Successful Machine Learning Strategies in an Environment of Intermittent Data Availability

Timothy Seabrook  
University of Oxford

I. ABSTRACT

This report introduces an online, fully adaptive Hybrid-ANN approach for the estimation of product properties across multiple crude distillation processes. An extension of eTS+ is considered, utilizing recursive density estimation (RDE), eClustering+ and fuzzily weighted recursive least-squares (fwRLS) for one-step prediction of fuel quality in a data-sparse, sensor-heavy environment.[49] eTS+ has been improved with mutual information based input variable selection, online monitoring of regressor utility, and online normalization to produce a system evolving in both features considered and number of rules for autonomous application to any refinery. The solution has only optional requirement for user-defined parameters and expert knowledge, and produces a transparent model for product estimation.

Keywords: Online, evolving, fully-adaptive, soft-sensors, ANN, fuzzy inference, mutual information, LASSO, recursive, prediction, least-squares, time-independent, data-sparse, autonomous, IVS, transparent, data-driven, clustering, eClustering+, eTS+, fWRLS, MIMO, systems modeling

II. INTRODUCTION

There exists a demand for improvements to modeling techniques in oil refinement operations. Oil and Gas industries are under increasing environmental and economic pressure to improve the efficiency of fuel production, both with the goal of improving product quality yields and for the reduction of pollutant side-products. The introduction of EU emissions targets for 2020 and 2030, along with more stringent specifications for fuel products, water management and waste production, calls for new tools offering a finer precision of process monitoring and control.[50]

Industrial plants have utilized inferential/soft sensors for real time prediction and monitoring of system variables since the 1990s. Through working in tandem with hardware sensors, software-based techniques have introduced great improvements in fault-detection, energy management and process control that not only serve to optimize performance, but also ensure the safety of refinery operations. Transparent soft-sensors can provide a greater understanding of process dynamics[40], leading to higher predictability and repeatability of desired results,[22] as well as reduced risk from process upsets. The subsequent implementation of soft sensors across thousands of applications and in various industries has resulted in an economic benefit estimated in the hundreds of millions of dollars.[47]

Currently, O&G operations often rely on static linear models based on ‘best-fit’ regression of first-principles models. These give a generalized indication of the relationship between column temperatures, pressure and flow and the resulting quality of fuels, but require frequent bias updates from costly lab sample analysis to remain accurate in lieu of dynamic crude feed properties. In addition, any modification to a system can result in drastic difference, requiring the reformation of first-principles models from expert knowledge.

A. Scope & Objectives

This report will explore the development of an online Evolving Intelligent Sensor (EIS) to improve on current static models for application across multiple refinery operations. It is necessary for the proposed system to be able to account for non-linear complexity in the underlying system process, as well as the presence of shifts and drifts in operating conditions for long-term adaptability. In addition, the developed system should be invariant to changes in sensor configurations to remain effective for use in unique and modified plants.

The objectives of this study are summarized as follows:

1) Demonstrate the successful use of data-driven models to estimate physical properties from refinery operating data.
2) Investigate the capability for model adaptability across long-term data streams.
3) Explore applicability of the system and derived models across multiple refineries.

Lab Analysis data has been provided for a case study coking refinery crude unit. Alternative crude units vary greatly in plant configuration, with different sensor placement and availability, differences in typical crude feeds and specification goals.

Features of interest for this report are the relative quality of four distillation column side-draws products, measured by their inflammability index: SS1 (HNV/LK); SS2 (2CL HK); SS3 (3CL LGO); SS4 (4CL HGO)
Lab measurements for each product are provided every 1-3 days for model training and verification, along with 105 plant sensor values at the time the sample was taken.

71 samples were provided for SS1.
84 samples were provided for SS2.
138 samples were provided for SS3.
148 samples were provided for SS4.

When live, the plant provides real-time sensor updates, over which product quality predictions are to be made.[33] Sensors may abruptly fail temporarily, be removed from use, or degrade in accuracy/utility over time. In addition, new sensors may be installed in a plant, for which there is no training data.

B. Contribution
eTS+ is benefited by the introduction of a combined offline-feature selection and online feature pruning methodology to improve adaptability for a hands-free method for system identification.

Minor improvements have been made across the formation of antecedents and consequents, using data-derived universally applicable thresholds. An extension to online normalization corrects model parameters to adapt for a dynamic data stream.

Anomaly and Fault detection has been introduced to detect and manage data stream anomalies and failed sensors.

Applicability for multiple plant configurations has been achieved with an operator-free approach to input variable selection, model derivation and feature prediction.

