions identified in Figure 8 validating the correctness of our method.

rule 2, resulting in a low stallness output from the fuzzy inference algorithm. This is because the stable regions demonstrate high and positive Uvel and low to moderate entropy and temperature values. So, in fuzzy logic, linguistic terms rule 1 can be articulated as IF Uvel is high AND entropy is low AND temperature is low THEN the predicted Stallness is low (close to 0.1 as marked by the expert). Similarly, rule 3 will be described as IF Uvel is moderate to high AND entropy is low AND temperature is moderate THEN the predicted Stallness is low. In contrast, the rule 2 can be interpreted as IF Uvel is negative and low AND entropy is high AND temperature is high THEN the predicted Stallness is high (close to 0.9 as marked by the expert). Note that, in this case, the stalled points will have high firing strength, i.e., close degree of match with rule 2, whereas low degree of match with rules 1 and 3 which will produce high stallness output.

Accuracy study of the fuzzy system. We conducted an accuracy analysis using a cross-validation technique called the repeated random sub-sampling validation [12]. The sample set consisted of 573460 points. The points were collected from both the stalled (stallness of 0.9) and stable regions (stallness of 0.1) from several time steps of a known data set with CMF = 13.8 kg/s. In this cross-validation, we randomly divided the data into two equal groups, and one was used for training and the other for validation. We repeated this process 1000 times to obtain robust accuracy results by taking the average (expected) outcome of all the trials. We found that the stallness of 92.67% points were predicted within a small deviation of 0.1, and 97.47% points were within the deviation of 0.15 of the user marked stallness values. Besides this, we also measured the percentage of false positive points, i.e., the points predicted as stalled while considering a deviation of 0.1, but the points should not be considered as stalled from the labels in the test data. We found that the average percentage of false positive points is 3.95%. Furthermore, if we increase the stallness deviation to 0.15, the percentage of false positive points reduces to 1.043%.

Verification using the visual-analytics system. Figure 6 depicts the result of our method when applied to all the time steps of the run with CMF=13.8 kg/s. The temporal pattern of the predicted stallness, shown in the top left panel, demonstrates that our method is able to capture the overall evolution of stall. This plot also shows that passages ranging from 5-15 start showing signs of stall around time step 200 where the dark blue cells reflect passages with larger stalled regions compared to others. In the bottom left panel, the global mass flow, indicated by the solid blue line, confirms the occurrence of the stall when it starts to drop around time step 375. The heat map colored by the relative deviation of local mass flow rate in the same panel shows that several passages deviate from the expected mass flow breaking the uniform flow through all the passages. This asymmetry in mass flow, caused by lower local mass flow rates, is detected around time step 325 (the scattered reddish cells around passages 10-15) indicating a potential stall inception. The relative mass flow finally drops significantly for those passages (the yellow regions) for later time steps as the stall grows.

Next, we provide a 2D radial plot based visualization of the rotor on the right in Figure 6 where the spatial locations of the stalled regions can be studied. Investigation of stalled regions in this way

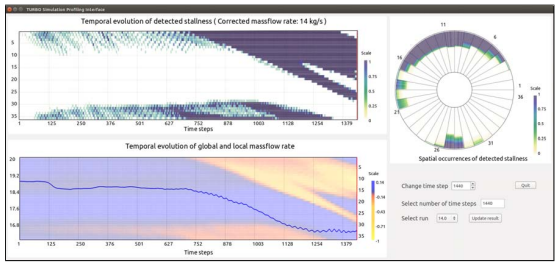


Figure 10: Result of simulation run with CMF=14.0 kg/s. This resulted in a stalled condition which is visible from the stallness and mass flow deviation plot.

