

Improved Anomaly Detection using Integrated Supervised and Unsupervised Processing

Bobby Hunt, David G. Sheppard, Charles J. Wetterer
Integrity Applications Incorporated - Pacific Defense Solutions

CONFERENCE PAPER

There are two broad technologies of signal processing applicable to space object feature identification using non-resolved imagery: supervised processing analyzes a large set of data for common characteristics that can be then used to identify, transform, and extract information from new data taken of the same given class (e.g. support vector machine); unsupervised processing utilizes detailed physics-based models that generate comparison data that can then be used to estimate parameters presumed to be governed by the same models (e.g. estimation filters). Both processes have been used in non-resolved space object identification and yield similar results yet arrived at using vastly different processes. The goal of integrating the results of the two is to seek to achieve an even greater performance by building on the process diversity. Specifically, both supervised processing and unsupervised processing will jointly operate on the analysis of brightness (radiometric flux intensity) measurements reflected by space objects and observed by a ground station to determine whether a particular day conforms to a nominal operating mode (as determined from a training set) or exhibits anomalous behavior where a particular parameter (e.g. attitude, solar panel articulation angle) has changed in some way. It is demonstrated in a variety of different scenarios that the integrated process achieves a greater performance than each of the separate processes alone.

1. INTRODUCTION

Because a wide variety of US military assets and systems make use of information gleaned from space-borne sensors and communications, Space Situational Awareness (SSA) has become a growing concern in the planning, deployment and execution of the strategy and tactics of US forces. The technological means to achieve SSA are in the suite of modern sensors, e.g., optical telescopes operating in a variety of wavelengths and passive and active ranging systems, such as microwave-regime radar. These sensors are confronted by a variety of different operational constraints, because the regions of space where SSA concerns exist span distances and orbital velocities from Low-Earth-Orbit (LEO, several hundred kilometers) to Medium-Earth-Orbit (MEO, a few thousands of kilometers) to Geosynchronous-Earth-Orbit (GEO, tens of thousands of kilometers). The variety is so great that no one sensor technology is suited to all possible SSA concerns.

An ideal SSA technology would provide detailed imagery in all wavelengths, enabling the complete interpretive analysis that has been highly developed for other military threats. This is not possible in a number of critical SSA venues, such as GEO, or observations from small telescopes. This is often referred to as the problem of Non-Imaging Space Object Identifications (NISOI).

When confronted by NISOI the only option is to record not detailed imagery but simple time histories of the light flux reflected from a space object. The challenge to SSA in this case is to determine useful information about the particular space object from analysis of the "light curve" (the time history of the reflected light flux). Advancing the state of knowledge and system capabilities for NISOI situations is the subject of the research that is reported herein.

The time history of reflected light from a space object is a specific type of signal. Signal processing during the past 70 years has taken two distinct and separate directions. One direction is Supervised Processing (SP), which is often referred to by the alternate descriptions of pattern recognition or pattern classification [1]. SP uses different methods to extract the essential information that is the minimum required for assigning any specific instances of a signal into different generic classes. The other distinct mode of signal processing is Unsupervised Processing (UP), which rely on sequential Bayes algorithms to estimate parameters of the signal and the underlying system that generated the signal [2].

2. SUPERVISED PROCESSING

In the case of SSA for NISOI neither SP nor UP has the complete answer to questions of interest about space object light curves. Combining the results from these two different paradigms is the purpose of the research reported herein. Integrated Supervised and Unsupervised Processing (ISUP), as described in this paper, is a system and a

methodology for fusing the results of SP and UP into a more complete picture of a space object from the light curves that can be collected from the object [3].

The history of the mathematics, science and engineering of signal processing during the past 70 years can be divided into two distinct categories, each with a rationale and organization that motivates the internal methods and procedures that implement the two different methods.

Supervised Processing (SP) of signals proceeds from the assumption that the signals collected from a specific object or system have some aspects of the structure or properties of the system preserved in the information of the signal. It is a basic premise, therefore, that when signals are collected from the same object, with the object behaving in the same conditions with the same parameter settings, that there are some essential elements of information in the signal that can be used to reliably assign the signals into a class of “baseline” behavior for the object or system. These essential elements of information in the signal are called “features”. If the object or system changes, then there is an equally basic premise that these features or values of the features will change accordingly, and it will then be possible to assign the signals into a different class of behavior for the object or system. The convention associated with SP is to refer to the assignment of signals into different classes as “pattern recognition” or “pattern classification” [1, 4-5].

SP derives the appellation of “supervised” from the methods used to develop the rules and procedures by which a signal can be assigned to a specific class. A volume of signal data is collected. The volume of signal data is assumed to be rich with specific instances of the different classes that are known to be representative of the system, when acting in the different behavior conditions and parameters that require classification. This volume of signal data is known as the “training data” and the names for the classes, into which signal groupings are assigned, are the “training labels”. During training an analysis algorithm is employed to adjust and set the parameters of a decision-making procedure that has the effect of assigning a single instance of a signal to a specific class. This procedure of training is called “supervised” because the training data and training labels “supervise” the algorithm that determines how any signal is assigned to a class. Insofar as a class represents knowledge or semantic content for different object or system behaviors, the training process inherently carries this semantic content into the algorithm that makes a classification.

The SP portion of ISUP utilizes a selection of classifiers based on Kernel Support Vector Machines (SVM) for two-class classification, and Kernel Support Vector Data Description (KSVD) for one-class classification. This selection of classifiers functions as an ensemble, feeding output to the Integration Engine. The strength of ensemble processing is in the diversity of its members, each of which may be fairly weak in terms of classification performance, but contribute something unique to the overall solution. For SP processing in ISUP, diversity comes from choosing different approaches to the Kernel SVM optimization problem, and different kernel types and sizes. This results in the demarcation of different class boundaries produced by each classifier from its selected support vectors. Diversity is also added to the Kernel SVDD classifiers by including one based on the exponential kernel, which has “heavier tails” than the Radial Basis Function (RBF) kernel. This structure is shown in Figure 1. A summary of the key algorithmic characteristics is found in Table 1.

When samples from the nominal and anomalous populations of light curves are available, SP processing includes two different implementations of Kernel SVM for two-class classification: Least-Squares SVM and Kernel SVM from the Matlab Statistics and Machine Learning Toolbox. More information on the SVM optimization problem can be found in [6]. The primary reference for LS-SVM is [7]. The LS-SVMlab Toolbox is described in [8], and is available under the Gnu Version 2 license at [9].

Kernel SVDD is an extension of Kernel SVM to the one-class problem. Further details on Kernel SVDD can be found in [10] and [11]. Further details on the KSVD function may be found on the DD_tools help page [12].