III. Literature Review
eTS+ and AnYa have been previously explored for application in similar product yield prediction problems.[1] These techniques represent hybrid fuzzy-neuro systems; eTS+ utilizes scalar parameterized fuzzy sets for antecedents and fuzzy linear consequent combination[2, 3, 7, 29] whilst AnYa uses non-parametric data clusters for antecedents, with it’s only fuzzy component in consequent combination. eTS+ and AnYa have improved upon ANFIS[32], GENFIS2[54] and DENFIS[16] algorithms[6] with online applicability, fewer parameters and greater computational efficiency. A full comparison is detailed in [31]. ANNs require the formation of antecedent rules for the first perceptron layer, the use of clustering serves to reduce the complexity of a NN by generalizing all possible inputs into a fewer number of representative groups.[8]

Clustering algorithms have seen extensive use across all disciplines involving large datasets for classification and prediction.[55] The general purpose of clustering is to form naturally representative groups, which are easily separable and individually compactly bounded. In a fuzzy rule formation, clusters are not required to hold exclusive membership over data and so may exhibit overlaps[18]. For autonomous and adaptive cluster formation in an evolving data stream, the number of clusters should be determined automatically from the underlying data distribution, without the need for operator defined, or problem-specific thresholds. Offline techniques such as K-means[34] and KNN[35], which require specification of k randomized cluster points, are unsuitable for use without frequent retraining in an evolving system. New clusters should form, and old clusters diminished as required by the dynamics of the data.

In Evolving Clustering Method (ECM), a prior belief is placed over the data space indicating the expected data distribution,[17] This expectation is used to prototype cluster locations, where the assignment of a sample to a cluster increases its influence radius up to a threshold maximum. The requirement of prior system knowledge, as well as the definition of a radius threshold makes it unsuitable for autonomous application.

ADD[5] was considered for use as an offline model antecedent training approach, to be used prior to online model updates. As an extension to subtractive clustering[13] and DDC[27], with automated radii calculation, ADD identifies peaks and troughs in the data distribution and forms natural groupings based on the inflexion points of the smoothed data-distribution. ADD starts by first taking the most representative point, and determines a prototype radius for the cluster. By optimizing the radii until clusters no longer exhibit changes in membership, the ‘ideal’ antecedent width is determined. Each subsequently formed cluster is removed from the dataset until few samples remain – which are then assigned to the nearest groups. For each of the clusters, those found to exhibit significant, moderate or slight overlap are combined or split amongst those remaining in order to create well-defined groupings. This method is suitable for datasets that are easily separable, but for those that demonstrate little or no clear separation the algorithm loses descriptive capability.

eClustering+ has been previously utilized in both eTS+ and AnYa to overcome the requirement for a user-specified number of clusters, as found in KNN and K-means, as well as for threshold radius definitions, such as in ECM. Based on subtractive clustering[45], eClustering+ selects highly representative data points as candidates for cluster formation, or else assigns non-representative samples to the cluster with which it is most similar. Samples that are identified as lying outside a 3-sigma norm of any cluster are considered to be outliers, and influence a higher opportunity for cluster formation. eClustering+ extends upon eClustering with a fully adaptive number of clusters, by monitoring the age, support and utility of each. Similarity to[42], clusters that lose representative capability are automatically diminished.

Different similarity metrics may be used with eCluster-
Typicality and Eccentricity analytics (TEDA) were proposed as an alternative statistical framework to frequentist probability theory, and have great suitability for Bayesian inference[4]. TEDA requires no prior assumptions of kernels, thresholds or parameters and builds upon a mutual dependence of data samples to recursively update an eccentricity metric based on sum-of-distance values. In a previous study, TEDA and RDE have been compared and found to be highly comparable for the judgement of data similarity, however due to the low number of training samples available in this study a model-wide frequentist approach to model learning is preferred.

In order to derive the relationship between an antecedent rule’s features and its corresponding output, a regression method is required. In eTS+ and AnYa[4], fuzzily-weighted recursive least squares (fwRLS) has been employed to proportionally derive an antecedent rule’s consequents.[6] Least-Squares forms a linear relationship between inputs and outputs of the form:

$$ Y = \psi A $$

\[ A = [a_1^T, a_2^T, ..., a_R^T]^T \] is a vector of local submodel consequent parameters. \( \psi = [\lambda_1 X^T, ..., \lambda_R X^T]^T \) is a vector of the inputs fuzzily weighted by antecedent firing level. 

$$ X^T = [1, x_k 1, x_k 2, ..., x_k n]^T. $$

Where \( R \) is the number of antecedent rules, and \( n \) is the number of feature dimensions.