for each time step allows the expert to verify that the stall cells are formed close to the blade tip. To verify this, we show isosurfaces of a high entropy value of 1.0, a negative Uvel value of -0.2, and a high temperature value of 1.0 in Figures 8a, 8b, and 8c respectively. All these surfaces indicate a common region (as marked in the figures) where the stall is happening. Figure 9a shows the isosurface extracted from the predicted stallness field using an expert specified high stallness value of 0.8 at T = 375. It is observed that the predicted region matches with the regions obtained from the data quite well in Figures 8a - 8c. Next, Figures 8d, 8e, and 8f show the isosurfaces of a later time step when the simulation has transitioned into a deep stall. We observe that there are two separate stall cells, and our method in Figure 9b is able to capture it. These two stalled regions are also marked in the stallness and the mass flow chart in Figure 6. A similar observation is seen in the spatial view for a specific time step. So, by studying the local mass flow chart and the fuzzy system predicted stallness map, the expert concluded that our method provides a comprehensive view of the evolution of stall.

9 In Situ Stall Analysis With Varying Parameters

We worked closely with an expert having more than 25 years of experience in flow simulations and is one of the primary developers of TURBO. Feedback was collected through regular bi-weekly meetings with the expert. The learning algorithm took 1.027 seconds to generate the fuzzy system with 3 rules using 573460 training samples. Note that, the appropriate number of rules are dependent on the target applications [29], and since we want to distinguish points which are either stalled or not, the minimum number of rules can be 2. However, that may result in high variability in each cluster. So, we tested with $3 \sim 5$ rules and found that the results are similar with minor variations without changing the overall outcome. Also, a higher number of rules increases computational cost of the algorithm. So 3 rules for all the experiments were used producing satisfactory results. The *in situ* code, developed in C++, is linked to TURBO as a new module. The *in situ* routines directly access the simulation memory minimizing any additional data copy.

9.1 Simulation with Stall (CMF = 14.0 kg/s)

We first tested our system using a new run with CMF = 14.0 kg/s. It was expected that this run would stall since, in a previous study [9], a run with a higher CMF of 14.2 kg/s produced stall. We simulated 4 revolutions resulting in a fully developed stall. As each revolution of TURBO runs 3600 iterations, for 4 revolutions, a total of 14400 iterations were run. *In situ* call was made at every 10^{th} iteration resulting in 1440 time steps. Here, the time step numbers used are in the units of tenths of simulation iterations due to the sampling rate of *in situ* call. At every 10^{th} time step, we applied our fuzzy rule-based algorithm and estimated the local mass flow rates. The stallness values estimated by the fuzzy algorithm were then aggregated *in situ* as discussed earlier in Section 6, and finally, the task-specific aggregated information was stored.

As can be seen from the stallness and local mass flow deviation charts in Figure 10, the simulation produces a stalled condition. In-

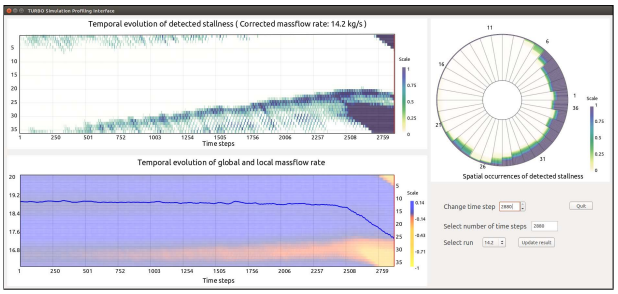


Figure 11: Result of simulation run with CMF=14.2 kg/s. Provided CMF value drives the simulation into a stalled state which our proposed method is able to detect correctly.

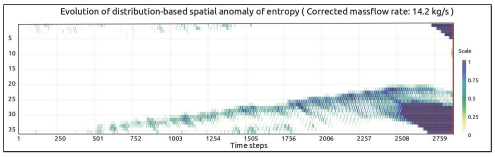


Figure 12: Distribution-based spatial anomaly plot of entropy variable for the run with CMF=14.2 kg/s, proposed in [13].

dications of a potential stall start appearing around time step 70 in the stallness overview chart. Finally, around time step 750 the stall happens when dark blue regions become persistent. The global mass flow rate also starts dropping around the same time step, and the local mass flow becomes significantly asymmetric, reflecting the occurrence of stall. Also, similar to the stallness pattern of CMF = 13.8 kg/s run, this run too produces two separate stalled regions within passages 1-12 and 15-20 starting around time step 1240. From these observations, the expert is able to verify the hypothesis that the CMF value of 14.0 kg/s indeed produces a stall.

