ν-Anomica: A Fast Support Vector based Novelty Detection Technique

Santanu Das*, Kanishka Bhaduri†, Nikunj C. Oza‡ and Ashok N. Srivastava§
NASA Ames Research Center, Moffett Field, CA 94035
*UARC, UC Santa Cruz, Santanu.Das-1@nasa.gov
†MCT Inc., Kanishka.Bhaduri-1@nasa.gov
‡NASA Ames Research Center, Nikunj.C.Oza@nasa.gov
§NASA Ames Research Center, Ashok.N.Srivastava@nasa.gov

***Abstract*—In this paper we propose ν-Anomica, a novel anomaly detection technique that can be trained on huge data sets with much reduced running time compared to the benchmark one-class Support Vector Machines algorithm. In ν-Anomica, the idea is to train the machine such that it can provide a close approximation to the exact decision plane using fewer training points and without losing much of the generalization performance of the classical approach. We have tested the proposed algorithm on a variety of continuous data sets under different conditions. We show that under all test conditions the developed procedure closely preserves the accuracy of standard one-class Support Vector Machines while reducing both the training time and the test time by $5 - 20$ times.**

***Keywords*-Anomaly Detection; Support Vector Machines; Kernel; Optimization;**

I. Introduction

Outlier or anomaly detection refers to the task of identifying abnormal or inconsistent patterns from a dataset. While they may seem to be undesirable entities, identifying them has many potential applications in fraud and intrusion detection, financial market analysis, medical research and safety-critical vehicle health management. Broadly speaking, outliers can be detected using either *supervised* or *semi-supervised* or *unsupervised* techniques [13] [5]. Unsupervised techniques, as the name suggests, do not require labeled instances for detecting outliers. In this category, the most popular ones are the distance-based and density based techniques. The basic idea of these techniques is that outliers are points in low density regions or those which are far from other points. In their seminal work, Knorr *et al.* [15] proposed a distance-based outlier detection technique based on the idea of nearest neighbors. The naive solution has a quadratic time complexity since every data point needs to be compared to every other to find the nearest neighbors. To overcome this, researchers have proposed several techniques such as the work by Angiulli and Pizzuti [1], Ramaswamy *et al.* [17], and Bay and Schwabacher [2]. Density-based outlier detection schemes, on the other hand, flag a point as an outlier if the point is in a low density region. The density of a point can be evaluated using several techniques such as the ones proposed in [12]. Supervised techniques require labeled instances of both normal and abnormal operation data for first building a model (*e.g.* a classifier) and then testing if an unknown data point is a normal one or an outlier. The model can be probabilistic such as Bayesian inference [9] or deterministic such as decision trees, Support Vector Machines (SVMs) and neural networks [14]. Semi-supervised techniques only require labeled instances of normal data. Hence they are more widely applicable than the fully supervised ones. These techniques build models of normal data and then flag as outliers all those points which do not fit the model.

Since this paper proposes a variant of unsupervised anomaly detection technique using support vector machines, we discuss more about this here. Support vector machines [21] [7] have been widely used for classification and regression. While the original idea of using SVM has been around for many years, recent interest has been kindled by the need for analyzing large datasets. Fehr *et al.* [10] presents a scheme for efficient learning of SVMs based on the intuition that most of the training time for non-linear SVMs is wasted in evaluating the kernel matrix. In their approach, they approximate a single SVM using a collection of simpler linear SVMs. Each of these simpler ones can be trained and tested in constant time, leading to low running time without any loss of accuracy. Such a construction can be viewed as a tree in which any intermediate node represents a hyper-plane and the leaf nodes correspond to pure labels of one class type.

Burges and Schölkopf [4] present a different technique for speeding up SVMs. Let

$$\Psi = \sum_{j=1}^{N_s} \alpha_j y_j \Phi(\mathbf{s}_j)$$

be the normal to the decision surface where α_j's denote

the Lagrange multipliers corresponding to the support vectors $\mathbf{s}_j$, y_j denotes the true class labels, $\Phi(\cdot)$ denotes the kernel function, and N_s denotes the number of support vectors. This computation scales linearly with the number of support vectors. To achieve speedup, the authors propose to approximate the normal using fewer support vectors (N_z) as,

$$\Psi^{'} = \sum_{j=1}^{N_z} \alpha_j y_j \Phi(\mathbf{s}_j).$$

The goal is then to minimize the L2-norm of the two normal vectors

$$\rho = \left\| \Psi - \Psi^{'} \right\|.$$

As has been shown in [4], there exists nontrivial values of $\Psi^{'}$ which ensures $\rho \neq 0$.

The work most closely related to this one is the reduced support vector machine (RSVM) idea presented in [16] and [6]. In these, an initial SVM is trained not on the entire training set, but rather on a subset of the training set called the active training set. Then, the SVM is evaluated on a validation set. If the accuracy is acceptable, the algorithm converges, else a set of misclassified points are selected from the remaining training set and added to the active training set. The approach in [6] first sorts the misclassified points according to their scores on the validation set and then divides the points into equal size subsets. When additional points are needed, it selects new points from each subset. In our approach we do not sort the points and thereby achieve lower running time.

