A Survey of Distance and Similarity Measures Used Within Network Intrusion Anomaly Detection

David J. Weller-Fahy, *Member, IEEE*, Brett J. Borghetti, and Angela A. Sodemann, *Member, IEEE*

***Abstract*—Anomaly detection (AD) use within the network intrusion detection field of research, or network intrusion AD (NIAD), is dependent on the proper use of similarity and distance measures, but the measures used are often not documented in published research. As a result, while the body of NIAD research has grown extensively, knowledge of the utility of similarity and distance measures within the field has not grown correspondingly. NIAD research covers a myriad of domains and employs a diverse array of techniques from simple k-means clustering through advanced multiagent distributed AD systems. This review presents an overview of the use of similarity and distance measures within NIAD research. The analysis provides a theoretical background in distance measures and a discussion of various types of distance measures and their uses. Exemplary uses of distance measures in published research are presented, as is the overall state of the distance measure rigor in the field. Finally, areas that require further focus on improving the distance measure rigor in the NIAD field are presented.**

***Index Terms*—Computer networks, anomaly detection, intrusion detection, machine learning, distance measurement.**

I. Introduction

THE GOAL of Network Intrusion Detection System (NID) is to automate the process of detecting when intrusions are occurring in a network. In a more general sense, intrusion detection can be framed as a subproblem within the network anomaly detection problem: determine whether traffic is normal (good) or anomalous (bad). Automated systems which discriminate between normal and anomalous often use some form of machine learning techniques such as classification or clustering to distinguish normal from abnormal traffic. At the heart of these systems is a comparison between the model of normal and the model of anomalous. These comparisons often rely on the ability to measure similarity or distance between a target and a known type, in order to determine whether to declare a new target anomalous or not. Thus, the efficacy of many anomaly detection systems (and many Network Intrusion Detection System (NIDS)) depend on the distance or similarity measures selected, and how they are used.

Manuscript received February 23, 2013; revised October 17, 2013 and April 1, 2014; accepted June 7, 2014. Date of publication July 11, 2014; date of current version March 13, 2015. This work was supported in part by the AFIT Center for Cyberspace Research (CCR).

D. J. Weller-Fahy and B. J. Borghetti are with the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH 45433 USA (e-mail: dave@weller-fahy.com; brett.borghetti@afit.edu).

A. A. Sodemann is with the Department of Engineering, College of Technology and Innovation, Arizona State University, Mesa, AZ 85212 USA (e-mail: angela.sodemann@gmail.com).

Digital Object Identifier 10.1109/COMST.2014.2336610

Unfortunately, much of the published work on NID fails to provide adequate detail on the distance and similarity measures used in the research. In many cases, distance measures are not mentioned. As a result, future research will have difficulty in replicating or comparing results to previous work. It is also difficult to explore the tradespace of distance measure selection when it is poorly documented in the research.

Some authors do present details on the distance measures used, and some even go as far as comparing performance across several choices of distance measures. We present these publications as exemplars in the field and make recommendations on what distance-measure details authors should include in their publications to improve research transparency, duplicability and comparability.

To provide a comprehensive review of how measures are applied to Network Intrusion (NI) datasets within NID research, this work examines how well the measures are identified, types of distance measures used within the field, and how they are used. The remainder of this paper is organized as follows. In Section II several examples of other surveys of distance measure use, both within and outside of the field of anomaly detection are discussed. Section III presents a primer on classification, clustering and anomaly detection, focusing on the importance of distance measures within the computational framework of machine learning. In Section IV we provide a detailed tutorial on the theory of distance measures, the types of measures researchers use, and a comparison of these measures. Section V provides a review of the publications using distance measures during the feature selection, classification, and clustering aspects of anomaly detection. We present lessons learned from the literature in Section VII and a conclusion in Section VIII.

II. Related Work

There have been many surveys of the field of Anomaly Detection (AD), including those which recommend different techniques for specific applications. In particular, Chandola *et al.* [1] provide a comprehensive review of the AD field, organized by the methods (e.g., statistical or classification) used to detect anomalies. While thorough, their approach is based on the type of detection used in the research rather than the type of distance measures used during detection. This work should provide a complementary look at the field of AD by providing a guide to how distance and similarity measures have been used.