9.2 Simulation with Stall (CMF = 14.2 kg/s)

Next, we tested our method using a run with CMF = 14.2 kg/s which is a known stalled condition. The simulation was run for 8 revolutions requiring us to process 2880 time steps. The results are presented in Figure 11 where the development of the stall is visible from both the stallness and mass flow deviation plots. The early indication of the stall is seen around time step 350 in the stallness plot. This grows consistently over time, and around time step 2508 the major flow breakdown happens reflected by the persistent and blue regions covering passage range 20-30. The consistent drop in global mass flow around time step 2508 validates this event. Note that, the local mass deviation chart, in this case, starts showing indications of a future stall starting around time step 500 onwards within passages 30 - 33. The deviation of mass flow gradually becomes large and persistent reflecting the evolution of flow instability. The expert explained that this deviation is caused by the drop in mass flow rate in the stalled regions where stall cells act as blockages. The spatial plot shows the stalled regions in a radial map for a specific time step confirming that the stalled regions are concentrated on the blade tips. Finally, we observe that compared to the previous run with CMF = 13.8 kg/s and CMF = 14.2 kg/s, this case produces stall in a different set of passages.

Comparison with existing methods. We computed the spatial distribution-based anomaly of the entropy variable proposed in [13] for this run. The anomaly plot is presented in Figure 12 which shows a similar stall pattern to that of our stallness chart in Figure 11. This confirms that our method is able to detect the locations of the stalled regions correctly. However, note that our method works *in situ*, whereas the distribution-anomaly would require additional computation time in the post-hoc phase to produce the results. Also,

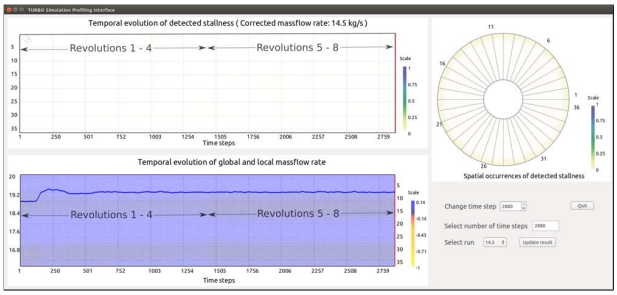


Figure 13: Result of simulation run with CMF=14.5 kg/s. The simulation resulted in a stable run which is observed from uniform mass flow chart and clean stallness plot.

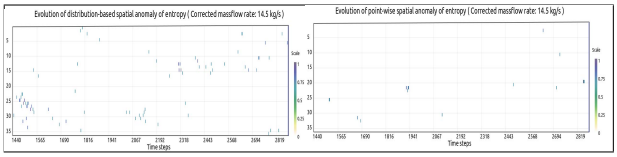


Figure 14: Spatial anomaly plot of entropy for the simulation run with CMF = 14.5 kg/s. It shows anomaly plot for revolutions 5-8. The left chart shows distribution-based spatial anomaly [13], and the right chart provides the point-wise spatial anomaly [9].

the spatial distribution anomaly based method needs data from all the passages, so, it would require additional data communication during the *in situ* processing further impacting the performance.

9.3 Simulation with Unknown Parameter Configuration (CMF = 14.5 kg/s)

After the case with CMF = 14.2 kg/s, we present the result of the run with CMF = 14.5 kg/s where the outcome was unknown. The simulation was initially run for 4 revolutions. By observing the result of first 4 revolutions, we found that the stallness plot was clean and the mass flow deviation plot was uniform. This can be seen in Figure 13 for time step ranges of first 4 revolutions as marked. To verify that this run was indeed a steady state case, the expert suggested continuing to run it for another 4 revolutions. As we can see from Figure 13, the revolutions 5-8 continued as a stable state.