For complex system behavior, a linear static model could be considered unsuitable, however the fuzzy combination of several linear models implicitly forms a non-linear relationship.[25] This supersedes the relative computational complexity of non-linear parameter derivation whilst retaining descriptive capability, and facilitates ease of online updates. A Bayesian sampling approach for consequents could be considered if the variables of interest were easy to measure,[36] however due to inherent difficulties in measuring fuel quality there exists very few samples. In addition, the dynamic nature of the operating conditions means that the ground ‘truth’ is constantly changing, and antecedent rules rarely have enough time to gather anywhere near the number of samples that’d be required to converge suitable parameters.

Gaussian Processes were considered as a means of modeling the dynamics of consequent parameters.[44] GPs represent a computationally effective, and accurate means of predicting complex non-linear relationships.[23] However, autonomous methods of selecting suitable kernel-functions are still in their infancy and with few samples available for each consequent rule, models may yet remain inaccurate. In future it may be useful to consider the application of GPs within the application of oil refinery physical property prediction as a means of deriving time-dependent non-linear models, however that is beyond the scope of this report.

Within an oil refinery, there exist a large number of sensors in operation, the majority of which are irrelevant to product quality prediction. In order to greatly simplify the model, whilst maintaining the ability to accurately infernce variables of interest, an effective input variable selection method is required.[12, 48]

Many approaches to input variable selection exist. Largely these fall into, embedded[30] or wrapper model-based approaches[19], and model-free filters.[24]

IVS techniques are often paired with dimensionality reduction methods, such as PCA,[11] ICA[38], LDA or Partial least-squares (PLS) in order to reduce cross-correlation redundancy across many sensors by creating new orthogonal features as a combination of sensors.[20, 26] Though this can be useful to create an efficient model utilizing fewer inputs, the resulting features are often harder to interpret as linear combinations of several sensors. This reduces the transparency of a model, which is a problem in safety-critical operations. Backwards elimination considers at first all available sensors, and judges the value of removing each one in turn. Forward selection works in the reverse, selecting few sensors at first, usually based on some criteria of complexity or accuracy.[43, 15]

Model-free varieties of IVS rely largely on correlation or information theoretic[53] to rank sensors based on their signal-similarity with a feature of interest. Correlation rankings measure only the linear relationship between two features, whilst information theoretic such as Mutual Information[52], Partial Mutual Information[39], Conditional/Joint, and Entropy based rankings understand non-linear relationships. Leave-one-out strategies like LASSO[37, 51] and Recursive Feature Elimination operate within the context of a predictor, by measuring the influence of removing each feature from the model.[21] Should the model improve in accuracy with the removal of a feature, then that variable is candidate for removal.

IV. METHODOLOGY

The overall system is composed of three stages: firstly training data is normalized for input variable selection; Following this, the AnYa ANN is sequentially trained using all available training data; Finally, AnYa model is used for online prediction, with updates made to the model upon
The density of sample $k$ is the Cauchy of the sum of its distance to the current global mean $\mu_k$. Samples with a high density indicate highly representative data points, and are put forward as candidates for cluster initialization should they be more representative than other already existing clusters. Outliers with a low density

2) eClustering+: Antecedents rules are formed when a newly received sample displays a higher density than current clusters, or when a sample is more than 2 standard deviations from its nearest cluster (is an outlier), and has an the lowest density seen with current clusters.

Conditions for Cluster Formation:
1) $\text{density}_k > \max_{j=1}^{N_c}(\text{density}_{y_j})$
2) $\text{density}_k < \min_{j=1}^{N_c}(\text{density}_{y_j})$ \&
\[ \min(d(x_k, x_c)) < \frac{N^2_d}{N_d + N_c} \ast 1\text{stdDev}(d(x_k, \mu_k^j)) \]

If a sample has a higher density than current cluster focal points, and is less than 1 standard deviation from a current cluster, then it replaces that cluster’s focal point.

Conditions for Cluster Replacement:
1) $\text{density}_k > \max_{j=1}^{N_c}(\text{density}_{y_j})$ \&
\[ \min(d(x_k, x_c)) < \frac{N^2_d}{N_d + N_c} \ast 1\text{stdDev}(d(x_k, \mu_k^j)) \]

If a sample satisfies neither criteria for cluster replacement or formation, then it is assigned to the nearest cluster, and antecedent firing levels are calculated as the inverse proportion of distances.

\[ \lambda_k^j = \frac{1 - \sum_{i=1}^{N^d_k} d(x_k, \mu_k^j)}{\sum_{j=1}^{N_c} \sum_{i=1}^{N^d_k} d(x_k, \mu_k^j)} \] (4)

New samples that are not candidates for cluster formation are given proportional membership to nearby clusters on a metric of their relative similarity/distance.