Within the field of Network Intrusion (NI) datasets the closest work to this is by Chmielewski and Wierzchoń [2] which examines the problems inherent in using the l_p-metric (defined in (5), where $p = r \geq 1$), fractional l_p-distance (defined in (5),

where $0 < p = r < 1$), and cosine similarity (defined in (35)) to measure the distance between different samples of high-dimensional data. Through experimentation using differing values of p on the l_p-metric, and using the resulting distance in an application of negative selection to a NI dataset, they conclude that values of p on the interval [0.5, 1.0] should provide an improvement in detection rate compared to other values.

Outside the field of NI datasets the closest work to this is by Cha [3], and provides a syntactic and semantic categorization of distance and similarity measures as applied to probability distribution functions, as well as an analysis of the correlation between different measures using clustering and presented in hierarchical clusters. While not a review of how the measures are used, Cha's work is a useful reference for distance measures with an interesting partitioning based on how well the distance measures correlate to each other. A similar work with different intent by Deza and Deza [4] provides a comprehensive enumeration of the main distance measures used within a variety of different fields. The cross-disciplinary manner in which the list of distance measures is treated is especially useful when trying to identify measures used in published works, as synonyms and similar formulations are referenced throughout.

In areas not directly related to measuring difference and similarity in multi-dimensional datasets, but which may be useful in examining multi-dimensional datasets, there are other significant works. Staab [5] examines the relationships between ontologies and similarity measures, especially in light of the use of measures within logical reasoning systems. Cunningham [6] develops a useful structure for reasoning about which similarity measures to use when first examining a problem, by providing a taxonomy of similarity mechanisms. Chung *et al.* [7] develop a novel measure of security for nodes within a cloud system. Many factors including number, vulnerability, and exploitability of each virtual machine are used in calculating the virtual machine security index, and then in selecting the countermeasure for the particular attack.

III. Primer on Classification, Clustering and Anomaly Detection

In Network Intrusion Anomaly Detection (NIAD), the goal is to determine whether or not a specific observation(activity on the network) is anomalous. Anomaly detection requires labeling observations. Throughout the field of Network Intrusion Detection (NID) research there are subtle differences between the terms used when referring to the different phases of Anomaly Detection (AD). This primer provides explanations of the phases of a NIAD system, and definitions of the terms used within those phases.

- *Observation*: A single entity of data. In Network Intrusion (NI), an entity could be a computer network data packet, or the status of a particular server at a specific time.
- *Feature*: A particular type of information. Observations generally have several features. In a Network Intrusion Detection System (NIDS), features could include packet length, destination IP address, and time-stamp.
- *Dataset*: A collection of observations, each containing values for each of the features. Often a dataset is expressed in matrix form with the rows representing observations, and the columns representing features.
- *Preprocessing*: Any manipulation of the dataset required to allow the researchers' AD tools to operate on the dataset. Preprocessing is presumed to have no effect on the outcome of the experiment. For example, conversion from the comma-separated-value format to a database table within a relational database may be required, leaving the values of the features within the dataset unchanged.
- *Transformation*: Any change applied to the data with the purpose of scaling, normalizing, or weighting the data prior to its use. While the values of the features are changed, the number of features and ordering of the feature values are preserved. For example, some labeling methods are more effective if the observations being labeled are normalized to make all feature values lie between 0 and 1 prior to labeling.
- *Feature generation*: Any creation of new features based on original or derived datasets. For example, conversion of a feature with seven possible nominal values to seven binary features, or development of a new feature which is the square root of the sum of the squares of two other features.
- *Selection*: An operation which uses only a subset of all available features or observations for use in labeling.
- *Supervised Method*: A method that requires training using labeled training examples (the training dataset), prior to executing the trained system on unlabeled observations. The labeled training examples may be hand-labeled by the researcher, or may be the output of some previously run process.
- *Unsupervised Method*: A method that does not require training prior to execution on unlabeled observations.
- *Classification*: The act of labeling each observation in the data as being a member of a particular class. Most methods of classification are supervised.
- *Clustering*: Partitioning observations into groups based on similarity. Often clustering is unsupervised: the group labels are selected after the groupings are generated.