The proposed *ν-Anomica* algorithm is faster than the standard benchmark one class SVMs while preserving the accuracy. It achieves this by developing the hyperplane in an incremental fashion. We show that, in many cases, *ν-Anomica* has similar prediction accuracy compared to classical one class SVM while reducing the running time dramatically. Our main contributions in this paper are:

- We propose a variant of one class SVM-based novelty detection algorithm called *ν-Anomica* with improved running time while retaining the accuracy of standard one-class SVMs.
- We demonstrate the capability of the algorithm in handling huge sizes of training data (both instances and attributes).
- We measure the performance of the proposed technique using different metrics, such as accuracy, sensitivity, and run time.
- We provide some useful insights regarding the effectiveness of proposed technique based on the experimental evaluation.

II. Novelty Detection with One Class SVMs

One class SVMs, an unsupervised learning method for estimating the density of the target support objects was introduced by Schölkopf [18]. Throughout this paper, we have considered positive labeled data points as normal and negative label data points as outliers. The model consists of a parameter ν that denotes the maximum allowance of outliers in the training data. The idea is to draw a separating hyperplane that can separate these outliers from the rest of the training examples, as shown in Fig. 1. Unlike the $2-$class SVMs classifier, in one class SVMs model, the separating hyperplane is constructed using positive labeled training data set only. Since a $N-1$ dimensional hyperplane can exist in the N-dimensional feature space, the primary task is to find the optimal separating plane that maximizes the margin between the hyperplane and the origin, which is the lone representative of the second class with negative label.

A. The Model

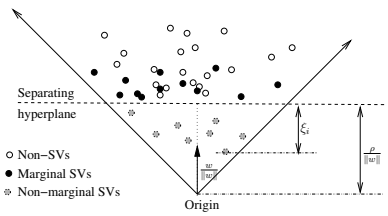


Figure 1. This figure illustrates the geometric interpretation of optimal hyperplane for one class SVMs.

We assume a set of labeled training data $\mathcal{D} = \{(\vec{x}_i)\}_{i=1}^{n}$ in the input space $\mathbb{R}$, where $\vec{x}_i \in \mathbb{R}^d$. We further assume that there exists a function ϕ that can be used to map variables from the input space to the feature space $\mathcal{F}$, i.e. $\phi : \mathbb{R}^d \rightarrow \mathcal{F}$. In feature space the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ property, where $\mathbf{x}_i := \phi(x_i)$ holds. Also Cover's theorem [21] states that nonseparable or nonlinearly separable features in the input space $\mathbb{R}$ is more likely to be linearly separable in the feature space $\mathcal{F}$, provided the transformation $\phi(.)$ is nonlinear and the dimensionality of the feature space is high enough. While evaluating the dot product in the feature space, the explicit calculation using ϕ can be avoided by simply evaluating the kernel function i.e. $k(x_i, x_j) := \langle \phi(x_i), \phi(x_j) \rangle$. However in order for this to hold, this the chosen inner-product kernel must satisfy Mercer's theorem [3]. For the majority of this paper, we

have used Radial Basis Function (RBF) kernel (Eqn. 1) that evaluates the distances between data points as,

$$k(\vec{x_i}, \vec{x_j}) = e^{\frac{-||\vec{x_i} - \vec{x_j}||^2}{2\sigma^2}} \tag{1}$$

where $||.||$ denotes the Euclidean norm and σ defines the kernel width.

Schölkopf [18] showed that in the high dimensional feature space it is possible to construct an optimal hyperplane by maximizing the margin between the origin and the hyperplane in the feature space by solving the following primal optimization problem,

$$\begin{aligned} \text{minimize} \quad & P(\mathbf{w}, \rho, \xi_i) = \frac{1}{2} w w^T + \frac{1}{\nu \ell} \sum_{i=1}^{\ell} \xi_i - \rho \\ \text{subject to} \quad & (\mathbf{w}.\phi(x_i)) \geq \rho - \xi_i, \quad \xi_i \geq 0, \quad \nu \in [0, 1] \end{aligned} \tag{2}$$

where ν is an user specified parameter that defines the upper bound on the training error, and also the lower bound on the fraction of training points that are support vectors, ξ is the non-zero slack variable, ρ is the offset, $\phi(x_i)$ represents the transformed image of x_i in the Euclidean space and $i \in [\ell]$. Throughout this study, we will use the scaled version [8] of the dual problem which takes the form of,

$$\begin{aligned} \text{minimize} \quad & Q = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) + \rho \left(\ell \nu - \sum_i \alpha_i \right) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq 1, \quad \nu \in [0, 1] \end{aligned} \tag{3}$$

where α_i and β_i are Lagrangian multipliers. The optimal solution must satisfy the exact Karush-Kuhn-Tucker (KKT) conditions which can be summarized as,