$\lambda_k^j$ represents the firing level of antecedent $j$ for sample $k$ where $N_c$ is the number of clusters and $N_d$ is the number of dimensions.

3) fwRLS: This proportional membership is used as the basis for the fuzzy combination of consequent parameters. Consequent parameters are updated using a fuzzily weighted recursive least squares approach, to regress parameters based on the proportional membership of a sample to each cluster. Should a new sample majority belong to one cluster, then that cluster would undergo the majority of parameter regression.

\[ c_k^j = c_{k-1} - \frac{\lambda_k^j c_{k-1} X_k Y_k X_k^T c_{k-1}}{1 + \lambda_k^j X_k (c_{k-1}^j X_k)} \] (5)
$c_i^j = \Omega I$ where $\Omega$ is a large positive number; $c$ is a $R(n+1)R(n+1)$ local submodel covariance matrix.

$$
\hat{a}_k = \hat{a}_{k-1} + C_k \psi_k y_k - \psi_k^T \hat{a}_{k-1}
$$

$k = [1, R]; \hat{a}_0 = 0; C_0^j = \Omega I; a = [a_0, a_1, \ldots, a_n]^T$. Solutions $a_i$ that minimise the weighted LS problems expressed by the local error can be obtained by applying a weighted pseudo-inversion.[10] $a^k$ is an estimation of the parameters based on $k$ data samples.

4) **Online Monitoring of Antecedents:** Throughout training and later updates, the cluster-based antecedents are monitored in age, utility and support.

The Age is the average sample age.

The Utility is the average firing level since formation.

The Support is the number of members assigned to the cluster.

$$
\text{Support}_{ik} = N_{ik}
$$

$k$ denotes the current time instant; $N_{ik}$ denotes the number of data samples (support) that are associated with the cluster;

$$
\text{Age}_{ik} = k - \frac{\sum I_l}{N_{ik}}; i = [1, R]
$$

$I_l$ denotes the time index of sample $l$.

$$
\text{Utility}_{ik} = \frac{\sum \lambda_i}{k - I_i}
$$

$t_i$ denotes the time index when the $i$th local submodel was created; $\lambda_i$ denotes the antecedent firing level.

These metrics are used to judge the prolonged ‘usefulness’ of a cluster in future predictions, based on its recent representative capabilities. Should a cluster’s average sample age extend significantly, if its average membership from new samples is low, or if it fails to encompass enough samples during its emergence, then it is removed.

$$
\text{Age}_{ik} < \text{mean(Age}_k) + 2\sigma(\text{Age}_k)
$$

$$
\text{Utility}_{ik} < \text{mean(Utility}_k) - 2\sigma(\text{Utility}_k)
$$

$$
\text{Support}_{ik} < 3 \land (k - I_i) < 10
$$

$\text{Age}_{ik}$ represents the Age of cluster $i$ at time $k$. $\sigma(\text{Utility}_k)$ denotes the standard deviation of Utility for clusters at time $k$. With these antecedent termination conditions, the dynamic operating conditions are accounted for. Old, or un-useful clusters are diminished to make way for new more representative groups. This inclusion of evolving rule antecedents can be seen as a forgetful approach to modelling, which is necessary for static consequents in a dynamic process environment.

---

**C. Stage 3a - Online Prediction (Mode I + Mode II)**

**Layer 0** - Anomaly & Fault Detection

**Layer 1** - Feature Select & Online Normalisation

**Layer 3** - Measure Proportional Memberships

**Layer 5** - Predict Output

After training, the system has two modes of operation:

- Mode I is used when only sensor data is provided.
- Mode II is used when lab analysis and sensor data is provided

During prediction, no model updates are made to either the antecedents or consequents. An anomalous sensor will be reported, but a prediction will still be made. Fault detection results in a discarded sample.

1) **Anomaly and Fault Detection:** Any raw signals containing a feature with a higher than 3-sigma deviation are classed as anomalies. Samples classed as anomalies are reported during prediction, but cannot be used in model updates. Sensors that produce a NaN value are logged, the first NaN value is treated as an anomaly. However, if the sensor produces a second error, then it is removed from the input stream.

An extension study should be performed into better strategies for handling missing data; with the large availability of sensor data it would be possible to build an accurate Bayesian model for feature-to-feature inference and IVS.[9] More rudimentary estimates based on cross-correlation could also be considered as a faster approach for rapid real-time responses. Feature-to-feature inference would also allow for the detection of sensor degradation, by modeling the sensor drift a bias update can be applied as a temporary fix and maintenance scheduled.

