EXPERIMENTS IN BAYESIAN DIAGNOSTICS WITH IUID-ENABLED DATA

Stephyn G. W. Butcher
John W. Sheppard
Department of Computer Science
The Johns Hopkins University
3400 N. Charles Street
Baltimore, MD 21218
{sbutche2,jsheppa2}@jhu.edu

Mark A. Kaufman, Hanh Ha
Craig MacDougall
NSWC Corona Division
US Navy
PO Box 5000
Corona, CA 91718
{*first.last*}@navy.mil

Abstract—The Department of Defense (DOD) has recognized the importance of improving asset management and has created Item Unique Identification numbers (IUIDs) to improve the situation. IUIDs will be used to track financial and contract records and obtain location and status information about parts in DoD inventory. IUIDs will also support data collection for weapon systems from build, test, operations, maintenance, repair, and overhaul histories. In addition to improving the overall logistics process, IUIDs offer an opportunity to utilize asset-specific data to improve system maintenance and support. An Office of the Secretary of Defense (OSD) Pilot Project to implement IUID on a Navy weapon system presents an immediate opportunity to evaluate this use of IUID data. This paper reports on experiments conducted to see if a set of asset-specific diagnostic classifiers trained on subsets of data is more accurate than a general, composite classifier trained on all of the data. In general, it is determined that the set is more accurate than the single classifier given enough data. However, other factors play an important role such as system complexity and noise levels in the data. Additionally, the improvements found do not arise until larger amounts of data are available. This suggests that future work should concentrate on tying the process of data collection to the estimation of the associated probabilities.

I. INTRODUCTION

As a method to improve its asset management capabilities, the DoD has instituted the IUID program where unique identification numbers are associated with DoD assets. These unique identification numbers are asset specific and through their consistent usage, responsible parties will be able to account for the DoD's massive inventories. However, the IUID program will not only be a boon to logistics; the IUID program will also support data collection from build, test, operations, maintenance, repair and overhaul histories. This data in turn will be able to support improved system maintenance, diagnosis and prognosis.

One of the interesting possibilities this data collection offers is the opportunity to improve models for diagnosis and prognosis by providing asset-specific data. Generally speaking, when constructing a classifier such as a diagnostic system, the more closely the distribution of the sample data and the distribution of the target population match, the more accurate the model will be. It follows that if the behavior of some assets deviates substantially from the average, asset-specific models for those assets should be more accurate than a composite model that covers all of the assets.

Fortunately, modern assets have large mean time between failure (MTBF). Unfortunately this means that large quantities of asset-specific data are not immediately forthcoming. This raises questions about how to develop models prior to the availability of the needed data. Are there advantages to aggregating the data we have until more specific data is available? How do we know when we should be aggregating data to make composite models and when should we start segregating the data to make (possibly more accurate) asset-specific models? Is quantity of data the only important factor or are there others? Although the experiments in this paper do not

specifically address all of these questions, the results suggest future directions of research.

This paper is but one in a set that discusses theoretical and experimental issues associated with applying Bayesian approaches to diagnostics and prognostics in an IUID enabled environment. The ultimate aim is to apply these results to the IUID data provided under an OSD Pilot Project for a US Navy weapon system.

The paper starts in section II by describing the IUID program. Section III discusses Bayesian approaches to diagnosis as a special case of classification. Section IV then describes the experimental design and Sections V and VI provide experimental results and a discussion of the results respectively. Section VII details plans for future work, and section VIII provides concluding remarks.

II. DIAGNOSIS AND IUID

The DoD has a significant concern about the tracking and support of individual systems within their inventory. With the conclusion of Desert Storm, it was found that 35,000 large shipping containers of supplies had gone unused and needed to be redeployed. The task of returning these containers (and their contents) to the supply system or redeploying them to other theaters was monumental. The Government Accounting Office (GAO) concluded that visibility and accountability over the $3.4 billion worth of material was lost in the process [4]. Incomplete tracking of DoD hardware location and history was the primary reason for the loss of accountability. In addition, valuable location, maintenance, reliability, and diagnostic information was not properly obtained, analyzed, or retained. The inability to access and use this information presented a significant obstacle to making effective program decisions, reducing maintenance and support costs, and enabling next-generation approaches to system support (e.g. advance diagnostics, reliability centered maintenance, and prognostics).

The DoD began several initiatives to get more effective control over its logistics. Two of these initiatives, Item Unique Identification (IUID) and Radio Frequency Identification (RFID) are of interest in this paper. Both IUID and RFID are DoD requirements [13], [14]. In short, between these two initiatives the location and history of individual items will be tracked worldwide.

RFID uses radio frequency tags on pallets, shipping containers, etc. to track a unique number. That number is in the supply database and is linked to the contents, destination, and origin of the package. IUID uses a data matrix (which is *not* RF) to identify individual items. The data matrix is similar to a two-dimensional representation of a bar code, is scalable, can range in size from a few tenths of an inch to up to 14 inches, and can represent up to 2,000 characters. A crucial part of IUID is the unique number—the Unique Item Identifier (UII). The UII is not just assigned by the manufacturer or depot. There is a protocol followed to generate the UII [12]. The number is then uploaded to a DoD registry where it is checked for uniqueness and logged into the system. The IUID that encodes the UII is then linked to a particular part for the lifetime of the part. IUIDs can be applied as labels, plates, laser etched directly, and through a variety of other techniques [1]. Proper identification of an asset is essential to correlating pedigree and reliability prediction of the asset as well as enabling reliability centered maintenance, advanced diagnostics and prognostics.