$$\begin{array}{ccc} \alpha_i = 1 & g(\vec{x_i}) < \rho & \xi_i > 0 \\ 0 < \alpha_i < 1 & g(\vec{x_i}) = \rho & \xi_i = 0 \\ \alpha_i = 0 & g(\vec{x_i}) > \rho & \xi_i = 0 \end{array} \tag{4}$$

where $g(\vec{x_j}) = \sum_i \alpha_i k(x_i, x_j)$. The value of the ρ can be recovered from the constraint of the primal problem by exploiting the solution w and pattern x_i corresponding to $0 < \alpha_i < 1$ while setting $\xi_i = 0$ under equality condition. There exist at least $\nu\ell$ training points with non-zero Lagrangian multipliers $(\vec{\alpha})$ and these points $\{x_i : i \in [\ell], \alpha_i > 0\}$ are called support vectors. Let $\mathcal{I}_0 = \{i \ : \ \alpha_i = 0\}$, $\mathcal{I}_m = \{i \ : \ 0 < \alpha_i < 1\}$ and $\mathcal{I}_{nm} = \{i \ : \ \alpha_i = 1\}$ be the set of indices of Lagrangian multipliers corresponding to non-SVs, marginal and non-marginal support vectors respectively. Once $\vec{\alpha}$ is known, SVMs compute the following decision function.

$$f(\vec{x_j}) = sign\left(\sum_{i \in \mathcal{I}_m} \alpha_i k(\vec{x_i}, \vec{x_j}) + \sum_{i \in \mathcal{I}_{nm}} k(\vec{x_i}, \vec{x_j}) - \rho \right) \tag{5}$$

If the decision function predicts a negative label for a given test point x_j, this implies that the test point is classified as outlier. Test examples with positive labels are considered as normal.

B. Virtual Decision Surface

The decision boundary is defined by a normal vector $\mathbf{w}$ (also referred as weight vector) is orthogonal to the plane and an offset ρ. All points $\mathbf{x}$ lying on this hyperplane must satisfy $g(\mathbf{x}) - \rho = 0$ where $\{g(\mathbf{x}) = \mathbf{w}.\mathbf{x}, \quad \forall \mathbf{w} \in \mathcal{F}\}$. Since the weight vector is a weighted sum of the features corresponding to the support vectors, one may be motivated to define two normal vectors ω and λ both perpendicular to the decision plane such that,

$$\gamma_n = \frac{\omega}{\|\omega\|} = \frac{\lambda}{\|\lambda\|}. \tag{6}$$

where γ_n is the unit normal along ω and λ. It is not too difficult to prove that,

$$\frac{g_\omega(z)}{g_\lambda(z)} = \frac{\|\omega\|}{\|\lambda\|} = \frac{\omega_0}{\lambda_0} \tag{7}$$

where ω_0 and λ_0 are the offset terms corresponding to normal vectors ω and λ. This is because the distance of the hyperplane from the origin remains unchanged i.e. $\frac{\omega_0}{\|\omega\|} = \frac{\lambda_0}{\|\lambda\|}$. An important conclusion is that for a fixed test point z, the ratio of the decision values evaluated using two different normal vectors (defined by two different sets of points) orthogonal to the same hyperplane is constant. This can further be expressed as,

$$\frac{f_\alpha(\vec{z})}{f_\beta(\vec{z})} = \frac{\sum_{\substack{i \in \mathcal{I}_m, \mathcal{I}_{nm} \\ j \in Z}} \alpha_i k(\vec{x_i}, \vec{z}) - \rho_\alpha}{\sum_{\substack{i \in \hat{\mathcal{I}}_m, \hat{\mathcal{I}}_{nm} \\ j \in Z}} \beta_i k(\vec{x_i}, \vec{z}) - \rho_\beta} = \eta \tag{8}$$

where η is a constant, $f_\alpha(\vec{z})$ and $f_\beta(\vec{z})$ are the decision functions (Eqn. 5) expressed in terms of Support Vectors corresponding to Lagrange's multiplier α_i and β_i. The fact that members of $\mathcal{I}_m \cup \mathcal{I}_{nm}$ and $\hat{\mathcal{I}}_m \cup \hat{\mathcal{I}}_{nm}$ may differ in number leads to the fact that the construction of the weight vector does not depend on the number of support vectors. It is well known that the positive semidefiniteness of the dual problem may result in redundant support vectors which defines the normal

vector. This means that some of the support vectors are a linear combination of other support vectors and implies that the removal of some of these linearly dependent support vectors will not change the hyperplane. In previous work [4], Burges and Schölkopf pointed out that the solution of the SVMs may not be the sparsest one and suggested ways of approximating the solution using virtual Support Vectors. For one-class SVMs, the existence of the parameter ν may be the source that introduces redundancies in the solution because it leads to a minimum required number of support vectors. In this research we are motivated to develop a scheme that searches for a reduced set of the transformed features in $\mathcal{F}$ which is sufficiently close to approximate the normal vector of the exact solution of one-class SVMs and thus retaining the same accuracy with lower running time.