2) **Online Normalisation:** During online prediction and model updates, online normalization is employed. Should a new sample lie outside the range of previously normalized samples, then the learned maximum and minimum of a feature is updated to ensure normalized samples remain between 0 and 1. By updating the normalization bounds, metrics such as means, consequents, covariance, and accumulated distance are also affected, and gain inaccuracy. A normalization ratio factor can be calculated from the new bounds and applied to feature metrics as a correction.

$$
\eta^1_k = \frac{(\eta^1_{k-1} \ast \max(x_{1:k-1})) - \min(x_{1:k-1}) + \min(x_{1:k})}{\max(x_{1:k})}
$$

$$
\eta^2_k = \frac{(\eta^2_{k-1} \ast \max(y_{1:k-1})) - \min(y_{1:k-1}) + \min(y_{1:k})}{\max(y_{1:k})}
$$

---

5
\[ \mu_{k-1} = \mu_{k-1} \ast \eta_k \]  
(15)

\[ \text{stdDev}_{k-1} = \text{stdDev}_{k-1} \ast \eta_k^2 \]  
(16)

Consequent covariance and parameters have dependency also on the product normalization ratio, and should be updated by the inverse.

\[ c_{k-1}^j = c_{k-1}^j \ast ([1, \eta_k^{\text{ix}1}, ..., \eta_k^{\text{ix}n}]^T \ast [1, \eta_k^{\text{ix}1}, ..., \eta_k^{\text{ix}n}]^T) \]  
(17)

\[ A_{k-1}^j = \frac{A_{k-1}^j \ast [1, \eta_k^{\text{ix}1}, ..., \eta_k^{\text{ix}n}]}{\eta_k} \]  
(18)

\( c^j \) denotes the covariance matrix of the \( j \)th cluster at time index \( k \).

\([1, \eta_k^{\text{ix}1}, ..., \eta_k^{\text{ix}n}]\) denotes a \( 1+n \) length normalisation ratio vector, where \( n \) is the number of features.

D. Stage 3b - Online Model Updates (Mode II only)

Layer 2 Update Antecedents
Layer 3 Measure Proportional Memberships
Layer 3a Online Monitoring of Antecedents Quality
Layer 3b Recursive Feature Elimination
Layer 4 Update Consequents

In ‘Update’ mode, both lab analysis and sensor values are provided. Anomalous and faulty measurements are discarded to avoid model pollution. Antecedent and consequents rules are updated with each new sample. Recursive Variable Elimination utilizes the predicted output to estimate the error introduced per feature.

1) Recursive Variable Elimination: Any subsequent input-output pairs that are introduced to the system first have their outputs predicted, to gauge the accuracy of the model through comparison with the known true output from lab analysis. A take-one-out direct-optimisation methodology akin to Recursive Feature Elimination (LASSO) is used to quantify the relative error introduced per feature.

\[ \hat{y}_j^n = \psi_j^n \hat{A}_j^n \]  
(19)

\( \hat{y}_j^n \) denotes the fuzzy predicted output for the \( j \)th consequent, without the \( n \)th feature.

\( \psi_j^n = [\lambda_1 X_n^T, ..., \lambda_N X_n^T]^T \) is a vector of features weighted by antecedent firing level where the features considered are: \( X_n^T = [1, x_{1:N \notin n}]^T \).

\[ \hat{Y}_k^n = \sum_{j=1}^{Nc} \hat{y}_j^n \]  
(20)

\[ \hat{A}_j^n = a_{0:Nd \notin} \]  
(21)

\( \hat{A}_j^n \) is an \( N_d \) length vector of local submodel consequent parameters without the \( n \)th feature.

Features that have a history of introducing error into predictions become candidates for removal.

\[ E_k^n = \hat{Y}_k^n - Y_k \]  
(22)

The error introduced by feature \( n \) at time index \( k \) is denoted as \( E_k^n \):

\[ \mu_{E_k^n} = \begin{cases} E_n & \text{if } k = 0 \\ \left( \frac{k-1}{k} \right) \mu_{E_{k-1}} + \left( \frac{1}{k} \right) E_k^n & \text{if } k > 1 \end{cases} \]  
(23)

The mean per feature error is used calculated recursively, by taking a fraction proportional to the number of previous samples.

If the mean error introduced by a feature is higher than the mean prediction error, divided by one less than the number of features \( N_d \), then it is removed.

\[ \mu_{E_k^n} > \frac{\mu_{E_k}}{N_d - 1} \]  
(24)

\( \mu_{E_k} \) represents the mean prediction error at time index \( k \), whilst \( \mu_{E_k^n} \) denotes the mean feature error.