During AD, the labeling process which uses transformation, feature generation, selection, and classification or clustering often depends on distance measures between feature values of two or more observations in the dataset. For example, feature generation may require Euclidean distance to generate a new feature from two others. Clustering depends on some notion of distance in order to determine which observations are close to each other in a space relevant for anomaly detection. Since distance measures are so important for AD, we discuss them in detail in the following section.

IV. Distance Measures

A. Theory of Distance Measures

This paper examines the use of distance and similarity measures as used in Network Intrusion Detection (NID). There are a set of fundamental definitions of distance measures as identified widely in the mathematical literature [4]. The definition of a

distance measure includes three requirements. A fourth requirement establishes a sub-category of distance measures called distance metrics. To define these requirements, we will use the function $dist()$ which takes as input two distinct variables A and B, and returns the value of the distance. Then, the requirements for a distance measure are as follows:

1) Non-negativity: The distance between A and B is always a value greater than or equal to zero.

$$dist(A, B) \geq 0 \tag{1}$$

2) Identity of indiscernibles: The distance between A and B is equal to zero if and only if A is equal to B.

$$dist(A, B) = 0 \text{ iff } A = B \tag{2}$$

3) Symmetry: The distance between A and B is equal to the distance between B and A.

$$dist(A, B) = dist(B, A) \tag{3}$$

Distances which conform to at least these three requirements are known as distance measures. Those distance measures which also satisfy one additional constraint also qualify as distance metrics:

4) Triangle inequality: Considering the presence of a third point C, the distance between A and B is always less than or equal to the sum of the distance between A and C and the distance between B and C. This requirement is also known as the triangle inequality.

$$dist(A, B) \leq (dist(A, C) + dist(B, C)) \tag{4}$$

Although distance measures are generally thought of as measures to be applied to points in physical 3D space, distance measures can also be applied to multi-dimensional data that may not represent locations in space. This allows the application of distance metrics to such non-spatial data as commonly found in NID applications. Additionally, there are methods for comparing the difference between two points which do not satisfy even the first three requirements. These methods do not qualify as distance measures, but are referred to instead as similarity measures. Next, we examine the measures that have been used in recent NID literature.

B. Types of Measures

In examining the types of distance measures used within the NI field, it is useful to consider distance measures as part of distinct families or categories. The families selected for this work are among those enumerated in the, "Encyclopedia of Distances", by Deza and Deza [4]. As there is no definitive taxonomy within the NID field, the measures and indexes examined will be ordered by their relationship to families of measures.

1) Power Distances: Power distances are distance measures which use a formula mathematically equivalent to the power (p, r)-distance formula in (5).

$$\left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{r}} \tag{5}$$

Power (p, r)-distance measures the distance between two vectors x and y of length n. This category includes several of the most common distance measures, including the Euclidean distance and the Manhattan distance. Although typically thought of as a physical distance between two locations in 3-dimensional space, these types of distances can be applied to vectors of any dimensionality, so long as the type of data is numerical. The particular distance measure indicated by this category is determined by the values of p and r. For example, where $p = r = 2$ the power (p, r)-distance defines the Euclidean metric, and where $p = r \geq 1$ defines the l_p-metric. Other values of p and r give other metrics. For $0 < p = r < 1$ the power (p, r)-distance is known as the fractional l_p-metric.

To provide some intuition about what different p and r values would mean when measuring the distance between two points, Fig. 1 shows Voronoi diagrams [8] constructed using the power (p, r)-distance while varying the value of p and r. In Fig. 1, the shaded regions indicate the complete set of points for which the region's seed point (shown as a black dot) is closer than any other region's seed point. This type of diagram is useful for visualizing the discrepancy between our intuitive concept of physical distance and the concept of distance according to other distance measures. Fig. 1 shows that power distances other than Euclidean do not necessarily match our intuitive understanding of physical distance. In the context of NID, Weighted Euclidean distance can be used as a dissimilarity measure. The difference between weighted and unweighted Euclidean distance is the addition of a weight vector (w) of length n to the formula in (5).