III. Proposed Approach: ν-Anomica

ν-Anomica proposes an approximate solution that permits one-class SVMs to train on huge data sets in much reduced time. The main idea of this algorithm is to start with an initial "feasible solution" of classical one-class model trained on a very reduced data set and guide the current solution towards the "target solution". Here the solution of the optimal hyperplane from the exact solution is set as the target. To achieve this goal, a controlled updating of the existing training pool with new examples in an iterative fashion has been adopted. In order to select the appropriate subset of new examples, we propose a two stage strategy. In the first step, we ensure that at each iteration the solution of the most updated model is along the direction of the optimal solution. Secondly, at each step the number of new members which control the step length is decided based on some model feedback. The work presented here exploits the fact that the ν parameter of one-class SVMs plays a very important role in defining the highest allowable fraction of misclassification of the training data. This means the one-class model, once built, should be able to correctly classify $1-\nu$ fraction of the entire training set as normal examples. For the rest of the paper we will refer to this as the "ν-criterion". Any newly developed model (based on a subset of the entire data set) which is a close approximation to the exact solution is bound to meet the "ν-criterion". Such a data set can be considered as a representative working set.

In the following, we will demonstrate the core idea of the proposed algorithm in steps. The ν-Anomica algorithm (Algorithm 1) starts with the assumption that two non overlapping data sets have been randomly chosen from the same distribution. One of these two sets was assigned for training purpose while the second set was kept for validation purpose. The model also assumes that the optimal value of the kernel parameter σ (Eqn. 1) has already been evaluated for a fixed ν. Under this condition, if a standard one-class model is successfully built on the entire training set, the model should satisfy the "ν-criterion".

Algorithm 1 Anomica

Argument:
Let the training set be $X = \{x_1, x_2....x_p\}$, $X \in \mathcal{R}^d$. Let X_1 be a chosen randomly chosen subset of X i.e. $X_1 = \{x_1, x_2, ..x_\ell\} \subset X$, where $\ell << p$ and $X_2 = X \backslash X_1$. Let $Z = \{z_1, z_2....z_r\}$, $Z \in \mathcal{R}^d$ be the validation set such that $X \cap Z = \emptyset$, where $\emptyset$ corresponds to null vector.
Notations: I represents indices.
Input: X_1, X_2, Z, σ and $\nu \in \{0, 1\}$.
Output: Lagrangian multipliers (α_i^*), Support Vectors (SVs^*) and Bias (ρ^*)
Initialization: Variable $I_{neg}^r = \emptyset$ and $I_{pos}^r = \emptyset$;

Step A: Compute α_i^* and ρ^* by minimizing

$$\frac{1}{2}\sum_{i,j\in X_1}\alpha_i\alpha_j k\left(x_i, x_j\right) + \rho\left(\ell\nu - \sum_{i\in X_1}\alpha_i\right)$$
$$\text{subject to} \quad 0 \le \alpha_i \le 1, \quad \nu \in [0,1], \quad i \in X_1$$

Step B: Obtain classification rate C_r^z on Z.

$$\mathcal{M} = \{ \quad z_m : m \in [r], f(\vec{z_m}) < 0\}$$
$$C_r^z = 1 - \frac{1}{N}\sum_{n=1}^{N} I\left(\mathcal{M}\right)$$

Step C: Check objective $E_r \approx C_r^z - (1-\nu)$.
Step D: $[\alpha_i^*, SVs^*, \rho^*]$=UpdateMember($E_r, X_1, X_2, Z$).

In the proposed technique, we start by randomly selecting a small subset from the entire training set and using this small subset to develop the initial One-Class SVMs model. Once the SVs are obtained, we validate the resulting model on the validation set. Since the current model is based on a very small subset of the entire training set, the classification accuracy of the model may not satisfy the "ν-criterion" on the hold out set. This is based on the fact that a correct model should be able to achieve the same level of classification accuracy (in this case $1-\nu$ because of "ν-criterion") on a hold out set which has been generated from a similar distribution to that of the training set. Here it is important to note that the proposed algorithm uses the "ν-criterion" as the target classification rate.

If the classification rate on the validation set is greater than $(1-\nu)$, it means that either the small subset of the training set has fewer positive examples or that the data points corresponding to the support vectors of this model are not good representative of the positive

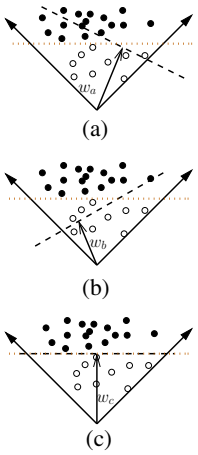


Figure 2. This figure shows the update rules of ν-Anomica. Subfigures (a) and (b) represent the over classified and under classified cases respectively. In subfigure (c) the evaluated classification rate of the current model meets the "ν-criterion". The target hyperplane and the current hyperplane is represented by dotted and dashed line respectively.

examples. This is analogous to saying that the most recently evaluated support vectors have defined a normal vector ($\mathbf{w}$) corresponding to a hyperplane (Fig. 2-b) that predicts too many positive members in the hold out set and thus does not satisfy the ν-criterion. Similarly, if classification rate is less than $(1-\nu)$, it implies that the current working set has too few negative examples (Fig. 2-a). Hence there is a necessity to update the initial working set with additional positive or negative examples only when any of the above two situations arises. Pseudo code of our algorithm for doing this is shown in Algorithm 2. This procedure is repeated until the ν-criterion is satisfied or close to being satisfied on the hold out set. The number of examples (positive or negative) to be selected from the entire remaining set is governed by a penalized weight function as shown in line 5 of the pseudo code (Algorithm 2), based on deviation of the classification rate on the validation set from the target $(1-\nu)$. Once the ν-criterion on the hold out set is satisfied (Fig. 2-c), the algorithm meets the stopping criterion, and hence terminates.