V. RESULTS AND DISCUSSION

The developed fuzzy-ANN has been successful in forming a data-driven approach to estimating physical properties of interest. The solution provides real-time prediction and update capability, whilst retaining long-term adaptability by updating the number of antecedents at each sample introduction. Varying levels of autonomy in initial variable selection can be utilized in cases where expert knowledge is not available. Currently, Recursive Input Variable Elimination may only remove features from the data stream, so model retraining is required when sensor configurations change.

- Demonstrate the successful use of data-driven models to estimate physical properties from refinery operating data.
- Investigate the capability for model adaptability across long-term data streams.
- Explore applicability of the system and derived models across multiple refineries.

Prediction of the product from an I/O data pair allows for the prediction accuracy to be quantified prior to consequent
updates from lab analysis data. Metrics of interest also include: number of features; number of antecedents; running time; and the model parameter dynamics.

What follows is an analysis and discussion into each component of the developed system.

A. **MI Input Variable Selection**

Mutual Information Input Variable Selection is analyzed for performance by first inspecting the relative prediction accuracy of models trained on autonomously selected features, versus a manual expert selection, and a combination of both sets.

The Mutual Information plot shows the ranked mutual information values for each variable, as well as the cut-off threshold for variable selection. The first negative inflexion of the smoothed mutual information plot is selected as a termination point.

This minimum criteria threshold should be later explored for autonomous optimisation, but it’s difficult to infer the true utility of a feature from mutual information alone. An extension considering conditional mutual information, where the candidate’s mutual information is considered given the mutual information between other features and the output $M_{IL}(Y|X_c, X_{1:n})$ would be useful in finding truly useful features, though the complexity of the search increases from $O(n)$ to $O(n^2)$.

The prediction accuracy is measured without the use of Recursive Variable Elimination and compared with the results from a manually selected set of sensors considered to be relevant. Each of the hand-picked sensors are searched for in the data stream and used if found. A merged ‘partial’ set of MI-IVS and manually selected sensors is also considered.

MI-IVS competes well with hand-picked features, improving on 3/4 products, but increases model complexity substantially. During full-auto, or partial-auto mode, recursive variable elimination should be utilized.

B. **Online Normalisation**

It’s important that both offline and online normalization techniques give the same results in all parts of the model. Antecedent and Consequent variable updates are performed in both parts. Metrics of interest also include: number of features; number of antecedents; running time; and the model parameter dynamics.

**TABLE I**

<table>
<thead>
<tr>
<th>Product</th>
<th>IVS Method</th>
<th>$N_d$</th>
<th>RMS Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS4</td>
<td>MI</td>
<td>10</td>
<td>2.5705</td>
<td>0.0570</td>
</tr>
<tr>
<td>SS4</td>
<td>Manual</td>
<td>3</td>
<td>3.7098</td>
<td>0.0779</td>
</tr>
<tr>
<td>SS4</td>
<td>Mixed</td>
<td>13</td>
<td>2.3680</td>
<td>0.0518</td>
</tr>
<tr>
<td>SS3</td>
<td>MI</td>
<td>9</td>
<td>3.9536</td>
<td>0.0864</td>
</tr>
<tr>
<td>SS3</td>
<td>Manual</td>
<td>3</td>
<td>4.7661</td>
<td>0.1061</td>
</tr>
<tr>
<td>SS3</td>
<td>Mixed</td>
<td>12</td>
<td>9.8531</td>
<td>0.1669</td>
</tr>
<tr>
<td>SS2</td>
<td>MI</td>
<td>13</td>
<td>12.0134</td>
<td>0.1416</td>
</tr>
<tr>
<td>SS2</td>
<td>Manual</td>
<td>3</td>
<td>6.8880</td>
<td>0.0860</td>
</tr>
<tr>
<td>SS2</td>
<td>Mixed</td>
<td>16</td>
<td>10.8299</td>
<td>0.1329</td>
</tr>
<tr>
<td>SS1</td>
<td>MI</td>
<td>7</td>
<td>7.2980</td>
<td>0.0917</td>
</tr>
<tr>
<td>SS1</td>
<td>Manual</td>
<td>5</td>
<td>8.7766</td>
<td>0.1298</td>
</tr>
<tr>
<td>SS1</td>
<td>Mixed</td>
<td>12</td>
<td>5.8316</td>
<td>0.0835</td>
</tr>
</tbody>
</table>

**Fig. 1.** SS4 Mutual Information, Ranked Input Variable Selection

**Fig. 2.** SS1 Mutual Information, Ranked Input Variable Selection

**Fig. 3.** Mutual Information SS4 Signal Response and Prediction, no RIVE. Mean % Error: 0.0570 RMS Error: 2.5705

Mean % Error: 0.0570 RMS Error: 2.5705
upon receipt of a sample outside the previously observed range.