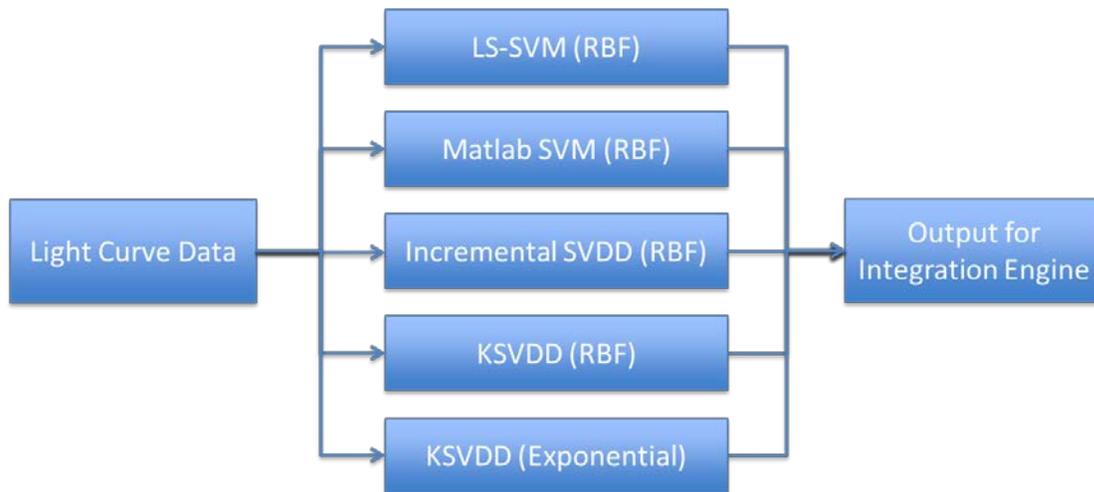


Figure 1. SP Classifiers

Table 1. Kernel SVM And SVDD Algorithm Summary

Algorithm	Kernel	Optimization	Source
Matlab SVM	RBF	Kernel scale by heuristic procedure; SVM training by Sequential Minimal Optimization;	MATLAB Statistics & Machine Learning Toolbox
LS-SVM	RBF	Kernel scale by Coupled Simulated Annealing & Simplex Method; SVM training by solution of a linear system of equations	LS-SVMlab 1.8 Open Source @ Cat. Univ. Leuven
Inc. SVDD	RBF	Kernel scale by method of artificial outliers; SVM training by online recursive algorithm	DD Tools Open Source @ Tech. Univ. Delft
KSVDD	RBF, Exponential	Kernel scale by method of artificial outliers; SVM training by Quadratic Programming (from Matlab Optimization Toolbox)	DD Tools Open Source @ Tech. Univ. Delft

3. UNSUPERVISED PROCESSING

Unsupervised Processing (UP) of signals proceeds from the assumption that the signals from a specific system may be too rich in structure or variations to allow simple class designations. Instead of knowledge of a set of classes of behavior there is, instead, knowledge of the structure of the system that actually generates the signals. This knowledge is embodied in some type of model, e.g., a mathematical description of the physics and interactions that create the signals that are observed from the system. This system, however, is not always known completely. There may be internal parameters or different aspects of the system's interactions with the surrounding environment that cannot be determined *a-priori*. In this case it is desired to use the observed signals to estimate values for the unknown parameters that determine the actual observed signals. A further complication is that the environment, in which signals from the system are observed, is typically not under the control of the observer. Thus, any structural or methodical advantages that could be derived from inserting system inputs and observing the corresponding

outputs are not possible. Lacking this systematic observation, the recourse is to assume that the system behavior is random and unpredictable, and to employ the methods of probability and statistics to estimate the relevant system parameters.

One of the most successful techniques for statistical estimation of system parameters is sequential Bayesian analysis. The unpredictable behavior of signals is assumed to be representable with a specific set of probability laws, e.g., Gaussian random processes. The observed signals are then the result of random inputs to the system model. Successive values of system outputs are determined from how the system model, with the unknown parameters, transforms the inputs into the observed outputs. As more signal output values are observed more information is obtained that is used to update initial parameter estimates and the associated posterior probabilities. With sufficiently enough data, for a system that satisfies certain constraints, the estimates will converge and produce stable representations of the unknown parameters.

The Kalman filter and its various derivatives are the most widely employed of sequential Bayesian methods [2]. This is called an “unsupervised” processing method because there are no volumes of training data and no predetermined classes known that can be used in the supervisory sense. The Unscented Kalman Filter (UKF) [13-15] is an estimation filter and can be used by itself and for individual hypotheses in a multiple hypothesis testing (MHT) analysis. Similarly, the Unscented Schmidt Kalman Filter (USKF) [16, 17] is a version of this where the uncertainty in “consider” terms can be included. In this paper, Multiple Hypothesis Testing (MHT) is used and refers to running simultaneous UKFs or USKFs in parallel and using the observations to assign weights to each hypothesis. Generally, this architecture is that of the Unscented Particle Filter (UPF) first proposed by van der Merwe et al. [18] and a subset of the Local Linearization Particle Filter (LLPF) [19]. As such, “particles” and “hypotheses” represent the same thing. MHT using the UKF or USKF as the engine is used in various estimation schemes, from initial orbit determination, as with the Constrained Admissible Region Multiple Hypothesis Filter (CAR-MHF), [20, 21] to a more refined simultaneous orbit, attitude and shape determination, as with Multiple Model Adaptive Estimation (MMAE). [22, 23]. The framework is in Figure 2.

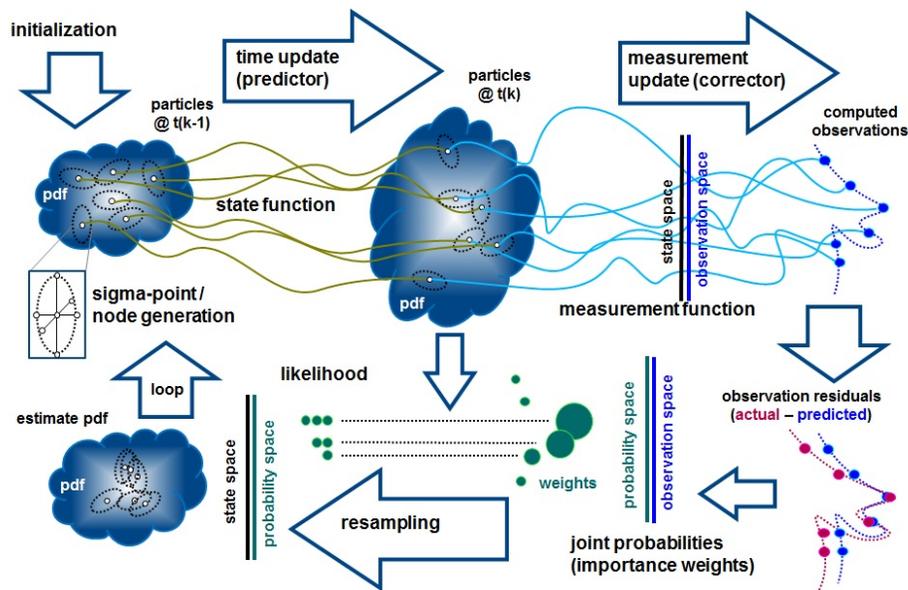


Figure 2. Generic MHT/UPF

The state function, measurement function, time update (predictor), measurement update (corrector), and sigma-point/node generation function are all detailed in the UKF and USKF references [13-17]. The initialization function and calculation of joint probabilities (importance weights) are discussed here.

The initialization function is used to create the individual hypotheses. Each hypothesis initiates a separate UKF or USKF. The initialization function is thus problem specific and does not take a single form. For example, CAR-MHF uses the constrained admissible region to specify acceptable initial conditions and the hypotheses are then

distributed in this region. Alternatively, MMAE uses a bank of possible shape models and the hypotheses are simply each model.

The joint probabilities (importance weights) are calculated by comparing the predicted measurements with the actual observations. The importance weight for the i -th hypothesis and k -th time step is calculated using

$$p_k^i = (2\pi)^{-m/2} |P_k^{uv,i}|^{-1/2} e^{-\frac{1}{2}(\hat{y}_k^i - \tilde{y}_k)(P_k^{uv,i})^{-1}(\hat{y}_k^i - \tilde{y}_k)} \quad (1)$$

where \tilde{y}_k is the actual observation vector, \hat{y}_k^i is the predicted measurement vector, $P_k^{uv,i}$ is the innovation covariance and m is the length of the measurement vector. The likelihood function is then

$$w_k^i = \frac{w_{k-1}^i p_k^i}{\sum_{i=1}^N w_{k-1}^i p_k^i} \quad (2)$$

where w_{k-1}^i is the hypothesis's weight from the previous step, and N is the total number of hypotheses.

A number of different resampling functions exist [24], such as Sequential Importance Resampling (SIR) and Markov Chain Monte Carlo (MCMC). In the analyses done here, however, no resampling is done and so the details of the various algorithms are not presented.