A. How does the ν-criterion influence the model?

We will further illustrate the role of "ν-criterion" by using a synthetic "*one class*" data set. The data set consists of samples drawn from a d-dimension Gaussian distribution with user specified mean (μ) and covariance (Σ). For simplicity we will use a $2-$dimensional data set drawn from a single distribution. We have chosen a linear kernel in the SVMs model to do the mapping.

Algorithm 2 UpdateMember(E_r, X_1, X_2, Z)

1: Let the operator $\diamond$ can take either $>$ or $<$ but one at a time.
2: **while** $E_r \neq 0$ **do**
3: **while** $E_r \diamond 0$ **do**
4: $I_{index}^{interest} \leftarrow I_{index}^{X_2}(S_{X_2} \diamond 0)$
5: Set k_1=length($I_{index}^{interest}$) and penalized weight $k_2 = k_1 \frac{|E_r|}{1-\nu}$
6: Randomly select $I_{index}^{k_2}$ indices from the possible k_1 indices
7: Update indices $I_{index}^{*} \leftarrow I_{index}^{interest}(I_{index}^{k_2})$
8: $X_1 \leftarrow \{X_1 \bigcup X_2(I_{index}^{*})\}$
9: $X_2 \leftarrow \{X_2 \backslash X_2(I_{index}^{*})\}$
10: Compute α_i^* and ρ^* by minimizing

$$\frac{1}{2}\sum_{i,j\in X_1}\alpha_i\alpha_j k\left(x_i,x_j\right)+\rho\left(\ell\nu-\sum_{i\in X_1}\alpha_i\right)$$
$$\text{subject to}\quad 0\leq\alpha_i\leq 1,\quad \nu\in[0,1],\quad i\in X_1$$

11: Evaluate decision function,

$$S_{X_2}=sign(\sum_{\substack{i\in\mathcal{I}_m,\mathcal{I}_{nm}\\ j\in X_2}}\alpha_i k(\vec{x_i},\vec{x_j})-\rho)$$

12: Obtain classification rate C_r^z on Z.

$$\mathcal{M}=\{\quad z_m : m\in[r]\,, f(\vec{z_m})<0\}$$
$$C_r^z=1-\frac{1}{N}\sum_{n=1}^{N}I\left(\mathcal{M}\right)$$

13: Check objective $E_r \approx C_r^z - (1-\nu)$.
14: **if** $E_r \approx 0$ **then**
15: Return
16: **end if**
17: **end while**
18: **end while**

With a fixed number of instances, the redundancies in the data set were controlled by varying the covariance of the distribution. In the first run, two data sets each of 1001 instances were generated from a distribution with same mean (0.001) but with two very different covariances. For one set the covariance was set to "machine precision" (eps) which is the minimum allowable spacing between two floating point numbers and $10^{20}\times$eps for the other set. The outcome of the One class SVMs model (with $\nu = 0.1$) on these two data sets has been summarized in Table I. It can be observed that even though the redundancies are varying widely from one set to the other, the total number of support vectors still remains the same because of the ν-criterion. Hence there is a possibility that the ν parameter may introduce redundancies in the solution.

The algorithm $\nu-$Anomica described in the earlier sections is an extension of the classical One-Class SVMs. It has been shown that for both these methods

Table I
HERE WE COMPARE TWO CASES TO CHECK THE REDUNDANCY OF CLASSICAL ONE CLASS SVMS USING SYNTHETIC DATA SET.

Training size	Covariance	SVs (Exact)	
		Non-margin	Margin
1001	eps	100	1
1001	$10^{20}\times$ eps	100	1

the fundamental optimization procedure is exactly the same. In the following we will present interesting study on how these two techniques may produce a different outcome and try to provide some insight on what makes them different.

We also included a separate experiment where both $\nu-$Anomica and classical one class SVMs were developed on the same data set and the corresponding SVs were noted. Each support vector obtained by the classical approach was evaluated using the same hyperplane constructed by the exact solution itself and the hyperplane constructed by the approximate solution. In Fig. 3, scores for the support vectors from both solutions have been compared. The plots represent the absolute values of the original scores, sorted in descending order. With normalization, these scores almost lie on the top of each other. This is because the decision values for both these method will be proportional (Eqn. 8).