III. BAYESIAN APPROACHES TO DIAGNOSTICS

Developing system models for diagnosis is complex and often depends on a detailed understanding of system performance and test engineering. Learning diagnostic models from field maintenance data offers considerable potential to develop or refine diagnostics for fielded systems. Simulation can also be used to generate data for purposes of learning. Several approaches exist for learning such models including case based reasoning, decision tree induction, neural networks, and Bayesian methods. We suggest applying Bayesian methods to diagnosis because they derive classification “rules” based on sound mathematical principles (namely, probability theory), they can adapt easily as more data is obtained, and they have been empirically demonstrated to perform well on a broad range of classification problems.

Previously, we provided a detailed derivation of a simple model for Bayesian diagnosis [10]. We have also demonstrated how both the Naïve Bayesian Network and what we have called the “Not So Naïve” Bayesian Network or Tree-Augmented Bayesian network perform on a small sample of IUID-enabled data for a US Navy

weapon system [9]. However, for the purposes of this paper, the Naïve Bayesian Network will be sufficient for the experiments we are going to perform. Therefore, we will concentrate on naïve Bayesian Networks.

The primary assumption for naïve Bayesian networks is that the evidence variables in the network (i.e., the tests) are conditionally independent of each other given the class (i.e., diagnosis). To start our discussion on the implications of this assumption, let us define our diagnostic networks as if they contain only one diagnosis variable with n possible values (corresponding to each of the diagnostic conclusions D_i). Thus, we will apply a simple network structure corresponding to the form shown in Figure 1. Note that this structure can be modified where there is a separate node D_i for each diagnosis rather than a single composite diagnosis node. This leads to the so-called naïve Bayes "multi-net" [2], [3].

Under the naïve Bayes model, we consider the diagnosis problem as finding the class label (i.e., diagnosis) that maximizes the *a posteriori* probability of the specific class given the set of observations:

$$D = \arg\max_{D_i \in \mathbf{D}} \Pr(D_i \mid o(T_1), \ldots, o(T_n)) = \arg\max_{D_i \in \mathbf{D}} \Pr(o(T_1), \ldots, o(T_n) \mid D_i) \Pr(D_i).$$

But the problem remains that the joint distribution over the tests is exponential in the number of tests. The naïve Bayes assumption states that we can treat each of these observations as if they are conditionally independent given the diagnosis, and this leads to the classification rule:

$$D = \arg\max_{D_i \in \mathbf{D}} \Pr(D_i) \prod_{j=1}^{n} \Pr(o(T_j) \mid D_i).$$

Given a set of training data mapping test results (outcomes) to actual faults repaired, we can "learn" the naïve Bayes network by observing that $\Pr(D_i)$ is simply the frequency of occurrence of a particular fault in the data set and similarly, $\Pr(o(T_j) \mid D_i)$ is the frequency of test outcome $o(T_j)$ considering only the particular diagnosis D_i. What is remarkable about this simple model is the considerable effectiveness it has demonstrated in numerous experiments and implementations [5].

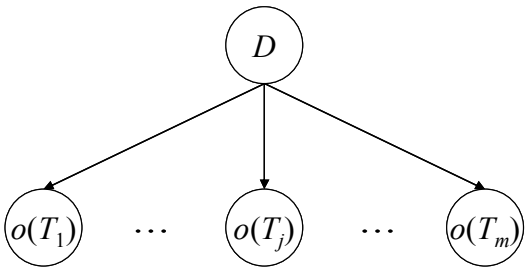


Figure 1. Naïve Bayes diagnostic network.

In considering the computational complexity of naïve Bayes networks, we must account for the complexity to learn the models as well as to use the models for diagnosis. The complexity of learning depends on deriving the probability estimates for $\Pr(D_i)$ and $\Pr(o(T_j) \mid D_i)$. Let n be the number of examples, $\delta = |\mathbf{D}|$, and $\tau = |T|$. It is reasonable to assume $\delta < n$. Then the complexity for deriving $\Pr(D_i)$ for all $D_i \in \mathbf{D}$ is $O(n + \delta) = O(n)$, and the complexity for deriving $\Pr(o(T_j) \mid D_i)$ for all T_j and D_i is $O(\tau \times n + \tau \times \delta) = O(\tau \times n)$. Classification involves multiplying $\tau + 1$ probabilities for each diagnosis and maximizing, so the complexity of classification is $O(\tau \times \delta)$.