Other distances related to the power (p, r) distance can also be used for NID, including the Mahalanobis distance and the Heterogeneous Distance Function. The Mahalanobis distance [9] is defined as shown in (6) [4]. Within (6), x and y are vectors which each have n elements, A is a positive-definite matrix (usually the covariance matrix of x and y), $\det A$ is the determinant of A, and $^{\mathrm{T}}$ indicates the transposition operation.

$$\|x - y\|_A = \sqrt{(\det A)^{\frac{1}{n}} (x - y) A^{-1} (x - y)^{\mathrm{T}}} \tag{6}$$

A simplified version of Mahalanobis distance developed by Wang and Stolfo [10] can be useful for quick computations during heavy computational loads required in high-throughput environments. The simplified measure, $m_{\mu,\sigma}$, is defined in (7) where x is a vector containing all the dimensions of a single observation, μ is the vector representing the center of mass of all data observations, n is the number of elements in x and μ, and $d(x_i, \mu_i)$ is the difference between the ith element of x and μ.

$$m_{\mu,\sigma}(x) = \sum_{i=1}^{n} \frac{d(x_i, \mu_i)}{\sigma_i} \tag{7}$$

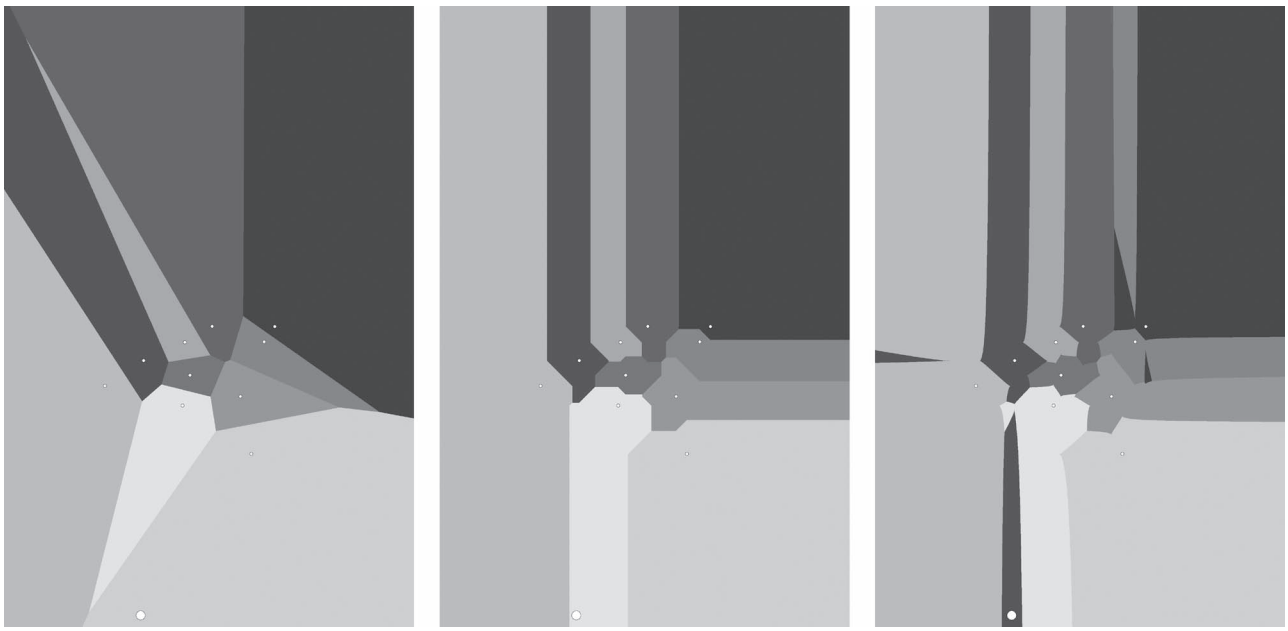

Fig. 1. Voronoi diagrams using power (p, r)-distance with $p = r = \langle 2, 1, 0.75 \rangle$ (from left to right).