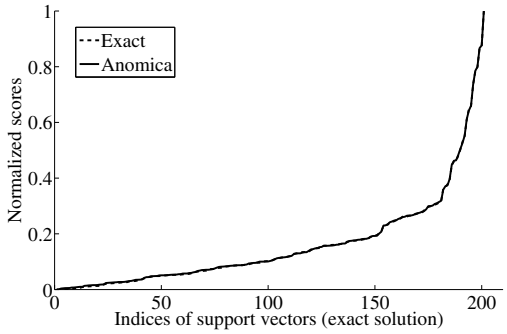


Figure 3. This figure represents the normalized scores from classical one class SVMs and $\nu-$Anomica.

IV. EXPERIMENTAL RESULTS

In this study, we have chosen two systems health management related data sets and one real-world astronomical data set as benchmark applications. These data sets represent diverse training set sizes, and input dimensionality and therefore builds a good platform to test the accuracy and scalability of these algorithms. Table II summarizes the characteristics of the data sets used for the experiments. Both one class SVMs and $\nu-$Anomica algorithms have been tested on a Dual core Pentium4 computer running Windows XP with 4 GByte of memory. The current version of our algorithms is based on the OSU SVM Classifier Toolbox (ver. 3.00) [1] and is written using Matlab. The OSU SVM Toolbox is an adaptation from the LIBSVM and uses Sequential Minimal Optimization (SMO) for solving the quadratic problem (Eqn. 3). To test these algorithms, nonlinear RBF kernel was used and the optimal setting of the kernel parameter was determined using the method described in [20]. In addition to that, it should be noted that for all analysis using $\nu-$Anomica the size of the initial subset is chosen to be 15% of the entire training set. However this parameter can vary depending on the problem size.

We first experiment with the emulated OPAD [19] (Optical Plume Anomaly Detection) data which is a set of time varying spectra profiles measured by an optical plume analysis in liquid propulsion engines. A second set of experiments were conducted on Sloan Digital Sky Survey (SDSS) photometry data (SDSS DR6 [2]) for testing the large scale training capabilities of our algorithms. The Commercial Modular Aero-Propulsion System Simulation (CMAPSS) data set has been used for the final set of analysis. The CMAPSS is a high fidelity system level engine simulation software for simulating user-specified transient engine behavior under normal and faulty conditions over flights. Detailed background on the CMAPSS framework can be found in [11]. The above data sets were split into non-overlapping training, validation and test sets as shown in table II.

Baseline results were obtained by running one-class SVMs model and compared with those obtained from ν-Anomica on the above data sets. Three sets of results were reported for analyzing the correct classification accuracy, sensitivity and time complexity of these algorithms. For CMAPSS data set, we will only summarize the outcomes of the analysis due to space limitations.

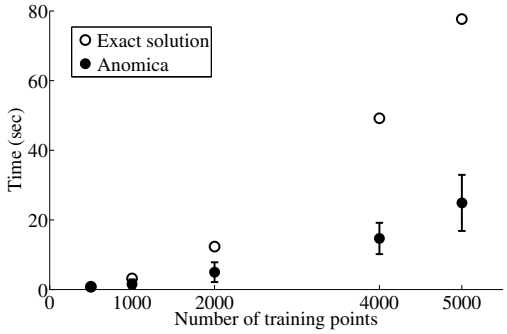
A. Run Time Analysis of the $\nu-$Anomica

Figures 4(a) shows the resulting training times for exact solution and ν-Anomica with five different sizes of training set on OPAD data. The exact solution uses the entire training set in all cases. ν-Anomica starts with an initial model built on a small subset of the entire training data set and updates the training set as it progresses towards the target $(1-\nu)$ classification rate on the validation set. In Fig. 4(a), we show the

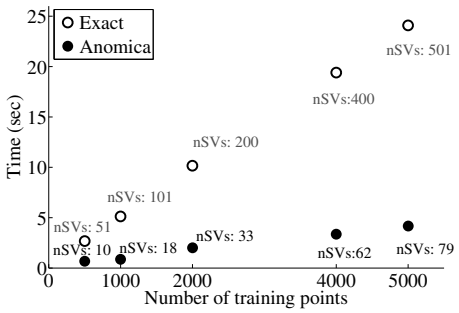
[1] http://svm.sourceforge.net/download.shtml
[2] http://www.sdss.org/dr6/

Table II
DETAILS ON THE DATA SETS USED TO TEST THE ν-ANOMICA ALGORITHMS

Data sets	Source	Variable Type	Number of Variables	Total Instances		
				Training	Validation	Testing
OPAD	Emulator	Continuous	1024	5×10^3	5×10^3	2×10^3
CMAPSS	Simulator	Continuous	29	500×10^3	20×10^3	100×10^3
SDSS	Real life data	Continuous	12	275×10^3	10×10^3	130×10^3



(a) This graph shows the mean training time complexity with symmetric error bars of $2\times\sigma$ long over 50 runs.



(b) This graph shows the mean test time complexity with symmetric error bars of $2\times\sigma$ long over 50 runs. In addition, for both classifiers, the number of support vector for each case has been indicated by the variable nSVs.