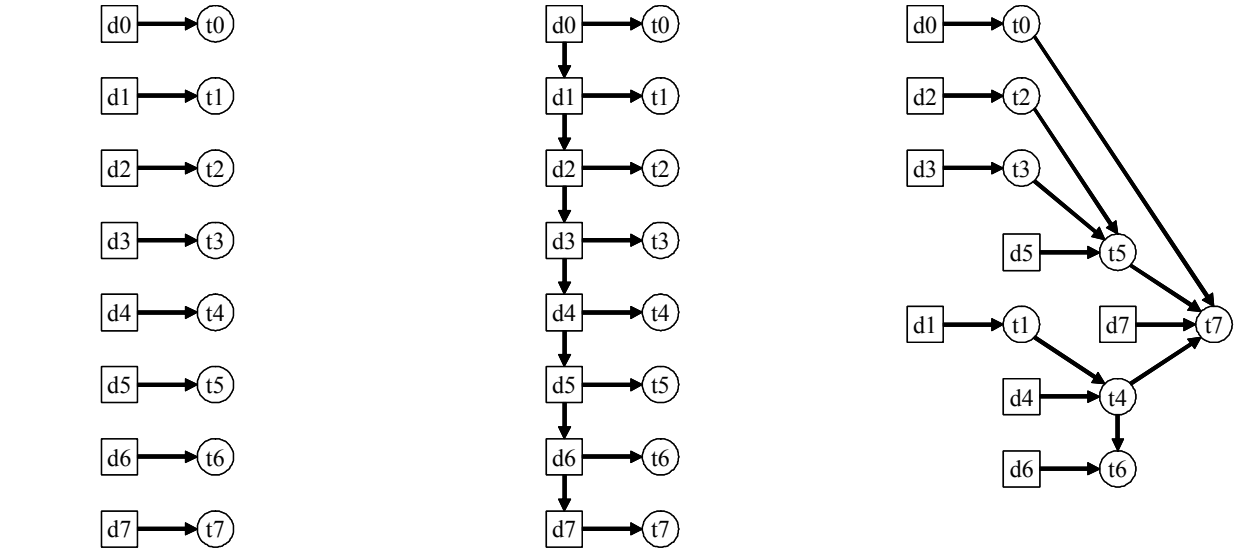
An important ramification of the classification rule above is that if any $\Pr(o(T_j) \mid D_i)$ should happen to be zero then the entire value of the expression for that particular D_i zeroes out. This is not generally what we want but may happen if our training data was missing low probability observations. The Bayesian solution is to resort to some sort of prior probability to keep the formula from "breaking." Although we will have more to say about this later, the approach we use is an *m*-estimate (or *Dirichlet prior*) calculated as follows [7]:

$$\Pr(o(T_i) \mid D_j) = \frac{n_c + mp}{n + m}$$

where n_c is the number of instances in the data pairing particular values for $o(T_i)$ and D_j, n is the total number of instances in the data corresponding to diagnosis D_j, p is a prior estimate for the probability, and m is the number of "virtual" examples in the data.

IV. EXPERIMENTS

The central hypothesis of this paper is that segregating data to train asset-specific networks



	t0	t1	t2	t3	t4	t5	t6	t7
d0	1	0	0	0	0	0	0	0
d1	0	1	0	0	0	0	0	0
d2	0	0	1	0	0	0	0	0
d3	0	0	0	1	0	0	0	0
d4	0	0	0	0	1	0	0	0
d5	0	0	0	0	0	1	0	0
d6	0	0	0	0	0	0	1	0
d7	0	0	0	0	0	0	0	1

a) one-for-one

	t0	t1	t2	t3	t4	t5	t6	t7
d0	1	1	1	1	1	1	1	1
d1	0	1	1	1	1	1	1	1
d2	0	0	1	1	1	1	1	1
d3	0	0	0	1	1	1	1	1
d4	0	0	0	0	1	1	1	1
d5	0	0	0	0	0	1	1	1
d6	0	0	0	0	0	0	1	1
d7	0	0	0	0	0	0	0	1

b) serial

	t0	t1	t2	t3	t4	t5	t6	t7
d0	1	0	0	0	0	0	0	1
d1	0	1	0	0	1	0	1	1
d2	0	0	1	0	0	1	0	1
d3	0	0	0	1	0	1	0	1
d4	0	0	0	0	1	0	1	1
d5	0	0	0	0	0	1	0	1
d6	0	0	0	0	0	0	1	0
d7	0	0	0	0	0	0	0	1

c) functional

Figure 2. Sample systems. a) Diagnoses are in a one-to-one relationship with the tests. b) Diagnoses are in a serial configuration. c) Diagnoses correspond to some hypothetical "functional" relationship with the tests.

creates a set of networks with a greater accuracy than aggregating the data to train a single composite network. Unfortunately, it isn't that simple. Although the Naïve Bayesian Network (NBC for Naïve Bayesian *Classifier*) has been shown to train well on relatively few examples [6], consider an aggregated data set of 100 samples. If there are ten assets, this leaves, on average, only ten instances per asset. If there are ten possible diagnoses, this leaves, on average, only one diagnosis per asset. At this point, we may not even be able to determine if assets actually have substantially different fault distributions. While the question is easy to answer for this contrived example, clearly *when* to segregate is as much an important part of the hypothesis as *should* we segregate.

To test our hypothesis, we decided first to generate some synthetic data. We imagine a hypothetical system having eight components that can be arranged in various ways. Each component is subject to failure and that failure is detected by a combination of eight tests. Each individual test is considered to pass if it detects 4.9 volts or more at the test point; alternatively, if it detects less than 4.9 volts, the test fails.[1] Depending on how the components are arranged, the diagnostic characteristics of each system are captured by a corresponding D-Matrix [11]. For these experiments, three systems were selected:

[1] One can also consider an equivalent upper limit where voltage greater than 5.1 volts yields failure; however, we decided to limit our synthetic data to failure modes where voltage was low.

Table 1. Sample distributions used in experiments.