Further verification by comparison with existing methods. We computed spatial distribution-based anomaly adopted by Dutta et al. [13], and spatial point-wise anomaly used by Chen et al. [9] for revolutions 5-8 of the 14.5 kg/s run. Both the distribution-based anomaly (left chart in Figure 14) and point-wise anomaly (right chart in Figure 14) show clean plots with sporadic noises which do not reflect stall. Hence, with this comparison and the results obtained by our method, the expert finally concluded that 14.5 kg/s run was a stable run. However, both these spatial anomaly methods require data from all the passages for each block to estimate the value of anomaly. Therefore, these methods are not inherently *in situ* friendly as they would require additional data communication. In contrast, the proposed method is parallel in nature which makes it a preferable choice for the expert when timely analyzable result extraction from an ensemble of simulation runs is essential.

9.4 Simulation without Stall (CMF = 16.0 kg/s)

Finally, we tested our method to another known case with CMF = 16.0 kg/s. The simulation was run for 4 revolutions to produce a stable data set. The stallness plot obtained was clean (similar to the CMF = 14.5 kg/s case) and the mass flow deviation plot was uniform throughout the run with a stable global mass flow rate over all the time steps validating it as a stable run.

Summary of stall analysis results. The above analyses allow the expert to obtain a detailed understanding of the influence of the parameter CMF on the simulation outcome. With our technique, the

Table 1: *In situ* computation time of the proposed method.

# revs.	Simulation time (hrs.)	Simulation I/O (sampled every 10th time step) (hrs.)	In situ fuzzy inference (hrs.)	In situ mass flow estimation (hrs.)	In situ I/O (mins.)
4	22.6	0.48	0.208	0.00022	0.000137
8	45.41	0.98	0.417	0.00043	0.00028

Table 2: Percentage computation timing of *in situ* processing.

4 revolutions		8 revolutions	
Simulation	In situ	Simulation	In situ
99.084%	0.916%	99.089%	0.911%

expert can perform ensemble parameter studies in a timely manner without worrying about the expensive I/O and large-scale data management issues. By studying the stall patterns generated by CMF = 13.8 kg/s (Figure 6), CMF = 14.0 kg/s (Figure 10), and CMF = 14.2 kg/s (Figure 11) runs, it is observed that the local mass flow shows earlier signs of a potential stall for the 14.0 and 14.2 runs, whereas, for the 13.8 case, the indication of stall comes quite late. In contrast, out fuzzy system based stall detection is found to be superior in detecting earlier signs of stall compared to the local mass flow for all of these cases. It is found that the stall can develop in different passage ranges and impact a different number of passages as well. In contrast, the stable runs produce clean stallness plots and uniform mass flow charts for 14.5 and 16.0 cases. Through our study, the unexplored range of the CMF parameter while searching for the true stall point is reduced from $14.2 - 16.0$ kg/s to a much smaller range of $14.2 - 14.5$ kg/s which reduces the uncertainty in the outcome of the simulation for the expert. The expert can use this knowledge for model refinement, and more importantly, tune the throttle setting for a safer operation of the engine.

10 Performance Study

The *in situ* experiments were conducted using a cluster, Oakley [8], at the Ohio Supercomputer Center, consisting of 694 nodes with Intel Xeon x5650 CPUs (12 cores per node) and 48 GB of memory per node. The *in situ* computation and I/O were performed on Lustre, a high-performance parallel file system.

Storage performance. The output raw data size of a single time step of the rotor is 690MB. Since each revolution of the simulation contains 3600 time steps, the total size of output data if all the time steps are stored will be 2.484 TB. Hence, without *in situ*, the expert used to skip every $25 \sim 35$ time steps to reduce the data size. In this work, we made *in situ* call at every 10^{th} time step which required us to process 360 time steps for each revolution, i.e., 248.4 GB data. Each of our experiment runs continued to a maximum of 8 revolutions which would require 1.9872 TB data storage for each run. Processing such large volume of data would take a significant amount of time and effort to produce analyzable results. In contrast, by performing the analysis *in situ*, we output only aggregated and task-specific data which takes 89.58 MB of disk space for 8 revolutions. Using our output data, we are able to produce visualization results almost instantly which reveal details about the simulation.