Figure 4. Training (a) and test (b) times of the one-class SVMs model and ν-Anomica with different sizes of the training sets using OPAD data.

mean training time over 50 runs for varying training sizes and their corresponding error bars. It is clear that with fewer training points the difference in training time for exact solution and ν-Anomica is low. As the size of the training data set increases, the computing time increases drastically for exact solution, however ν-Anomica shows much better performance. Table III presents the performance of these algorithms on the SDSS. It can be observed that the proposed technique outperforms one-class SVM model for all the test cases and the performance gain factor increases with increasing training set size. In Fig. 4(b), we present the time required to evaluate the OPAD test sets. As the number of SVs increase the resultant test time proportionally increases and this particular trend can be seen in the plot. Since ν-Anomica requires fewer SVs while building a model, the test time is lower compared to the classical approach. On SDSS data set, with 275k training and 130k test instances, ν-Anomica is on an average approximately 15 times faster than the classical method. With increasing training instances such as with CMAPSS data, ν-Anomica consistently performs on average 18 times faster with 500k training and 100k test instances.

B. Classification Accuracy and Prediction Performance

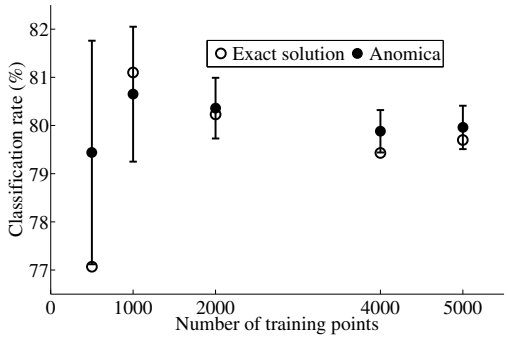


Figure 5. Figure comparing the classification rate of the test set using classical one class SVMs and ν-Anomica algorithm with different sizes of the training sets using OPAD data.

It could be of real interest to find out if the computational advantage of ν-Anomica trades off with the detector's ability to match the classification accuracy of the exact solution of one class SVMs. Figure 5 shows a comparison of the detection rates of both algorithms and these results were obtained on the same test set while the sizes of the training sets were varied. It can be seen that ν-Anomica overall provides similar accuracies when compared to one-class SVM but computed with much reduced training times. As the training size increases, the models get more accurate and as a result the classification rate of both the model gets more closer and

consistent. This is because introducing more training examples brings in additional useful information that aid correct detection and classification.

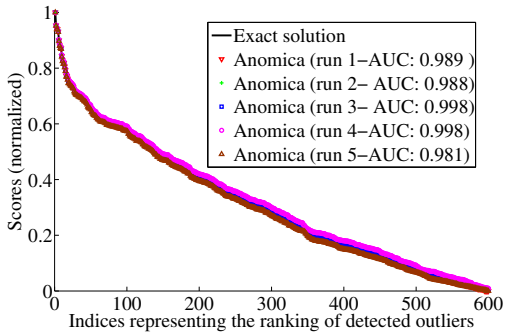


Figure 6. Figure showing the normalized scores of the outliers detected in a test set from OPAD data using one class SVMs and ν-Anomica, arranged in a descending order.

Now we present an analysis on predicting the "outlierness" of new unseen patterns. Figure 6 indicates that ν-Anomica ranked the points in terms of their "outlierness" comparably to classical one-class SVMs. This can be observed from the plot where both one-class SVMs and ν-Anomica have been used to predict a set of outliers in an unlabeled data set and their corresponding outlier scores were compared. These outliers were sorted based on the absolute values of their scores and thereafter normalized. Finally, to investigate the accuracy in separating the sequence of outliers from normal patterns, ROC analysis on the predictions of ν-Anomica was accomplished and the area under the ROC (AUC) was computed for each run. Here we have assumed that the sequence of outliers detected by one-class SVMs are the ground truth. Results obtained show that ν-Anomica consistently performed well in detecting the presence of these outliers and for each case the AUC was very close to 1.

V. Conclusion

In this paper, we presented a new method for faster anomaly detection using a modified one-class SVMs. Compared to classical one-class SVM all our experiments showed a competitive speedup (up to factor 15-18 on these data sets). The proposed method reduces the number of the operations needed to compute a reduced and near optimal training set. The model developed on this working set is a close approximation of the exact solution and can be represented with much less number of SVs. Hence both training time and test time is significantly reduced. However ν-Anomica can achieve very close classification accuracies (losing less than 1% in most cases) compared to one-class SVMs. The paper demonstrates the preliminary success of the proposed method on a wide variety of data sets. Also from all the experimental observations we find that the model converges in finite number of iterations which ensures that the cardinality of the final training set is always less than the cardinality of the entire training set. We note that the current version of the paper doesn't have a theoretical upper bound on the number of support vectors but we intend to consider this in our future research.

References

[1] F. Angiulli and C. Pizzuti. Outlier Mining in Large High-Dimensional Data Sets. *TKDE*, 17(2):203–215, 2005.