	Diagnosis							
Distribution	d0	d1	d2	d3	d4	d5	d6	d7
Composite	14%	11%	11%	17%	8%	11%	17%	11%
1 Bad Actor #1	7%	7%	7%	53%	7%	7%	7%	7%
2 Bad Actors #1	5%	5%	5%	36%	5%	5%	36%	5%
3 Bad Actors #1	28%	3%	3%	28%	3%	3%	28%	3%
1 Bad Actor #2	7%	7%	53%	7%	7%	7%	7%	7%
2 Bad Actors #2	36%	5%	5%	5%	5%	5%	5%	36%
3 Bad Actors #2	3%	28%	3%	3%	3%	28%	28%	3%
Non-Uniform #1	21%	11%	3%	21%	11%	3%	21%	11%
Non-Uniform #2	3%	23%	11%	3%	23%	11%	3%	23%
Non-Uniform #3	13%	3%	26%	13%	3%	26%	13%	3%
Uniform	13%	13%	13%	13%	13%	13%	13%	13%

"one-for-one," "serial," and "functional" which gives rise to three D-Matrices (Figure 2).

Because the main point of these experiments is to examine whether it is better to build asset-specific or composite models, the experiment is only interesting if the data exhibit some variability at the asset level. For our purposes, we represent an asset as a particular distribution of component failures. For example, "Asset A" might always have trouble with component 1. In this case, the probability of component 1 failing will be relatively higher than the probability of components 2—8 failing. For these experiments, we created ten such hypothetical failure distributions each expressing a different behavioral quirk such as "one bad actor", "two bad actors", "three bad actors", "non-uniform" and "uniform". Table 1 shows the distributions as well as the overall, composite, distribution.

These D-Matrices (systems) and component failure distributions (assets) form the foundation for generating the synthetic data. For any given data set size, *N*, there are *N* data points created for *each* asset given a particular system. For example, if *N* = 100 and the system is "one-for-one", creating data for the "one bad actor #1" asset involves creating data that includes seven each of signatures 0–2 and 4–7 but 53 of signature 3. This process is repeated for each system, each asset and each *N* value: 25, 50, 100, 250, 500, 1000, 2500 and 5000.

An NBC can easily learn the concepts represented by the D-matrices to 100% accuracy –regardless of whether the data is aggregated or segregated—because the concepts represented are all linearly separable [8]. In order to test our hypothesis we require at least the possibility that these can diverge. In order to do so, we note that copying the diagnostic signature over and over to create the data implies an assumption that the data is noise free. In the real world, measurement is rarely noise free, so a final pass is made over the data to introduce noise.

As mentioned above, a test passes if it measures 4.9 volts or more and it fails if it measures less than 4.9 volts. Using various standard deviations, noise is introduced into the data in the following manner. If the result of a particular test is supposed to indicate a "pass" (a "0" in the data), a random voltage reading is generated with 5.0 mean and the specified standard deviation using a Gaussian distribution. If the resulting value is 4.9 or more, the test is kept as a pass. If it is lower than 4.9, the test result is changed to a "failure". Similarly, if the result of a particular test is supposed to indicate a failure, a random voltage reading is generated with 4.8 mean and the specified standard deviation. A resulting value of 4.9 or more changes the test result to a "pass". If it is lower than 4.9, it is kept as a failure. Using this process, the data generated above is "perturbed" with random noise. Standard deviations of 0.00 to 0.1 in 0.01 increments are used for a total of 11 different noise distributions.

Taking all possibilities into account, we ran experiments on three systems, ten assets, eight possible data set sizes, and 11 noise levels for a total of 2,640 data sets. Using this data, experiments were run to create NBCs for each of ten assets using asset-specific data for a particular system, *N*, and noise level as well as a composite NBC using aggregated data. This means that the composite NBC was trained and tested with 10*N* data whereas the asset-specific classifiers were each trained with *N*. This comports well with real world experience—if one had data for ten assets and had the option of creating ten classifiers or one aggregate classifier, one wouldn't throw 90% of the data away.

For all experiments, the NBC learning algorithm was repeated over 30 runs with 66% of the data used to train the NBC and 34% of the data used to test the NBC. New data was generated for each run. Where random selection is required, all randomization is stratified first by system (if necessary) and then by diagnosis (class). The

Dirichlet prior is set with p = 0.001% and m = 1. The intention with setting p so low is to make sure that the classification rule doesn't degenerate on the one hand but, on the other hand, the classification is not influenced. Choosing a diagnosis at random breaks all classification ties.

V. RESULTS

The results are presented for each system in tables of the following form. For each value of N and noise-level, there are 10 asset-specific classifiers based on segregated data and one composite classifier based on all of the data. The counts in each table are for the number of asset-specific classifiers that were as good as or better than the composite classifier. The asset-specific classifiers not in the counts are those with a statistically significant accuracy lower than the composite's accuracy at the 0.05 level using a test of difference of means. If more than five asset-specific classifiers are at least as good as the composite classifier then the count is bold face. If fewer than five are at least as good, then the count is shown in italics (Tables 2–4). Some of the tables only show the count of asset-specific classifiers that were better than the composite classifier, using the same significance test. Bold face and italics have the same interpretation (Tables 5–7).