Computation time savings. Table 1 shows the *in situ* computation times of different components and I/O. We observe that compared to the simulation time, our *in situ* fuzzy inference method and the estimation of passage-wise mass flow rate take only a small fraction of the additional time. Also, we accumulate the information at each node for all the time steps and write out the aggregated stallness and mass flow values at the final time step. This reduces the *in situ* I/O significantly. From Table 2, we observe that the *in situ* processing only takes on average 0.914% of the total computation time which reflects that our method is well suited for *in situ* analysis. However, in the absence of *in situ* processing, we would perform the off-line computation of the fuzzy rule-based feature prediction and local mass flow rate estimation. The timings of this post-hoc computation with raw data I/O were estimated on a standard Linux machine with an Intel core i7-2600 CPU, 16 GB of RAM are presented in Table 3. It is seen that with an increased number of revolutions, the raw I/O becomes the dominant component. Also, compared to the estimation of the mass flow rate, the fuzzy prediction takes a long time since the estimation of each rule's contribution requires an evaluation of GMFs. So, by performing the feature analysis *in situ*, we bypass the expensive post-hoc I/O and computation time and improve the overall performance of the exploration process.

Table 3: Off-line computation time in the absence of *in situ* analysis.

# revs.	Raw data I/O (hrs.)	Fuzzy inference (hrs.)	Local mass flow estimation (hrs.)
4	6.7	4.04	0.19
8	12.38	7.96	0.36

11 Discussion

The above results with thorough expert evaluation and the performance study demonstrate the efficacy of our method in robust stall analysis in large-scale flow data sets. Here, we propose an off-line learning and *in situ* prediction based analysis pathway. In the off-line phase, the fuzzy system learns the feature specific multivariate patterns from the expert identified regions. After learning from a known data set, the inference algorithm is employed *in situ* on the unknown cases when the simulation is run using different parameter settings. By performing *in situ* feature detection and task-specific information aggregation, we overcome the expensive I/O bottleneck, and output only reduced information. By comparing and contrasting the feature specific information, obtained from multiple runs, our technique reveals important details about the evolution of stall and its spatial spread. The local passage-wise mass flow also emerges as a more effective measure to locate the early flow asymmetry compared to the traditional global mass flow. Together with the spatial-temporal stallness charts, and local mass flow plots, our method is able to provide a comprehensive view of the simulation to the expert. We also derive new knowledge about the impact of the CMF parameter in stall inception that can potentially help in refining stall preventive technologies. Note that, the fuzzy rule-based feature detection is generalizable to other multi-field features when a precise descriptor is unavailable, and a predictive algorithm is required. The *in situ* friendly nature makes our method suitable for feature extraction from large-scale data sets in a timely manner.

12 Conclusion and Future Works

We demonstrate the efficacy of an *in situ* machine learning based exploration approach for robust feature analysis in large-scale simulations. Our method learns multivariate relations from the expert-highlighted regions in the data and generates several fuzzy rules which are then applied to the simulation with unknown parameter settings. Detection of the target feature for new simulations is done *in situ* which bypasses the expensive I/O and allows exploration of data in a timely manner. The effectiveness of our approach is shown by applying it for the detection of the stall in large flow simulations. In the future, we wish to employ our method for simulation steering. We also hope to apply it to other large-scale simulations for robustly detecting features under uncertainty.

Acknowledgments

This work was supported in part by NSF grants IIS- 1250752, IIS-1065025, and US Department of Energy grants DE- SC0007444, DE-DC0012495, program manager Lucy Nowell.

References

[1] J. Ahrens, S. Jourdain, P. OLeary, J. Patchett, D. H. Rogers, and M. Petersen. An image-based approach to extreme scale in situ visualization and analysis. In *SC14: International Conference for High Perfor-*

mance Computing, Networking, Storage and Analysis, pages 424–434, 2014.

[2] A. Banerjee, H. Hirsh, and T. Ellman. Inductive learning of feature-tracking rules for scientific visualization. In *Workshop on Machine Learning in Engineering (IJCAI-95*, 1995.