[2] S. D. Bay and M. Schwabacher. Mining Distance-based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule. In *Proceedings of KDD'03*, pages 29–38, 2003.

[3] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *DMKD*, 2:121–167, 1998.

[4] C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. In *Proceedings of NIPS'97*, pages 375–381, 1997.

[5] V. Chandola, A. Banerjee, and V. Kumar. Anomaly Detection: A Survey. *ACM Computing Surveys*, 2008 (to appear).

[6] C. Chang and Y. Lee. Generating the Reduced Set by Systematic Sampling. In *IDEAL'04*, number 3177, pages 720–725, 2004.

[7] N. Christianini and J. S. Taylor. *An Introduction To Support Vector Machines And Other Kernel-Based Learning Methods*. Cambridge, 2000.

[8] Chih chung Chang and Chih jen Lin. Libsvm: a library for support vector machines, 2001.

[9] K. Das and J. Schneider. Detecting Anomalous Records in Categorical Datasets. In *Proceedings of KDD'07*, pages 220–229, NY, USA, 2007.

[10] J. Fehr, Z. K. Arreola, and H. Burkhardt. Fast support vector machine classification of very large datasets. In *Data Analysis, Machine Learning and Applications*, number 3177, pages 11–18, 2008.

[11] D. K. Frederick, J. A. DeCastro, and J. S. Litt. Users guide for the commercial modular aero-propulsion system simulation (c-mapss). 2007. Technical Report: NASA/TM2007-215026.

[12] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori. Inlier-Based Outlier Detection via Direct Density Ratio Estimation. In *Proceedings of ICDM'08*, pages 223–232, Pisa, Italy, 2008.

Table III

IN THIS TABLE WE COMPARE THE PERFORMANCE OF CLASSICAL ONE CLASS SVMS AND ANOMICA USING DIFFERENT METRICS ON SDSS DATA SET. FOR $\nu-$ ANOMICA μ AND σ REPRESENTS THE MEAN AND THE STANDARD DEVIATION OVER 50 RUNS WITH A RANDOM INITIAL SET FOR EACH RUN.. THE SUBSCRIPTS E AND A STANDS FOR "EXACT" AND "ANOMICA" ALGORITHM RESPECTIVELY.

Data sets (Training)	Classification Rate (CR) (%)			Number of SVs ($nSVs$)			training time (tr) (in seconds)			test time (tst) (in seconds)		
	Exact	Anomica		Exact	Anomica		Exact	Anomica		Exact	Anomica	
N	μ_E^{CR}	μ_A^{CR}	σ_A^{CR}	μ_E^{nSVs}	μ_A^{nSVs}	σ_A^{nSVs}	μ_E^{tr}	μ_A^{tr}	σ_A^{tr}	μ_E^{tst}	μ_A^{tst}	σ_A^{tst}
5000	90.64	90.13	0.3	514	90	3.57	1.5	0.3	0.12	10.56	2.08	0.08
10000	90.33	90.33	0.27	1012	165	2.86	7.3	1.0	0.37	21.0	3.63	0.08
20000	90.23	90.15	0.25	2015	315	5.02	34.0	2.4	0.66	43.71	6.63	0.12
30000	90.16	90.14	0.21	3010	464	2.66	86.4	4.6	1.03	89.24	9.95	0.33
50000	90.08	90.33	0.18	5012	766	3.77	263.4	12.0	2.62	138.75	15.71	0.09
100000	90.24	90.2	0.18	10011	1514	12.84	1094.7	40.7	3.29	277.72	31.23	0.37
150000	90.01	90.12	0.17	15013	2268	7.26	2613.3	114.1	23.93	422.2	50.39	1.35
200000	90.07	90.07	0.15	20013	3012	2.38	4730.4	203.0	5.75	553.9	84.24	2.27
275000	90.03	90.48	0.14	27511	4161	8.95	9033.4	546.0	55.32	759.96	115.7	0.44

[13] V. Hodge and J. Austin. A Survey of Outlier Detection Methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.

[14] W. Hu, Y. Liao, and V. R. Vemuri. Robust Anomaly Detection using Support Vector Machines in Computer Security. In *Proceedings of ICML'03*, pages 168–174, 2003.

[15] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based Outliers: Algorithms and Applications. *The VLDB Journal*, 8(3-4):237–253, 2000.

[16] K. Lin and C. Lin. A Study on Reduced Support Vector Machines. *TNN*, 14:1449–1459, 2003.

[17] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient Algorithms for Mining Outliers from Large Data Sets. *SIGMOD Rec.*, 29(2):427–438, 2000.

[18] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Comput.*, 13(7):1443–1471, 2001.

[19] A. Srivastava, B. Mathew, and S. Das. Algorithms for Spectral Decomposition with Applications to Optical Plume Anomaly Detection. In *JANNAF'08*, 2008.

[20] Runarsson-R. T. Unnthorsson, R. and T. M. Johnson. Model selection in one class nu-svms using rbf kernels. In 16^{th} *conference on Condition Monitoring and Diagnostic Engineering Management*. Växjö University Press, 2003.

[21] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.