4. INTEGRATION ENGINE

The two different processing modes of SP and UP are fused into a final decision. The actual mechanism of fusing the different outputs of the SP and UP components could be another committee machine or an ensemble. There is little theoretical or empirical evidence to guide the actual choices that can be made among all the different possible combinations of classifier methods, ensemble methods and committee machine methods. Again, the guiding principle was to construct a working procedure that was simple, easy to train and interpret in the process of developing the integration of SP and UP components.

The actual committee machine employed for fusing SP and UP was an ensemble that was trained on the outputs of the five SP classifiers combined with specific variables estimated from the UP components. This is an example of the merger of different variables by an ensemble that serves as a committee machine, demonstrating the flexibility of ways in which different methods of processing the light curve data can be combined to make a processing architecture that is neither fish nor fowl, but a pragmatic structure of maximum utility.

Figure 3 is the basic structure of how the SP and UP processing is implemented and the results of both aggregated together for a final result. This is the Integration Engine (IE).

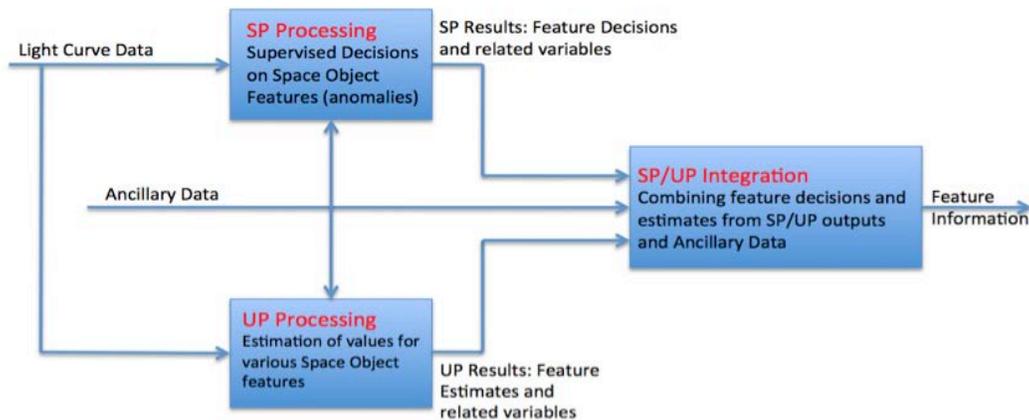


Figure 3. The Structure of The Integration Engine

The basic input to all processing is of two types: light curve data and ancillary data. Ancillary data is any form of information that may prove useful in the fusion of the light curve data, e.g., date of observation, time of day, type of orbit, etc.¹

The light curve data is subjected to the classifier processing in SP, the upper center block in the diagram of Figure 7. A total of five different classifiers were used, as presented previously. These five classifier outputs are combined in an ensemble classifier that constructs many weak learners with the method of Classification and Regression Trees (CART). The result of the ensemble classification is to reach a decision: does the light curve conform to a baseline that is considered “normal”, or has there been a change (the nature of the change is not known) such that the light curve is now considered to be an “anomaly”? Thus, the ensemble operations of SP are, in simplest description, a committee machine that creates the SP decision as to “normal” or “anomaly”.

In parallel with, and independent of, SP there is UP operating on the same data, shown in the lower center block in the diagram of Figure 7. This UP processing is a variety of Kalman filter that estimates the values of various state variables that are determinable from the light curve data. The estimates and the operations of the filter are on the basis of a constructive geometry model of the space object and the interactions of solar illumination on the model.

The final operation is integration of SP and UP. This is the Integration Engine (IE) in the block at the right center of the diagram in Figure 7. This is another ensemble classifier. IE is trained with data from the classifier outputs of SP and the estimated variable produced in UP. The ensemble operation of IE is, thus, a committee machine that makes the final decision that a light curve is “normal” or “anomaly”. The “normal” or “anomaly” status of a light curve is the feature of interest that is produced by this architecture. It is not the only possible such feature, of course. The decision that a light curve is anomalous can trigger a process to send the UP estimates of various variables to a set of further feature decisions or to enter those values into a database of features as the latest values of the space object.

The basic input to IE is of two types: outputs from SP and outputs from UP. The inclusion of ancillary data is also possible, but is not currently a part of the baseline IE.

The light curve data was subjected to the classifier processing in SP, where a total of five different classifiers were used, as presented previously in the section on SP. The output of each SP classifier is a numerical result from computations with its support vectors. All classifiers were trained on the same data set, but the diversity of techniques involved results in a different set of support vectors for each classifier, as well as different weights associated with them. Additionally, the kernel scale can vary between them. The result is that each classifier potentially learns to generalize from the training data set differently. This diversity is very important to IE processing, and allows IE to “sift” through these differences and learn how to improve on their performance.

In parallel with, and independent of, SP there is UP operating on the same data, shown in the lower center block in the diagram of Figure 3. This UP processing was a variety of Kalman filter that estimates the values of various state variables that are determinable from the light curve data. The estimates and the operations of the filter are on the basis of a constructive geometry model of the space object and the interactions of solar illumination on the model.

Several types of ancillary data were examined for usage by IE, but several experiments confirmed that the effect on performance was minimal.

The IE fusion approach is based on the ensemble classification concepts of Bagging and Classification and Regression Trees (CART). The ensemble consists of many decision tree classifiers, each trained on a different subset of the training data (called a “bagged” data set).

The baseline implementation of IE has the following steps, seen in overview in Figure 4.

¹ Ancillary data in the form of date, time, solar phase angle and orbital elements were tested in development, but none provided any gain in performance over the basic light curve data itself. This does not justify a conclusion, however, that ancillary data will not prove useful in other instances beyond this research.

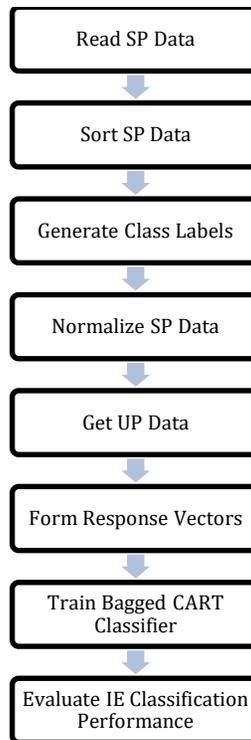


Figure 4. IE Processing Steps

1. Read the Data. IE does not deal directly with the light curve data. Instead, the actual outputs of the SP and UP processing are extracted from the data files. These are the actual values used in the IE processing.
2. Sort the SVDD classifier results. This ensures that the ordering is consistent with the ordering of the LS-SVM and ML-SVM classifier results. The data simulated and originally used in SP had a different ordering of light curves, so a re-sort is necessary for all light curves to have the same truth labeling. This is done by finding the light curves in the ML-SVM data that match the corresponding light curves in the data for the three SVDD-based classifiers.
3. Get the actual classifier outputs and store them for the construction of all the classifier records for IE training. At this point, there is a complete set of outputs for all five of the classifiers.
4. Add correct labels to the extracted SP. Assign the correct labels to each of the SP classifier outputs. The correct labels are as obtained from the ML-SVM classifier, which are also the labels in the LS-SVM classifier. The "true" or "normal" label is +1, and the "false" or "anomaly" label is -1. The same sorting process as above is repeated for the extraction and assignment of labels.
5. Normalization. After assigning labels, there is a normalization process to give a comparable range of data into IE. The normalization is with respect to the maximum above classifier threshold and the minimum below it. All normalized classifier outputs then fall between -1 to +1. This allows a comparable range of values for the computations of IE
6. Get the UP data. Extract the values of Kalman Filter states and variables that are produced by UP and available to use in IE processing. The variable "Weights" is used. Previous development and testing demonstrated that it contains the most useful information from UP for the IE output.
7. Form the Reponse Vectors. The SP and UP output generated from the "operations" data are combined to construct the classification response vectors by combining the output of the five different SP classifier results with the UP variable "Wt."
8. Train IE. Multiple learners are used in CART bagging. Cross-Validation is used with multiple-fold training. Train a tree classifier with bagging and cross-validation. Note that cross-validation is most useful in the cases where there are not a sufficient number of test samples that are not used in training, and is the standard method to convincingly validate the training results for accuracy of classification. This produces data to compare against straight bagging results. The Bagged CART classifier is used for the actual IE processing.