The tables for relative accuracies are based on simple averages of the differential accuracy between the ten asset-specific classifiers and the composite classifier. If the difference is less than 0.0%, (i.e., if the set of asset-specific classifiers is less accurate on average than the composite classifier), then the percent is shown in italics. If the difference is greater than 0.0%, then the difference is shown in bold face (Tables 8–10).

Table 2 shows the "as good as" counts for the "one-for-one" system. As expected, all the asset-specific classifiers were at least as good as the composite classifier when the noise level was 0.00. In fact, all of the classifiers were 100.0% accurate. This accuracy continues up until a noise level of 0.03 when some of the asset-specific classifiers begin to lose accuracy and are no longer as good as or better than the composite. This can be seen in Table 5 where the "better than" counts are shown and in Table 8 where the difference in accuracy is 0.0% exactly.

However after noise level 0.03, more of the asset-specific classifiers begin to lose accuracy relative to the composite classifier. These are mostly for N = 50 and 100 with noise levels between 0.06 and 0.09. For most of the other combinations of N and noise, a majority of the specific classifiers are at least as good as the composite classifier. The patch of lowered accuracy, most visible in Table 5 is very close to what one would expect. It appears to show that for the "one-to-one" system, when N is relatively small and the noise is relatively high—but not too high—composite classifiers are better than a set of asset-specific classifiers.

It should be noted that when N = 25 with eight components and a uniform distribution, this translates into approximately three examples per fault for that asset-specific classifier. With a train/test split of 66/34, this means that two of the examples are used for training and one for testing. This is probably not the best train/test methodology for such a small N but we wanted to keep the methodology the same for all of the experiments.

As Table 3 shows, very similar results were obtained for the slightly more complicated "serial" system. As before, all of the classifiers are equally accurate—and 100% accurate—with no noise or low noise. There is a threshold point in the noise level after which some asset-specific classifiers become less accurate than the composite classifier. Looking at Table 6, however, where just those classifiers that were strictly better than the composite classifier are shown, an interesting pattern emerges. Leaving aside N = 25, for a given noise level, the number of asset-specific classifiers that are better than the composite classifier increases as N increases. What is interesting to note is that the lower the noise level, the larger N must be before the asset-specific classifiers are better than the composite classifier. Put a different way, the higher the noise level, the sooner, in terms of N, a set of asset-specific classifiers outperforms a single composite classifier created from aggregated data.

Finally, Table 4 displays the "as good as" results for the "functional" system. The functional system is arguably the most complicated of the three and is most like the type of D-matrix one might encounter in practice. Here again there are no surprises with no or almost no noise. All of the classifiers are 100% accurate and so there is no advantage for asset-specific classifiers. But once again, there is a point reached, noise = 0.06, where the larger N is, given a noise level, the more accurate the set of asset-specific classifiers is relative to the composite classifier. Table 7, which shows only those asset classifiers that were better than the composite classifier, tells a similar story.

Table 2. Number of Asset Specific Classifiers out of 10 that are as good or better than the Composite Classifier, One-For-One System

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	**10**	**10**	**10**	**10**	**9**	**10**	**8**	**8**	**7**	**7**	**7**
50	**10**	**10**	**10**	**10**	**10**	**9**	5	*4*	*3*	*4*	**8**
100	**10**	**10**	**10**	**10**	**10**	**8**	*2*	**6**	**7**	**10**	**10**
250	**10**	**10**	**10**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**
500	**10**	**10**	**10**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**
1000	**10**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**	**9**	**9**
2500	**10**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**	**9**	**9**
5000	**10**	**10**	**10**	**10**	**9**	**9**	**9**	**9**	**9**	**9**	**9**

Table 3. Number of Asset Specific Classifiers out of 10 that are as good or better than the Composite Classifier, Serial System.

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	**10**	**10**	**10**	**10**	**9**	**8**	**8**	**7**	**6**	**7**	**6**
50	**10**	**10**	**10**	**10**	5	5	*2*	*2*	*1*	*1*	*2*
100	**10**	**10**	**10**	**10**	**7**	*2*	*0*	*0*	*1*	*2*	5
250	**10**	**10**	**10**	**8**	*4*	*0*	*0*	*1*	*4*	**6**	**9**
500	**10**	**10**	**10**	**10**	**8**	*0*	*0*	*3*	**9**	**9**	**9**
1000	**10**	**10**	**10**	**10**	*1*	*3*	*4*	*2*	**9**	**10**	**9**
2500	**10**	**10**	**10**	**10**	*2*	**9**	**9**	**8**	**9**	**10**	**9**
5000	**10**	**10**	**10**	**10**	*4*	**9**	**9**	**9**	**9**	**9**	**9**

Table 4. Number of Asset Specific Classifiers out of 10 that are as good or better than the Composite Classifier, Functional System

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	**10**	**10**	**10**	**10**	**9**	**9**	**8**	**8**	5	5	**6**
50	**10**	**10**	**10**	**10**	**8**	**6**	5	*1*	*0*	*0*	*0*
100	**10**	**10**	**10**	**10**	**10**	**8**	*0*	*0*	*1*	*2*	*4*
250	**10**	**10**	**10**	**10**	**8**	*0*	*0*	*2*	5	**7**	**8**
500	**10**	**10**	**10**	**10**	**7**	*1*	*3*	**6**	**8**	**8**	**9**
1000	**10**	**10**	**10**	**10**	*3*	**6**	**7**	**7**	**8**	**9**	**9**
2500	**10**	**10**	**10**	**10**	*3*	**10**	**9**	**9**	**9**	**9**	**9**
5000	**10**	**10**	**10**	**10**	**7**	**9**	**9**	**9**	**9**	**9**	**9**