The figures show that firstly the batch normalisation technique retains the information from the raw data effectively, and then secondly that the online normalisation technique accurately imitates the offline normalisation, with use of the normalisation ratio correction as detailed earlier in this report.

C. RDE

Cluster formation is determined partly by the density value of a data sample. There are two criteria for cluster formation:

1) A sample has a higher density than all current clusters.
2) A sample is more than 2-sigma from all current clusters, and has a lower density than all current clusters.

The distance between each sample and cluster is calculated as the sum Euclidean 1-norm distance for each feature.

The standard deviation of inputs from the mean is updated recursively and used to quantify outliers, for which holding the lowest density of existing clusters is accepted. Multiplying the mean standard deviation by the number of features is equivalent to a 1-sigma Euclidean distance. As new clusters are added, a larger proportion of the data-space is ‘covered’ by a cluster’s influence. A 2-sigma distance in \( N_d \)-dimensional space for \( N_c \) clusters is calculated as \( \frac{N_d}{N_d+N_c}2\sigma \). This is used as the outlier classification threshold.

The density of each cluster is recalculated at each iteration, allowing samples with a density currently higher than a cluster’s to form a new group, or replace an existing.
D. eClustering

If a sample is put forward as a candidate for cluster formation, then eClustering decides what to do with it:

- New clusters are formed if a candidate sample is more than 1-sigma from current clusters.
- A cluster is replaced if it is closest to the sample, and less than 1-sigma away.

Any new or replacement cluster sample is allocated an antecedent firing level of 1 (100% membership), to receive full training from the consequent parameter regression. Any sample that initiates a new antecedent has its consequent parameters initialized as the fuzzily combined parameter regressed during the previous update.

E. Online Monitoring of Antecedents

There are three conditions that satisfy requirement for antecedent removal:

- Age is higher than $\mu_{\text{Age}} + 2\sigma_{\text{Age}}$
- Utility is lower than $\mu_{\text{Utility}} - 2\sigma_{\text{Utility}}$
- Support is lower than 3 after 10 new samples.

$\sigma$ is calculated as the standard deviation.

A large number of clusters formed are diminished because they do not reach the criteria for number of supporting samples received early in their lifetime. Age and Utility have thresholds formed from a combination of the mean and standard deviation, giving a parameter-free approach. The thresholds for cluster removal may otherwise be optimised, or catered toward other applications.
F. Recursive Input Variable Elimination

RIVE estimates the error contributed by each variable, by predicting the output without the consideration of one feature. The relationship between the termination criteria and the per-feature error is shown below.

If a feature contributes mean positive error higher than \( \frac{1}{N_d - \mu E} \), then it is removed.

Table II: MI-IVS with RIVE Comparison for all products, non-RIVE results in brackets.

<table>
<thead>
<tr>
<th>Product</th>
<th>IVS Method</th>
<th>( N^RIVE_d )</th>
<th>RMS Error</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS4</td>
<td>MI</td>
<td>9 (10)</td>
<td>2.7561 (2.5705)</td>
<td>0.0609 (0.0570)</td>
</tr>
<tr>
<td>SS4</td>
<td>Mixed</td>
<td>9 (13)</td>
<td>2.6972 (2.5680)</td>
<td>0.0560 (0.0518)</td>
</tr>
<tr>
<td>SS3</td>
<td>MI</td>
<td>(8) 9</td>
<td>3.9938 (3.9536)</td>
<td>0.0862 (0.0864)</td>
</tr>
<tr>
<td>SS3</td>
<td>Mixed</td>
<td>6 (12)</td>
<td>7.2632 9.8531</td>
<td>0.1137 (0.1669)</td>
</tr>
<tr>
<td>SS2</td>
<td>MI</td>
<td>6 (13)</td>
<td>10.8378 (12.0134)</td>
<td>0.1263 (0.1416)</td>
</tr>
<tr>
<td>SS2</td>
<td>Mixed</td>
<td>10 (16)</td>
<td>14.9216 (10.8299)</td>
<td>0.1607 (0.1329)</td>
</tr>
<tr>
<td>SS1</td>
<td>MI</td>
<td>4 (7)</td>
<td>6.8656 (7.2980)</td>
<td>0.0759 (0.0917)</td>
</tr>
<tr>
<td>SS1</td>
<td>Mixed</td>
<td>6 (12)</td>
<td>5.1031 (5.8316)</td>
<td>0.0730 (0.0835)</td>
</tr>
</tbody>
</table>

The termination criteria is a proportion of the mean prediction error, and can be seen falling, and will stabilise over time. It may be useful to consider a sliding window method for a fully adaptive recursive variable selection method for dynamic sensor utility.