- Evaluate IE. The bagged CART classifier is evaluated with a separate set of "evaluation" data in order to determine the overall generalization capability. IE is applied to the "evaluation" data response vectors to produce the final outputs. The bagging error and the ROC curves are calculated by averaging the performance over all of the bagged subsets of the data.

All the scenarios to be considered are classification problems with only two classes, nominal and anomalous, where anomalous was taken to be a positive classification. As such, a given component (SP, UP, and IE) specifies a prediction, and this prediction can be compared to truth to construct a confusion matrix (alternatively named contingency table). A sample confusion matrix is shown in Figure 5.

	Anomalous (P)	Nominal (N)
Anomalous	True positive (TP)	False positive (FP) (Type I error)
Nominal	False negative (FN) (Type II error)	True negative (TN)

Figure 5. Confusion Matrix

The columns represent truth while the rows represent the prediction, and so P and N in the confusion matrix represent the total truth values. The performance measure to be used in this study is the Receiver Operating Characteristics (ROC) curve and the associated Area Under the Curve (AUC) [25]. A ROC curve plots the true positive rate (TPR) versus the false positive rate (FPR). A single point on the ROC graph would represent a single discrete classifier. In this study, however, the classifier allows for a continuous variation of the boundary between "nominal" and "anomalous", and thus a continuous curve for all FPR values between zero and one is produced. A sample ROC curve with an AUC = 0.774 is shown in Figure 6.

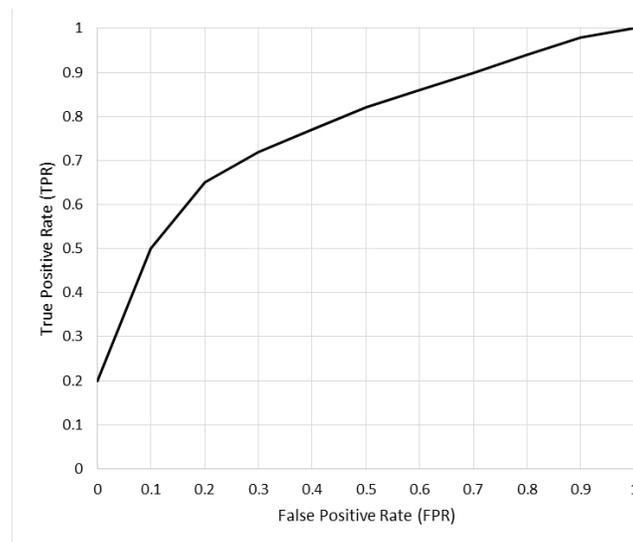


Figure 6. Sample ROC curve

A ROC curve along the lower-left to upper-right diagonal of Figure 6 yields an AUC = 0.5, which essentially indicates the classifier is no better than a coin toss. AUC values less than 0.5 indicate the classifier is worse than a coin toss!

5. SCENARIO DESCRIPTION

The scenario was focused on satellites in GEO. It used simulated data of Galaxy 15 with the shape model shown in Figure 7. Note that the solar panel of Galaxy 15 was not flat but divided into two separate components. The “deployment” angle that defines the magnitude of the bend was fixed at 10 degrees. The Cook-Torrance BRDF [26, 27] was used to describe each surface with the pertinent parameters listed in Table 2. These BRDF values provide brightness measurements comparable to actual small telescope observations.

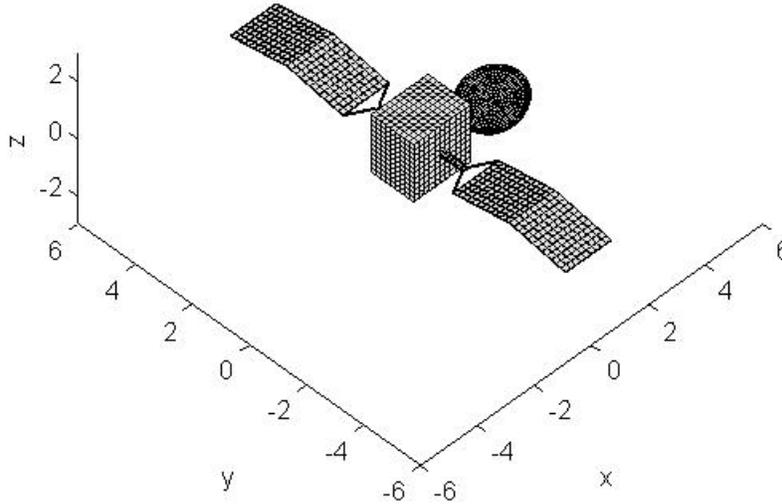


Figure 7. Galaxy 15 shape model

Table 2. Surface Model BRDF Parameters

	Bus (and Dish for Galaxy 15)	Solar Panel(s)
m – microfacet slope	0.10	0.05
ρ_B - reflectance	0.90	0.70
d_B - diffuse fraction	0.70	0.20
ρ_V - reflectance	0.85	0.65
d_V - diffuse fraction	0.68	0.17

For Galaxy 15, the satellite bus z-axis points to a particular spot on the surface of the Earth, the x-axis points towards the velocity vector, and the solar panels track the Sun by rotating about the y-axis. In the nominal case, there are specific east-west offsets to the solar panels. Two different “anomalous” modes of operation were also simulated. In each, the solar panel east-west offsets are slightly different. Two cases were simulated and analyzed, an “original” configuration and a “harder” configuration. Solar panel east-west offsets for these two cases are shown in Table 3.

Table 3. Galaxy 15 Model Solar Panel East-West Articulation Angles (Degrees)

Solar panel east-west articulation angles (degrees)	Original (θ_1, θ_2)	Harder (θ_1, θ_2)
nominal	+3, -8	+6, +1
anomalous #1	-1, -10	+7, +1.5
anomalous #2	+4, -8.5	+5.5, +1.5

Simulated observations were generated for an observer in Maui in two wavebands (Johnson-B and Johnson-V). The two-line elements (TLEs) for the satellite SSN 28884 (Galaxy 15) were used to calculate the orbit. The training period (used to train SP) runs for every other day during 2012 (183 days), while the operations period (analyzed by both SP and UP and used to train IE) runs during 2013 and includes a sampling of days, but only days without eclipse (80 days). Also, an evaluation period (analyzed by SP, UP and IE to assess performance) runs during 2013 and includes a sampling of days different than the operations period but also only days without eclipse (80 days). In the training period, 100 days correspond to times with “nominal” observations while the other days correspond to both types of “anomalous” observations in roughly equal frequency. In the operations and evaluation periods, half the days correspond to times with “nominal” observations while the other half correspond to both types of “anomalous” observations. Figure 8 displays a sample light curve from the training period corresponding to nominal observations (2012 Nov 30). The time between individual measurements was set to 600 seconds.