Table 5. Number of Asset Specific Classifiers out of 10 that are better than the Composite Classifier, One-For-One System

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	*0*	*0*	*0*	*0*	*0*	*3*	5	*3*	*4*	*3*	*4*
50	*0*	*0*	*0*	*3*	*0*	*1*	*0*	*0*	*0*	*0*	*0*
100	*0*	*0*	*0*	*2*	*0*	*0*	*0*	*0*	*0*	*1*	*3*
250	*0*	*0*	*0*	*2*	*2*	*0*	*0*	*0*	*2*	*2*	**6**
500	*0*	*0*	*0*	*2*	*1*	*0*	*0*	*2*	*3*	**8**	**9**
1000	*0*	*0*	*0*	*3*	*1*	*1*	*1*	*2*	**7**	**9**	**9**
2500	*0*	*0*	*0*	**8**	*0*	5	**7**	5	**8**	**9**	**9**
5000	*0*	*0*	*0*	5	*2*	**8**	**9**	**9**	**9**	**9**	**9**

Table 6. Number of Asset Specific Classifiers out of 10 that are better than the Composite Classifier, Serial System.

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	*0*	*0*	*0*	*0*	*0*	*3*	5	*3*	*4*	*3*	*4*
50	*0*	*0*	*0*	*3*	*0*	*1*	*0*	*0*	*0*	*0*	*0*
100	*0*	*0*	*0*	*2*	*0*	*0*	*0*	*0*	*0*	*1*	*3*
250	*0*	*0*	*0*	*2*	*2*	*0*	*0*	*0*	*2*	*2*	**6**
500	*0*	*0*	*0*	*2*	*1*	*0*	*0*	*2*	*3*	**8**	**9**
1000	*0*	*0*	*0*	*3*	*1*	*1*	*1*	*2*	**7**	**9**	**9**
2500	*0*	*0*	*0*	**8**	*0*	5	**7**	5	**8**	**9**	**9**
5000	*0*	*0*	*0*	5	*2*	**8**	**9**	**9**	**9**	**9**	**9**

Table 7. Number of Asset Specific Classifiers out of 10 that are better than the Composite Classifier, Functional System.

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	*0*	*0*	*0*	*0*	*3*	*3*	4	*2*	*2*	*2*	*4*
50	*0*	*0*	*0*	*1*	*0*	*1*	*0*	*0*	*0*	*0*	*0*
100	*0*	*0*	*0*	*4*	*2*	*1*	*0*	*0*	*0*	*1*	*2*
250	*0*	*0*	*0*	*3*	*1*	*0*	*0*	*0*	*1*	*2*	**6**
500	*0*	*0*	*0*	*2*	*1*	*0*	*0*	*3*	*3*	**7**	**7**
1000	*0*	*0*	*0*	5	*0*	*2*	*4*	4	5	**8**	**8**
2500	*0*	*0*	*0*	**8**	*2*	**8**	**8**	**8**	**8**	**8**	**9**
5000	*0*	*0*	*0*	**6**	*2*	**8**	**8**	**8**	**8**	**9**	**9**

Table 8. Average Differential Accuracy of all Asset Specific Classifiers compared to the Composite Classifier, One-For-One System

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	0.0%	0.0%	0.0%	**0.1%**	**0.5%**	**3.8%**	**3.8%**	**4.6%**	**2.5%**	**3.0%**	**4.1%**
50	0.0%	0.0%	0.0%	**0.1%**	**0.2%**	**0.2%**	*–2.8%*	*–4.9%*	*–5.5%*	*–3.9%*	**0.2%**
100	0.0%	0.0%	0.0%	**0.0%**	**1.0%**	*–0.4%*	*–4.1%*	*–4.4%*	*–0.4%*	**2.1%**	**4.5%**
250	0.0%	0.0%	0.0%	**0.0%**	**0.9%**	**1.0%**	**2.2%**	**2.7%**	**3.9%**	**4.9%**	**6.1%**
500	0.0%	0.0%	0.0%	**0.1%**	**0.6%**	**1.5%**	**4.8%**	**7.0%**	**7.7%**	**8.3%**	**8.3%**
1000	0.0%	0.0%	0.0%	**0.1%**	**0.4%**	**2.3%**	**4.8%**	**7.3%**	**9.0%**	**9.8%**	**10.2%**
2500	0.0%	0.0%	0.0%	**0.1%**	**0.6%**	**2.6%**	**4.9%**	**7.3%**	**8.7%**	**9.8%**	**10.7%**
5000	0.0%	0.0%	**0.0%**	**0.0%**	**0.7%**	**2.5%**	**4.9%**	**7.1%**	**8.8%**	**9.8%**	**10.6%**

Table 9. Average Differential Accuracy of all Asset Specific Classifiers compared to the Composite Classifier, Serial System.