[3] A. C. Bauer, H. Abbasi, J. Ahrens, H. Childs, B. Geveci, S. Klasky, K. Moreland, P. O'Leary, V. Vishwanath, B. Whitlock, and E. W. Bethel. In Situ Methods, Infrastructures, and Applications on High Performance Computing Platforms. *Computer Graphics Forum*, 2016.

[4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.

[5] J. Bernard, M. Hutter, M. Zeppelzauer, D. Fellner, and M. Sedlmair. Comparing visual-interactive labeling with active learning: An experimental study. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):298–308, Jan 2018.

[6] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 1981.

[7] A. Biswas, D. Thompson, W. He, Q. Deng, C.-M. Chen, H.-W. Shen, R. Machiraju, and A. Rangarajan. An uncertainty-driven approach to vortex analysis using oracle consensus and spatial proximity. In *2015 IEEE Pacific Visualization Symposium (PacificVis)*, pages 223–230, April 2015.

[8] O. S. Center. Oakley supercomputer. `http://osc.edu/ark:/19495/hpc0cvqn`, 2012.

[9] C.-M. Chen, S. Dutta, X. Liu, G. Heinlein, H.-W. Shen, and J.-P. Chen. Visualization and analysis of rotating stall for transonic jet engine simulation. *IEEE Trans. on Vis. and Comp. Graphics*, 22(1):847–856, 2016.

[10] J.-P. Chen, M. D. Hathaway, and G. P. Herrick. Prestall behavior of a transonic axial compressor stage via time-accurate numerical simulation. *Journal of Turbomachinery*, 130(4):041014, 2008.

[11] H. Childs. Data exploration at the exascale. *Supercomputing frontiers and innovations*, 2(3), 2015.

[12] W. Dubitzky, M. Granzow, and D. Berrar. *Fundamentals of Data Mining in Genomics and Proteomics*. Springer US, 2007.

[13] S. Dutta, C. M. Chen, G. Heinlein, H. W. Shen, and J. P. Chen. In situ distribution guided analysis and visualization of transonic jet engine simulations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):811–820, Jan 2017.

[14] S. Dutta and H.-W. Shen. Distribution driven extraction and tracking of features for time-varying data analysis. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):837–846, 2016.

[15] S. Dutta, J. Woodring, H. W. Shen, J. P. Chen, and J. Ahrens. Homogeneity guided probabilistic data summaries for analysis and visualization of large-scale data sets. In *2017 IEEE Pacific Visualization Symposium (PacificVis)*, pages 111–120, April 2017.

[16] N. Fabian, K. Moreland, D. Thompson, A. C. Bauer, P. Marion, B. Gevecik, M. Rasquin, and K. E. Jansen. The paraview coprocessing library: A scalable, general purpose in situ visualization library. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 89–96, 2011.

[17] N. Fout and K.-L. Ma. Fuzzy volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2335–2344, 2012.

[18] R. Fuchs, J. Waser, and M. E. Groller. Visual human+machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1327–1334, Nov 2009.

[19] M. D. Hathaway, G. Herrick, J. Chen, and R. Webster. Time accurate unsteady simulation of the stall inception process in the compression system of a us army helicopter gas turbine engine. In *Users Group Conference, 2004*, pages 166–177, June 2004.

[20] Y. He, M. Mirzargar, S. Hudson, R. Kirby, and R. Whitaker. An uncertainty visualization technique using possibility theory: Possibilistic marching cubes. *International Journal for Uncertainty Quantification*, 5(5):433–451, 2015.

[21] C. Heinzl and S. Stappen. Star: Visual computing in materials science. *Computer Graphics Forum*, 36(3):647–666, 2017.

[22] S. Jiang, X. Zhou, T. Kirchhausen, and S. T. C. Wong. Tracking molecular particles in live cells using fuzzy rule-based system. *Cytometry Part A*, 71A(8):576–584, 2007.

[23] A. Kumpf, B. Tost, M. Baumgart, M. Riemer, R. Westermann, and M. Rautenhaus. Visualizing confidence in cluster-based ensemble weather forecast analyses. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):109–119, Jan 2018.