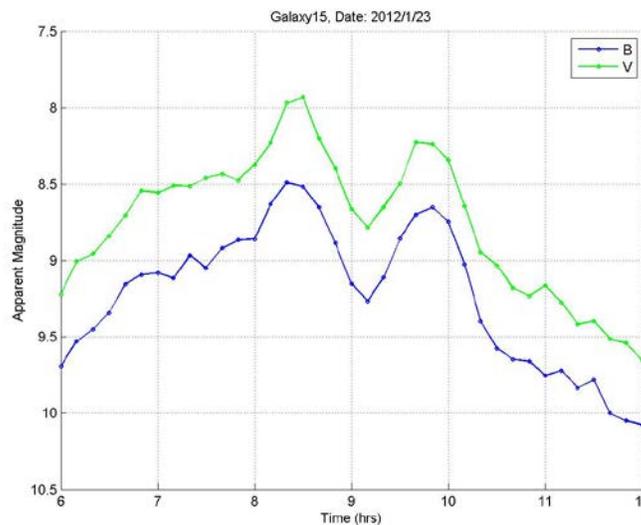


Figure 8. Sample Light Curves during Original Training Period

6. RESULTS

To reiterate, SP does not require knowledge of the space object orbital parameters, geometric structure, or surface properties. It does require that the data collected be placed into a multidimensional space in which each dimension represents essentially the same empirical measurement of the object under study. For the geosynchronous space objects in this scenario, as long as the observations are made during a consistent time period, the observational geometry profile during the collection does not change very much from day to day. The simulated light curves for this scenario possess the properties of a uniform sampling rate during a consistent time period. As can be seen in the results, this was sufficient to achieve acceptable performance with no further attempts at normalization of the light curves.

The performance of the SP classifiers is summarized in Table 4. The performance of the SP classifiers was quite good on the "original" data sets. ROC curves are not shown for these sets. The "harder" data sets better demonstrate the evaluation of potential Integration Engine performance on overall system performance.

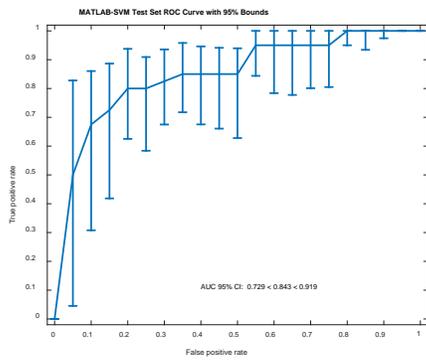
In generating these results, the SP classifiers were trained on the "training" data set, followed by testing with the "operations" data set. The SP classifiers were not exposed to the "operations" data set during training. Again, the light curve data was not normalized to zero-mean and unit variance, as is the usual practice in pattern classification.

Table 4. Summary of Results for SP Classifiers

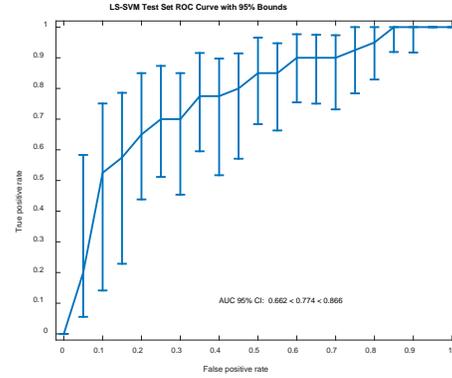
Variation	Data Set	AUC	
original	training	ML-SVM	$0.992 \leq 0.998 \leq 1.000$
		LS-SVM	$0.958 \leq 0.983 \leq 0.994$
		INC-SVDD	$0.640 \leq 0.731 \leq 0.803$
		KSVDD (RBF)	$0.663 \leq 0.753 \leq 0.824$
		KSVDD (Exp)	$0.706 \leq 0.793 \leq 0.863$
	operations	ML-SVM	$0.932 \leq 0.972 \leq 0.991$
		LS-SVM	$0.926 \leq 0.972 \leq 0.990$
		INC-SVDD	$0.626 \leq 0.758 \leq 0.860$
		KSVDD (RBF)	$0.624 \leq 0.756 \leq 0.851$
		KSVDD (Exp)	$0.626 \leq 0.758 \leq 0.865$
	evaluation	ML-SVM	$0.891 \leq 0.953 \leq 0.984$
		LS-SVM	$0.929 \leq 0.975 \leq 0.992$
		INC-SVDD	$0.674 \leq 0.785 \leq 0.875$
		KSVDD (RBF)	$0.601 \leq 0.732 \leq 0.842$
		KSVDD (Exp)	$0.595 \leq 0.724 \leq 0.829$
harder	training	ML-SVM	$0.982 \leq 0.993 \leq 0.998$
		LS-SVM	$0.978 \leq 0.992 \leq 0.997$
		INC-SVDD	$0.479 \leq 0.558 \leq 0.648$
		KSVDD (RBF)	$0.466 \leq 0.560 \leq 0.652$
		KSVDD (Exp)	$0.536 \leq 0.628 \leq 0.715$
	operations	ML-SVM	$0.751 \leq 0.856 \leq 0.923$
		LS-SVM	$0.680 \leq 0.797 \leq 0.882$
		INC-SVDD	$0.427 \leq 0.566 \leq 0.694$
		KSVDD (RBF)	$0.444 \leq 0.582 \leq 0.710$
		KSVDD (Exp)	$0.409 \leq 0.547 \leq 0.677$
	evaluation	ML-SVM	$0.765 \leq 0.861 \leq 0.921$
		LS-SVM	$0.776 \leq 0.869 \leq 0.932$
		INC-SVDD	$0.467 \leq 0.599 \leq 0.727$
		KSVDD (RBF)	$0.459 \leq 0.601 \leq 0.717$
		KSVDD (Exp)	$0.477 \leq 0.605 \leq 0.723$

The two-class classifiers were very successful in learning to represent the training data. It should be pointed out that the one-class classifiers were trained exclusively on the nominal samples from the "training" data sets. Since these data sets include anomalous samples, it was possible to evaluate their performance on the full "training" data sets. These indicate that the one-class classifiers have significant difficulty in fully learning the support of the nominal distribution even on the "training" data sets.

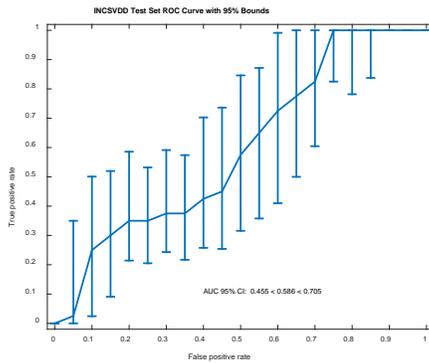
Figure 9 shows the performance of the SP classifiers on the "harder" operations data set. Pointwise confidence intervals on the true positive rate (TPR) were computed using vertical averaging and sampling using bootstrapping with 1000 replicas. These results demonstrate that this variant was indeed more difficult. Performance of the two-class classifiers is still acceptable, but one-class classifier performance is weak. However, note again that in the context of ensemble processing, weak classifiers can be used to improve overall ensemble performance.



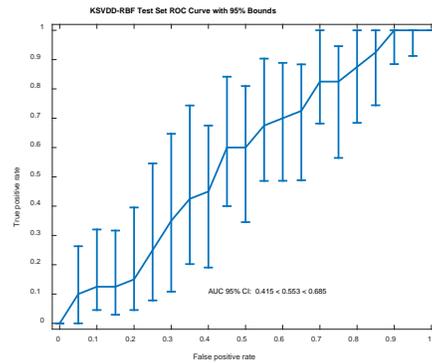
Matlab SVM



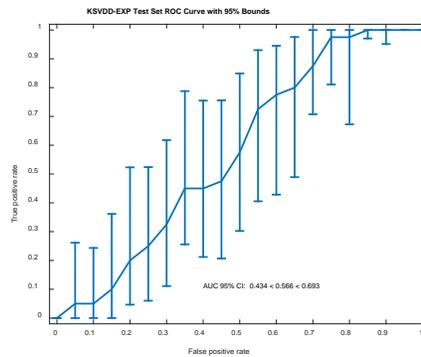
LS-SVM



INCSVDD



KSVDD (RBF Kernel)



KSVDD (Exponential kernel)

Figure 9. SP Classifier Performance On "Harder" Operations Data Set: Matlab SVM; Least-Squares SVM; Incremental SVDD; KSVDD (RBF Kernel); And KSVDD (Exponential Kernel).