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	0.0%	0.0%	0.0%	*–0.1%*	*–0.5%*	**0.2%**	**2.1%**	**0.3%**	**0.8%**	*–0.6%*	**2.0%**
50	0.0%	0.0%	0.0%	*–0.1%*	*–1.5%*	*–2.0%*	*–7.3%*	*–9.5%*	*–10.2%*	*–8.4%*	*–6.7%*
100	0.0%	0.0%	*0.0%*	*–0.1%*	*–0.8%*	*–3.1%*	*–7.4%*	*–9.5%*	*–7.8%*	*–4.8%*	*–1.7%*
250	0.0%	0.0%	0.0%	*–0.2%*	*–1.7%*	*–6.4%*	*–7.5%*	*–4.6%*	*–1.9%*	**0.5%**	**2.6%**
500	0.0%	0.0%	0.0%	**0.0%**	*–0.2%*	*–3.9%*	*–4.3%*	*–1.5%*	**1.0%**	**3.2%**	**5.0%**
1000	0.0%	0.0%	0.0%	**0.0%**	*–0.5%*	*–0.7%*	*–1.0%*	*–0.6%*	**1.9%**	**4.2%**	**6.2%**
2500	0.0%	0.0%	0.0%	**0.1%**	*–0.4%*	**0.4%**	**0.9%**	**0.9%**	**2.0%**	**4.4%**	**6.5%**
5000	0.0%	0.0%	0.0%	**0.0%**	*–0.1%*	**0.6%**	**1.2%**	**1.6%**	**2.2%**	**4.4%**	**6.4%**

Table 10. Average Differential Accuracy of all Asset Specific Classifiers compared to the Composite Classifier, Functional System.

	Noise (Standard Deviation)										
N	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.10
25	0.0%	0.0%	0.0%	**–0.0%**	*–0.1%*	**1.5%**	**2.1%**	**0.3%**	*–4.3%*	*–3.6%*	*–3.3%*
50	0.0%	0.0%	0.0%	*–0.2%*	*–0.8%*	*–0.9%*	*–4.9%*	*–9.7%*	*–11.2%*	*–10.9%*	*–9.4%*
100	0.0%	0.0%	*0.0%*	*–0.0%*	**0.2%**	*–1.3%*	*–6.2%*	*–8.6%*	*–8.1%*	*–5.6%*	*–3.9%*
250	0.0%	0.0%	0.0%	*–0.1%*	*–0.2%*	*–4.0%*	*–5.0%*	*–3.1%*	*–1.5%*	*–0.4%*	**1.6%**
500	0.0%	0.0%	0.0%	**0.0%**	*–0.2%*	*–2.5%*	*–2.2%*	*–0.1%*	**2.3%**	**3.0%**	**3.9%**
1000	0.0%	0.0%	0.0%	**0.0%**	*–0.4%*	*–0.4%*	**0.2%**	**1.2%**	**2.9%**	**4.2%**	**5.2%**
2500	0.0%	0.0%	0.0%	**0.1%**	*–0.3%*	**0.6%**	**1.5%**	**2.1%**	**3.1%**	**4.6%**	**5.7%**
5000	0.0%	0.0%	**0.0%**	**0.0%**	**0.0%**	**0.8%**	**1.7%**	**2.4%**	**3.1%**	**4.5%**	**5.9%**

This does not, however, seem to be the end of the story. As the systems have become more complicated another interesting pattern has emerged. First, at low levels of noise, 0.00 to 0.02, no classifier is better than any other classifier. Second, a transitional range is entered where some asset-specific classifiers are more accurate than the composite classifier and, depending on the noise, some asset-specific classifiers are less accurate than the composite classifier. This noise range is approximately 0.03 to 0.05. Finally, a third range of noise shows a pattern of increasing accuracy for asset-specific classifiers as *N* increases but increased accuracy is attained sooner, the more noise the data has.

VI. DISCUSSION

The purpose of these experiments was to test the hypothesis that a set of asset-specific classifiers would be more accurate, on average, than a single classifier created from all of the data. The results reported clearly show that this is *sometimes* the case. If there is no or low noise in the data, it appears that whether one uses a set of asset-specific classifiers or a single composite classifier doesn't matter for accuracy because they should all have approximately the same accuracy. Even for noise level 0.04, the average accuracy differential only ranged from -0.8% to 0.0%, which was not statistically significant.

On the other hand, when there was "medium" noise, it was difficult to predict how much data would be required before the set of classifiers would outperform the single classifier, especially for complicated systems. Except for the largest datasets, even though some of the asset-specific classifiers were better than the composite classifier, overall the accuracy was still less than that of the composite classifier.

At "high" levels of noise, the set of asset-specific classifiers were definitely more accurate than the composite classifier provided enough data was available. The lesson here seems to be that the noisier the data, the sooner asset-specific classifiers will be beneficial as long as the data is noisy *enough*.

Looking at the largest N, however, it appears to be clear that given enough data, no matter what the noise level, a set of asset-specific classifiers will always be at least as good as a composite classifier built from the same data and possibly better, the higher the noise level. All of the other trends merely suggest that under certain circumstances one can take advantage of this increased accuracy earlier (with smaller *N*).

These experiments appear to reveal a threshold of data that must be reached for asset-specific models to be beneficial. As a practical matter, specific assets do not fail everyday—in fact, the most commonly observed state of affairs is "no failure". Even for the smallest *N* of 25 used here, it may take some time to gather sufficient data to even determine if the assets have significantly different failure distributions and thus warrant asset-specific classifiers. The question naturally arises as to what should be done until there is enough data. Although the experiments in this paper examined the broad outlines of what might happen, the practical application of the results will have to be determined in future work.