[24] H. Lehmann and B. Jung. In-situ multi-resolution and temporal data compression for visual exploration of large-scale scientific simulations. In *IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV), 2014*, pages 51–58, 2014.

[25] X. Li and H.-W. Shen. Adaptive Volume Rendering using Fuzzy Logic Control. In *IEEE VGTC Symposium on Visualization*, 2001.

[26] J. F. Lofstead, S. Klasky, K. Schwan, N. Podhorszki, and C. Jin. Flexible IO and Integration for Scientific Codes Through the Adaptable IO System (ADIOS). In *Proceedings of the 6th International Workshop on Challenges of Large Applications in Distributed Environments*, CLADE '08, pages 15–24. ACM, 2008.

[27] K. L. Ma. Machine learning to boost the next generation of visualization technology. *IEEE Computer Graphics and Applications*, 27(5):6–9, Sept 2007.

[28] C. Muelder and K. L. Ma. Interactive feature extraction and tracking by utilizing region coherency. In *2009 IEEE Pacific Visualization Symposium*, pages 17–24, April 2009.

[29] N. Pal, V. Eluri, and G. Mandal. Fuzzy logic approaches to structure preserving dimensionality reduction. *Fuzzy Systems, IEEE Transactions on*, 10(3):277–286, Jun 2002.

[30] P. Purkait, N. R. Pal, and B. Chanda. A fuzzy-rule-based approach for single frame super resolution. *IEEE Transactions on Image Processing*, 23(5):2277–2290, May 2014.

[31] T. Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 2004.

[32] S. Shafii, H. Obermaier, R. Linn, E. Koo, M. Hlawitschka, C. Garth, B. Hamann, and K. I. Joy. Visualization and analysis of vortex-turbine intersections in wind farms. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1579–1591, 2013.

[33] F. Y. Tzeng, E. B. Lum, and K. L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Vis. and Computer Graphics*, 11(3):273–284, May 2005.

[34] V. Vishwanath, M. Hereld, and M. E. Papka. Toward simulation-time data analysis and i/o acceleration on leadership-class systems. In *2011 IEEE Symposium on Large Data Analysis and Visualization (LDAV)*, pages 9–14, 2011.

[35] J. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, Jan 2018.

[36] B. Whitlock, J. M. Favre, and J. S. Meredith. Parallel in situ coupling of simulation with a fully featured visualization system. In *Proceedings of the 11th Eurographics Conference on Parallel Graphics and Visualization*, EGPGV '11, pages 101–109, 2011.

[37] P. C. Wong, H. W. Shen, C. R. Johnson, C. Chen, and R. B. Ross. The top 10 challenges in extreme-scale visual analytics. *IEEE Computer Graphics and Applications*, 32(4):63–67, July 2012.

[38] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann. In-situ sampling of a large-scale particle simulation for interactive visualization and analysis. In *Proceedings of the 13th Eurographics / IEEE - VGTC Conference on Visualization*, pages 1151–1160. Eurographics Association, 2011.

[39] J. Woodring, J. Ahrens, T. J. Tautges, T. Peterka, V. Vishwanath, and B. Geveci. On-demand unstructured mesh translation for reducing memory pressure during in situ analysis. In *Proceedings of the 8th International Workshop on Ultrascale Visualization*, pages 3:1–3:8. ACM, 2013.

[40] J. Woodring, M. Petersen, A. Schmeißer, J. Patchett, J. Ahrens, and H. Hagen. In situ eddy analysis in a high-resolution ocean climate model. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):857–866, 2016.

[41] H. Yu, C. Wang, R. W. Grout, J. H. Chen, and K. L. Ma. In situ visualization for large-scale combustion simulations. *IEEE Computer Graphics and Applications*, 30(3):45–57, 2010.

[42] L. Zhang, Q. Deng, R. Machiraju, A. Rangarajan, D. Thompson, D. K. Walters, and H.-W. Shen. Boosting techniques for physics-based vortex detection. *Computer Graphics Forum*, 33(1):282–293, 2014.