UP only requires a viable shape/surface model to process the results. Two separate models were used. In both, the full shape model of Figure 7 was used. In the first, the BRDF parameters of Table 2 were used as is, while in the second, slightly corrupted BRDF parameters with an associated uncertainty were used. These BRDF parameters are shown in Table 5. An analysis where a simplistic model (e.g. 5 facets representing Galaxy 15) was also created and used in the UP algorithm and this was reported previously [28].

Table 5. Model BRDF Parameters (Corrupted)

	Bus and Dish	Solar Panel
m – microfacet slope	0.0983 ± 0.0050	0.0567 ± 0.0050
ρ_B - reflectance	0.8742 ± 0.0500	0.6868 ± 0.0500
d_B - diffuse fraction	0.6864 ± 0.0500	0.2316 ± 0.0500
ρ_V - reflectance	0.8374 ± 0.0500	0.6524 ± 0.0500
d_V - diffuse fraction	0.7360 ± 0.0500	0.1839 ± 0.0500

Each training/operations/evaluation day was processed using the UP algorithm (USKF) described in previously. For Galaxy 15, the solar panel east-west articulation angles were estimated while the BRDF parameter values were considered. In each case, there were twelve different initial hypotheses corresponding to different solar panel east-west articulation angle pairs. Figure 10 shows the output for both the original and harder cases from the operations period. The initial hypothesis states are shown in black, the final estimated states are shown in red, the composite state accounting for the final weights is shown in green, the “nominal” state (as determined from the training set) is shown in blue, and the 1σ , 2σ , and 3σ probability distribution function (pdf) contours are shown as green lines. The pdf was built from a weighted Gaussian mixture of the final states for the different hypotheses, and the pdf value associated with the “nominal” articulations was then calculated. It was the associated “sigma” value (constrained to be between 0.1 and 10) as determined by the probabilities (i.e. 1σ contour corresponds to the fraction 0.6827 of the pdf contained within) that was used to determine whether a particular day was considered “nominal” or “anomalous”.

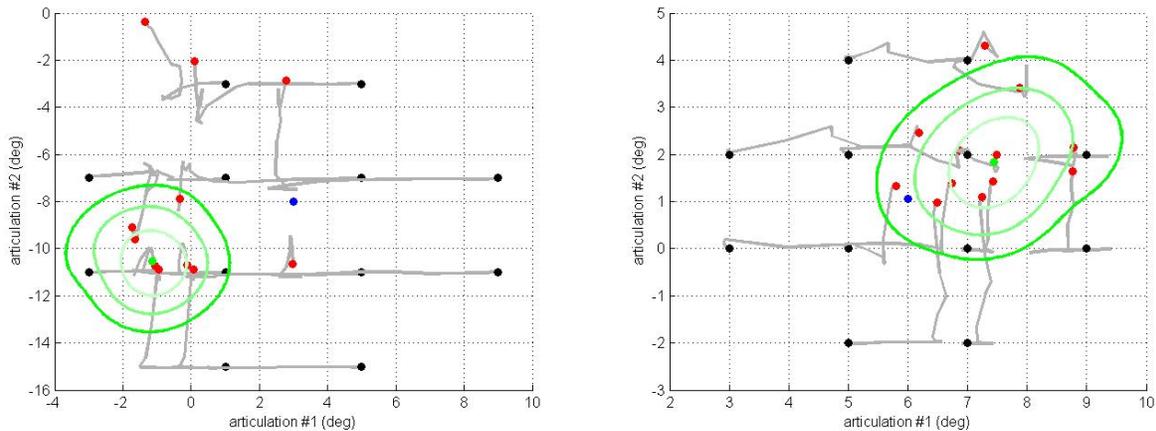


Figure 10. USKF Results Example For Galaxy 15 Original (Left) And Harder (Right) Data Sets

Figure 11 displays the pdf values for the original case with corrupted model for the operations period as an example. Nominal days are represented by green dots, and anomalous days are represented by yellow and red dots.

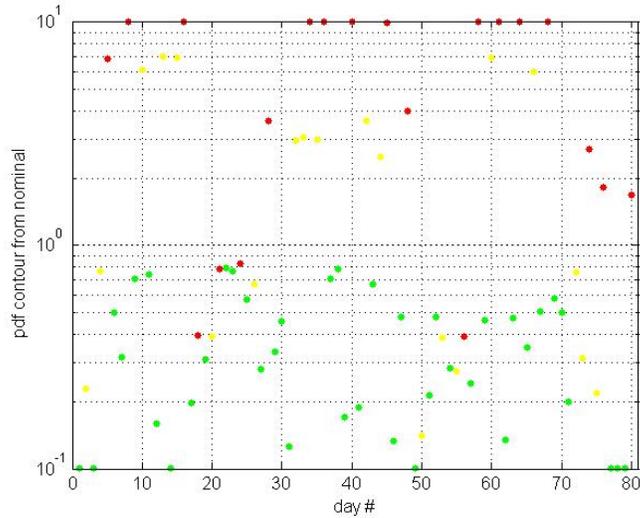


Figure 11. Probability Distribution Function Sigma Value Of Nominal For Original Data Set And Corrupted Model Case

Adjusting the threshold continuously from below the lowest pdf value to above the highest pdf value generates a Receiver Operations Characteristic (ROC) curve. The area under the curve (AUC) was used to quantitatively compare ROC curve results. Figure 12 displays the ROC curve for the original data sets with the corrupted model for the operations period. Note the results of the operations data corrupted model were used to train IE, and the evaluation data corrupted results were used to evaluate and compare performance.

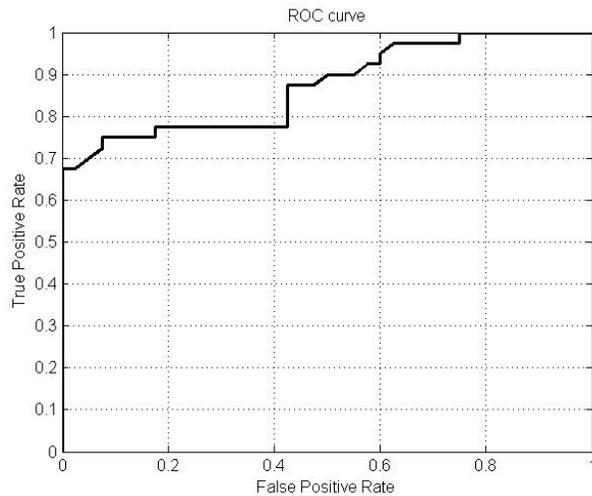


Figure 12. ROC Curves Original Data Set And Corrupted Model Case

Table 6 lists all UP results for all data sets that were processed as well as a consolidated “all” value which includes all non-eclipse training, operations, and evaluation data. The Matthew’s Correlation Coefficient (MCC) represents the maximum in that value for a particular point along the model ROC curve at the threshold specified. On its own, this threshold was used to establish UP’s vote as to whether a particular set of observations was nominal or anomalous.

Table 6. UP Results For All Data Sets Processed

Variation	AUC (true)	AUC
original	training	0.852 ± 0.028
	training (no eclipse)	0.880 ± 0.029
	operations	0.873 ± 0.035
	operations (true model)	0.968 ± 0.012
	evaluation	0.920 ± 0.028
	all (no eclipse)	0.895 ± 0.018
harder	training	0.659 ± 0.041
	training (no eclipse)	0.733 ± 0.042
	operations	0.801 ± 0.047
	operations (true model)	0.879 ± 0.035
	evaluation	0.842 ± 0.041
	all (no eclipse)	0.785 ± 0.026

The uncertainties in the AUC values were estimated using randomly generated Gaussian distributions of various widths and separations and the average and standard deviations of the resulting AUC values for the ROC curves given a sample size of $N = 184$ (training), $N = 138$ (training, no eclipse), $N = 80$ (operations or evaluation), or $N = 298$ (all, no eclipse). This estimate of uncertainty was checked by also calculating the standard deviation of AUCs generated when drawing 80 random samples from the $N = 138$ no-eclipse training set and was found to be consistent. Note again it was the training sets that were used by SP (with knowledge of nominal versus anomalous data) to create the various classifiers, the operations sets that were used by IE (again with knowledge of nominal versus anomalous data) to create an optimized classifier for each scenario and the evaluation sets that were used to measure and compare performance between SP, UP, and IE (no prior knowledge of nominal versus anomalous data). Figure 13 displays all the results. Again, UP never has prior knowledge of nominal versus anomalous data.