VII. FUTURE WORK

Given enough data, a set of asset-specific classifiers is more accurate than a single composite classifier. Until there is *enough* data, however, the results are mixed. If the data has no or low noise, aggregation/segregation doesn't matter because the accuracy of the classifiers is going to be the same. If the noise is high, then there is some advantage to be gained by using asset-specific classifiers sooner rather than later. Even before an *N* of 50 or even 100, however, is there a way to take advantage of these findings?

One place to start looking is in the specification of the *Dirichlet prior*. For the experiments in this paper, the Dirichlet prior was set to 0.001%—just enough to prevent the classification rule from breaking but not enough to affect classification. An improved approach might use Dirichlet priors to actually affect classification in an intelligent way. For example, for the current set of experiments, the D-matrix for each system is known. Given a historical distribution of diagnoses, the D-matrix could be used to calculate a noise-free set of prior probabilities. If these priors were then used during the training of the classifier, accuracy might be improved. If no D-matrix is available or if it is unknown, expert knowledge could be substituted to estimate priors and possibly even the network. This could also provide an intelligent starting point for training the network. Either approach could be used in the beginning to create the composite classifier and both will be investigated in future work.

The next step would be to use the composite classifier for the prior probabilities for the individual asset-specific classifiers. In the limit, as

more and more data is collected, the composite classifier would be updated into the respective asset-specific classifiers, depending on which asset data was used for training. The main challenge here will be to arrive at an appropriate update rule. For example, if the probabilities from the composite classifier were used as the priors, the question is what *m* would be the best to maintain good performance when *N* is low (and the prior *p* should dominate) when *N* is high (and the training should dominate). This will also be investigated in future work.

VIII. CONCLUSION

This paper started out by describing the DoD's IUID program. The eventual goal of the research of which this paper is but a part is to use the IUID based data to build effective diagnostic and prognostic models. It was hypothesized that asset-specific modeling would be a good step in the direction of reaching that goal. To test the hypothesis in a well-controlled way, experiments were conducted and the results were reported.

In general, the hypothesis was well supported. A set of asset-specific classifiers was eventually more accurate than a single composite classifier given a large enough *N*. However, it was determined that "enough data" was not the only factor in determining if a set of specialized classifiers was more accurate than a single general classifier. System complexity and noise level both had an influence on relative accuracies between the two. Additionally, it was determined that the trends observed in the experiments were limiting cases. In practice, it would take quite some time to collect "enough data."

Nevertheless, the approach appears to be promising. For future work it was suggested that experiments be conducted to improve estimation of the *Dirichlet priors* from either a known D-Matrix or expert knowledge for the composite classifier. The composite classifier could then serve as the Dirichlet prior while training asset-specific classifiers. In that case the emphasis would be on finding an effective weight for the prior.

REFERENCES

[1] Department of Defense Guide to Uniquely Identifying Items Assuring Valuation, Accountability and Control of Government Property, Version 1.5, June 7, 2005.

[2] Duda, R., Hart, P., and Stork, D., *Pattern Classification*, New York: John Wiley & Sons, 2001.

[3] Friedman, N., Geiger, D., and Goldszmidt, M., "Bayesian Network Classifiers," *Machine Learning*, 29:131–163, 1997.

[4] Government Accounting Office Report NSAID-92-258, "Operation Desert Storm, Lack of Accountability Over Materiel during Redeployment", September 1992.

[5] Kononenko, I., "Semi-naïve Bayesian Classifier," In Y. Kodratoff (ed.), *Proceedings of the Sixth European Working Session on Learning*, Berlin: Springer-Verlag, 1991, pp. 206–219.

[6] Langley, P., Iba, W., and Thompson, K., "An Analysis of Bayesian Classifiers," *Proceedings of the Tenth National Conference on Artificial Intelligence*, San Mateo, CA: AAAI Press, 1992, pp. 223–228.

[7] Mitchell, T. *Machine Learning*, New York: The McGraw-Hill Companies, 1997.

[8] Sheppard, J., and Butcher, S., "On the Linear Separability of Diagnostic Models," to appear in *AUTOTESTCON Conference Record*, New York: IEEE Press, 2006.

[9] Sheppard, J., Butcher, S., Kaufman, M., and MacDougall, C., "Not-So-Naïve Bayesian Networks and Unique Identification in Developing Advanced Diagnostics," *Proceedings of the IEEE Aerospace Conference*, New York: IEEE Press, March 2006.

[10] Sheppard, J. and Kaufman, M., "A Bayesian Approach to Diagnosis and Prognosis Using Built In Test," *IEEE Transactions on Instrumentation and Measurement*, Special Section on Built-In Test, Vol. 54, No. 3, June 2005, pp. 1003–1018.

[11] Simpson, W. and Sheppard, J., *System Test and Diagnosis*, Norwell, MA: Kluwer Academic Publishers, 1994.

[12] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Guidelines for the Virtual Unique Item Identifier (UII) Version 1.0, December 29, 2004.

[13] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Memorandum, Radio Frequency Identification (UID) Policy, July 30, 2004.

[14] Under Secretary of Defense (Acquisition, Technology and Logistics) (OUSD(AT&L)) Memorandum, Update for Policy for Unique Identification (UID) of Tangible Items, September 3, 2004.