Features that exhibit a negative per-feature error contribute accuracy to predictions. It was important not to simply remove features that have a positive error contribution, as cooperative feature contribution is not considered, removing a lowly error may introduce precision error, or remove important uniquely held information of the product from the feature stream.

The table shows mixed results, with SS1 and SS3 benefiting from RIVE, SS4 losing accuracy, and SS2 giving mixed results. It is unclear why these results are seen, though it may be due to the relative mutual information contained in features that are removed. SS2 was seen to perform better with manual selection, and fewer features, whilst other products did not, the removal of features from it’s MI-candidate set increased accuracy, showing that many may have been redundant. Meanwhile, feature elimination in the Mixed set was detrimental. This may have been due to an essential feature being wrongly removed, due to a lack of full feature information. Improving the metrics of feature utility could be considered for further research.

1) Running Time: The system’s running time has been calculated on a dual 3.2GHz processor with 8GB RAM and a 1000MHz//1500MB GPU.

- 0.0029s for Stage 1 - MI IVS over 105 features for 64 samples
- 0.0018s for Stage 2 - Time taken for model training per sample.
- 0.0007s for Stage 3a - Time taken to predict a property measured per sample.
• 0.0029s for Stage 3b - Time taken to predict and update model, per sample.

The speed of this system makes it ideal for online prediction and model updates in real-time applications.

2) Transparency: The number of antecedents rises with model complexity, but doesn’t exceed 12 clusters at any instant. A combination of up to 12 consequent parameters can be difficult to visualise, however a single fuzzily combined parameter signal can be better understood by an operator.

3) Autonomy: The system has no user-defined thresholds or parameters, though tuning of MI-IVS allows for a higher number of candidate features to be suggested. The system eliminates non-useful features, and also those that have given faulty signals. If a sensor is wrongly removed from the features of interest, then an operator should intervene. In addition, if new sensors are added to a plant after the model is built, then the model will require retraining. An investigation should be made into effective methods to reintroduce, or to introduce new features into an already trained model.

VI. CONCLUSION

The final system has produced accurate predictions that improve on currently used techniques, and successfully meet objectives to demonstrate the utility of adaptive data-driven models.

MI-IVS and RIVE have shown to be effective for feature selection and removal, producing valuable predictions and selecting few useful sensors from a large number of candidates. In cases, particularly versus MI-IVS without RIVE, a manual selection of features was beneficial. It is important however that if expert knowledge is held, then it should be utilised. Improvements can be made to MI-IVS and RIVE to maximise the product information explained by the features selected.

Techniques developed focussed largely on universal-parameter and parameter-free approaches to autonomy, which generalise well and require no tuning. This is suitable for unknown, dynamic and complex systems, further investigation should be made however, into alternative data-driven metrics and thresholds particularly for variable selection and antecedents monitoring.

For the application of O&G product property prediction, this system would be greatly improved by time-dependency in the model. The utilisation of information from previous time steps and a derived understanding of the process dynamics helps a system to understand where the process has come from, and where it is leading to. This could also be used to predict feature utility and the evolution of model parameters.

TABLE III

<table>
<thead>
<tr>
<th>Product</th>
<th>IVS Method</th>
<th>(N_{\text{new}})</th>
<th>(N_{\text{of}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS4</td>
<td>MI</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>SS4</td>
<td>Mixed</td>
<td>9</td>
<td>18</td>
</tr>
<tr>
<td>SS4</td>
<td>Manual</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>SS3</td>
<td>MI</td>
<td>12</td>
<td>27</td>
</tr>
<tr>
<td>SS3</td>
<td>Mixed</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>SS2</td>
<td>MI</td>
<td>11</td>
<td>27</td>
</tr>
<tr>
<td>SS2</td>
<td>Mixed</td>
<td>9</td>
<td>23</td>
</tr>
<tr>
<td>SS1</td>
<td>Manual</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>SS1</td>
<td>MI</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td>SS1</td>
<td>Mixed</td>
<td>10</td>
<td>23</td>
</tr>
<tr>
<td>SS1</td>
<td>Manual</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

MAX AND TOTAL NUMBER OF CLUSTERS PER PRODUCT AND MI METHOD

REFERENCES