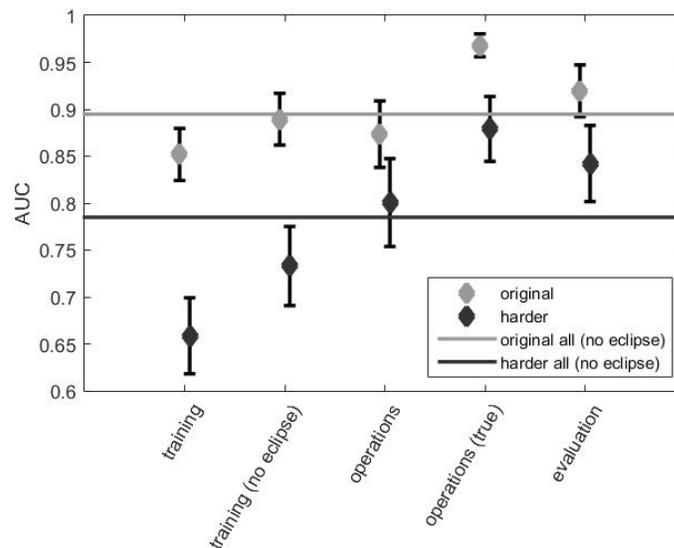


Figure 13. AUC Values For UP Analysis Of Various Cases And Data Sets

Including the light curves with eclipse periods in the training set degraded the performance while using the true BRDF values in the operations set improved the performance significantly.

The performance of the IE on the "original" data sets was very good, as summarized in Table 7. ROC curves for this data set are not shown. For each satellite, another more challenging data set was created, and called the "harder" set. The "harder" data set poses more problems for SP and UP, and thus better supports evaluation of potential Integration Engine impact on overall system performance.

Table 7. Summary of Results for IE Ensemble

Variation	Data Set	AUC
original	operations	0.9738
	evaluation	0.9666
harder	operations	0.9024
	evaluation	0.9541

In generating these results, the IE bagged CART ensemble was trained on the outputs of SP and UP on the "operations" data set, followed by testing with the "evaluation" data set. IE was not exposed to the "evaluation" data set during training. Additionally, the classifier outputs from SP were normalized to a consistent range. In general, the IE was very successful at learning to represent the SP and UP output on the operations data. Note that only data for the case in which the IE ensemble decision was reached based on an average vote among the member classifiers is shown. Other variants were tried, as well, including majority voting. This is an implementation detail that can be further investigated in future work.

Figure 14 shows the performance of the IE ensemble on the "harder" "evaluation" data set for both satellites. Note the improvement in going from "operations" data to the "evaluation" data. The cross-validation results on "operations" data were expected to understate the performance of IE on the "evaluations" data. The improvement in the performance between the "operations" and "evaluation" sets was larger than expected and can be considered a "lucky" outcome. With a small data set, the improvement can be attributed to the properties of several anomaly samples. Comparison of the IE results to those of SP and UP alone demonstrate again that "the whole is greater than the sum of the parts," and that IE was finding improvements from merging these diverse approaches.

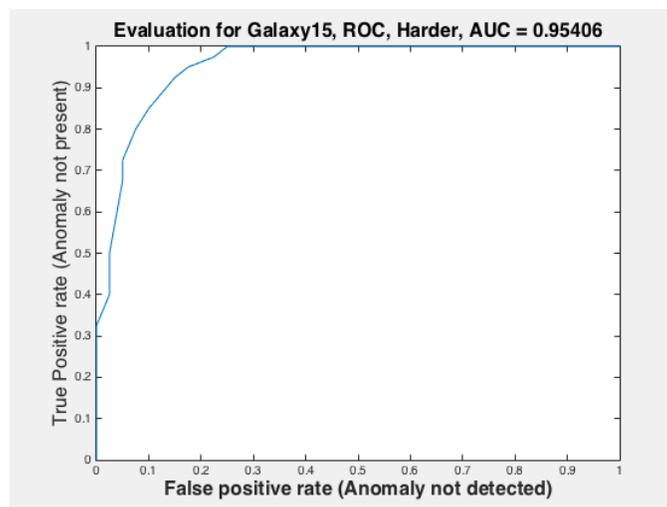


Figure 14. IE Ensemble Performance On The "Harder" Evaluation Data Set

Figures 15 and 16 compare the performance of the individual SP classifiers, UP and IE (both using SP inputs only and using ISUP) for all data sets. Of note is the improved performance of IE when compared to SP and UP separately. This is particularly evident in the "harder" variation in Figure 16.

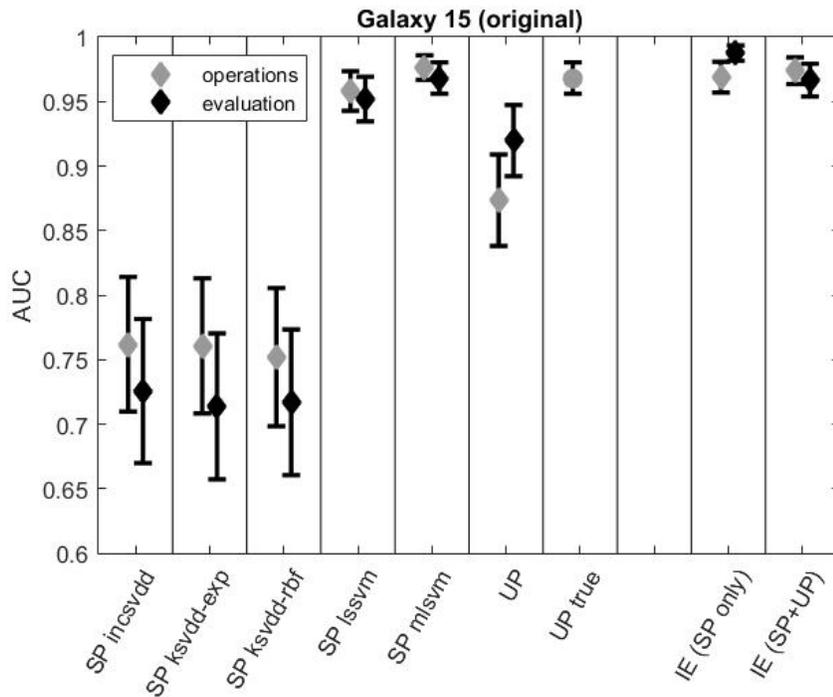


Figure 15. Original Consolidated Performance Plot

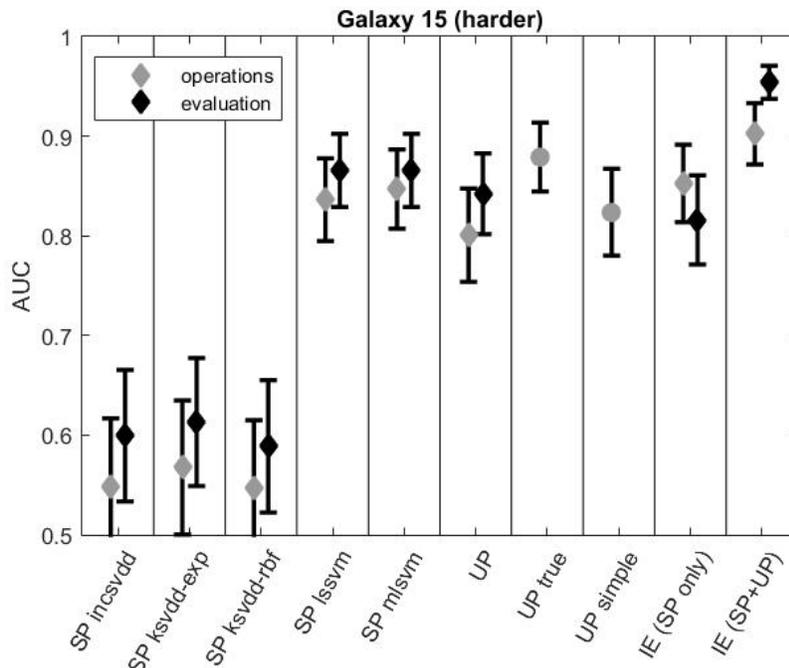


Figure 16. Harder Consolidated Performance Plot

7. CONCLUSIONS

ISUP has shown that photometric data can provide meaningful feature identification and change detection when used in a system architecture that integrates the results of separately processing Supervised and Unsupervised photometry data. ISUP has demonstrated that SP can do change detection effectively and without any a priori modeling of a space object. ISUP has demonstrated that UP can do change detection through the estimation of model parameters and subsequent analysis of the weights produced by sequential estimation processing (e.g., Kalman filters).

ISUP has demonstrated that UP can effectively estimate the parameters of the space object constructive model such as the surface BRDF, attitude & orientation, and some types of articulation. ISUP has demonstrated that combining SP and UP into a hierarchical structure is a credible approach to the fusion of their learning outputs, and further demonstrated that the results of SP and UP can be successfully combined. ISUP has demonstrated that there is promise in UP low-complexity modeling of objects and light curves. These simple models utilize seasonal variations in their parameters and are able to adapt and reproduce the light curves simulated using much more complex models with static parameters.

The most important result from the development and testing of ISUP is the demonstration that the fusion by IE of outputs from SP and UP results in better anomaly detection performance than either technique operating on its own.

8. REFERENCES

- [1] Duda, R. O., Hart, P. E., and Stork, D. G., *Pattern Classification* (2nd Edition), Wiley-Interscience, Hoboken, NJ, 2001.
- [2] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter*, Artech House, Boston, 2004, pp. 3-80.
- [3] Hunt, B.R., Hamada, K., Wetterer, C.J., "Feature Identification from Unresolved Electro-Optical Data," Air Force Research Laboratory Phase I Final Report (FA9451-12-M-0311), March 2013.
- [4] Haykin, S. O., *Adaptive Filter Theory* (5th Edition), Prentice-Hall, Upper Saddle River, NJ, 2013.
- [5] Breiman, L., J. Friedman, R. Olshen, and Stone, C., *Classification and Regression Trees*, CRC Press, Boca Raton, FL, 1984.
- [6] Christianini, N., and J. C. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press, 2000.
- [7] Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J., *Least Squares Support Vector Machines*, World Scientific, Singapore, 2002 (ISBN 981-238-151-1).
- [8] Pelckmans K., Suykens J.A.K., Van Gestel T., De Brabanter J., Lukas L., Hamers B., DE Moor B., Vandewalle J., "LS-SVMLab : a Matlab/C toolbox for Least Squares Support Vector Machines", Internal Report 02-44, ESAT-SISTA, KU Leuven (Leuven, Belgium), 2002., Lirias number: 21472.
- [9] LS-SVMLab Toolbox, Version 1.8: available at <http://www.esat.kuleuven.be/sista/lssvmlab/>
- [10] Tax, D.M.J. and Duin, R.P.W., "Support Vector Domain Description," *Pattern Recognition Letters*, Vol. 20, No. 11-13, pp. 1191-1199, Dec 1999.
- [11] Tax, D. M. J., and Duin, R. P. W., "Support Vector Data Description," *Machine Learning*, Vol. 54, pp. 45-66, 2004.
- [12] DDtools2015, Tax, D. M. J., June, Version 2.1.2, "DDtools, the Data Description Toolbox for Matlab," 2015 (available at http://prlab.tudelft.nl/david-tax/dd_tools.html).
- [13] Julier, S.J., Uhlmann, J.K., "A New Extension of the Kalman Filter to Nonlinear Systems," *Proceedings of SPIE – The International Society for Optical Engineering*, Vol. 3068, Orlando, FL, Apr 1997, pp. 182-193.
- [14] van der Merwe, R., Wan, E.A., "The Square Root Unscented Kalman Filter for State and Parameter-Estimation," *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 6, Salt Lake City, UT, May 2001, pp. 3461-3464.

- [15] Crassidis, J.L., Markley, F.L., “Unscented Filtering For Spacecraft Attitude Estimation,” *Journal of Guidance, Control, and Dynamics*, **26**, No. 4, 2003, pp. 536-542.
- [16] Schmidt, S. F., “Application of state-space methods to navigation problems,” *Advances in Control Systems*, vol. 3, 1966, pp. 293–340.
- [17] Stauch, J. & Jah, M., “On the Unscented Schmidt Kalman Filter Algorithm,” *Journal of Guidance, Control, and Dynamics*, accepted for publication
- [18] van der Merwe, R., Doucet, A., de Freitas, J.F.G, and Wan, E., “The Unscented Particle Filter,” Technical Report CUED/F-INFENG/TR 380, Cambridge University Engineering Department, 2000.
- [19] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter*, Artech House, Boston, 2004, pp. 48-57.
- [20] DeMars and K., M. Jah, “Passive Multi-Target Tracking with Application to Orbit Determination for Geosynchronous Objects,” AAS Paper 09-108, 19th AAS/AIAA Space Flight Mechanics Meeting, Savannah, Georgia, Feb 2009.
- [21] DeMars, K., M. Jah and P. Schumacher, “The Use of Short-Arc Angle and Angle Rate Data for Deep-Space Initial Orbit Determination and Track Association,” AAS Paper 10-153, 20th AAS/AIAA Space Flight Mechanics Meeting, San Diego, CA, Feb 2010.
- [22] Linares, R., Crassidis, J. L., Jah, M. K., and Kim, H., “Astrometric and Photometric Data Fusion for Resident Space Object Orbit, Attitude, and Shape Determination Via Multiple-Model Adaptive Estimation,” AIAA Paper #2010-8341, *AIAA Guidance, Navigation, and Control Conference*, Toronto, Canada, Aug 2010.
- [23] Linares, R., M. K. Jah, J. L. Crassidis, F. A. Leve, T. Kececy, “Astrometric and Photometric Data Fusion for Space Object Mass and Area Estimation,” *Acta Astronautica*, 99, Jun–Jul 2014, pp.1–15.
- [24] Hol, J.D., T.B. Schon, and F. Gustafsson, “On Resampling Algorithms for Particle Filters,” *Nonlinear Statistical Signal Processing Workshop*, Cambridge, United Kingdom, Sep 2006 (Institute of Electrical and Electronics Engineers)
- [25] Fawcett, T “An Introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, pp. 861–874.
- [26] Cook, R. L., and Torrance, K. E., “A reflectance model for computer graphics,” *ACM Transactions on Graphics*, Vol. 1, No. 1, Jan 1982, pp. 7-24.
- [27] Schlick, C., “An Inexpensive BRDF Model for Physically-Based Rendering,” *Computer Graphics Forum*, **13**, No. 3, 1994, pp. 233-246.
- [28] Wetterer, C. J., Hunt, B., Sheppard, D., 2015. Using simplistic shape/surface models to predict brightness in estimation filters. The Advanced Maui Optical and Space Surveillance Technologies Conference, Wailea, Maui, Hawaii, September 15-18 (Maui Economic Development Board).