The Heterogeneous Distance Function $(H(x, y))$ [11] is formulated as follows, where x and y are vectors of m elements that come from datasets X and Y, respectively. First, the distance between elements containing continuous values is defined in (8), where σ_j is the variance of the jth attribute of dataset X.

$$d_{\text{diff}}(x_j, y_j) = \frac{|x_j - y_j|}{4\sigma_j} \tag{8}$$

Next, the distance between elements containing discrete values is defined in (9) where $N_{j,x}$ is the number of records in which the value of the jth attribute is x_j, $N_{j,x,i}$ is the number of records in which the value of the jth attribute is x_j and the class is i, and k is the number of output classes.

$$d_{\text{vdm}}(x_j, y_j) = \sum_{i=1}^{k} \left| \frac{N_{j,x,i}}{N_{j,x}} - \frac{N_{j,y,i}}{N_{j,y}} \right| \tag{9}$$

As continuous and discrete distances are now defined, the conditional distance can be defined. Note the formulation in (10) accounts for missing attributes by providing a value where one or the other attribute is missing (x_j or y_j).

$$d_j(x_j, y_j) = \begin{cases} 1, & x_j \text{ or } y_j \\ d_{\text{vdm}}(x_j, y_j), & x_j, y_j \text{ discrete} \\ d_{\text{diff}}(x_j, y_j), & x_j, y_j \text{ continuous} \end{cases} \tag{10}$$

Finally, the heterogeneous distance function can be defined, which uses a modification of power (p, r)-distance where $|x_i - y_i|$ is replaced by the conditional distance defined in (10), $d_j(x_j, y_j)$, resulting in (11).

$$H(x, y) = \sqrt{\sum_{i=1}^{m} d_j^2(x_j, y_j)} \tag{11}$$

2) Distances on Distribution Laws: The second type of measure, distances on distribution laws, describes those measures based on the probability distribution of the dataset. Those include most forms of entropy, as well as conditional probability distributions. These distance measures are based upon distribution laws, and apply to probability distributions over variables with the same range.

One of the most common distance measures within this category is the Bhattacharya coefficient [4], which can be used to rank features according to the ability of each feature to distinguish one class from the others [12]. The Bhattacharya coefficient is shown in (12), where P_1 and P_2 are probability distributions over the domain X, $p_1(x)$ is the probability of x occurring in P_1, and $p_2(x)$ is the probability of x occurring in P_2.

$$\rho(P_1, P_2) = \sum_{x \in X} \sqrt{p_1(x) p_2(x)} \tag{12}$$

Another distribution law distance is the χ^2-distance. Equation (13) is standard χ^2-distance, where x and y are vectors of length n, $p(x_i)$ is the probability of the occurrence of the ith element of x, and $p(y_i)$ is probability of the occurrence of the ith element of y.

$$d(x, y) = \sum_{i=1}^{n} \frac{(p(x_i) - p(y_i))^2}{p(y_i)} \tag{13}$$

A modified χ^2-distance [13] can sometimes be useful for NID use, to measure the distance among rows and columns in a correspondence matrix between the original network data and generated datasets. The modified version proceeds as follows: Assume E is an $m \times n$ matrix, as shown in (14) at the bottom of the next page, with coordinates in each element (instead of values) for clarity. Each pair of vectors to be compared are either column or row vectors, but both must have the same orientation. Each column vector is length $m - 1$, and each row vector is length $n - 1$. The nth element of each row contains

the sum of the values in elements 1 through $n-1$ of that row, while the mth element of each column contains the sum of the values in elements 1 through $m-1$ of that column.

To compare two row or column vectors, an equation similar to (13) is used. For example, to compare the two row vectors shaded in (14), r_k and r_l, the necessary formula is in (15). In (15), r_{ki} is element i in row k, r_{kv} is the sum of all values in the row vector r_k, and c_{ui} is the sum of all values in the column vector c_i. If $i = s$, then the column vector c_i would be the shaded portion of column s in (14).

$$d(r_k, r_l) = \sum_{i=1}^{v-1} \frac{\left(\frac{r_{ki}}{r_{kv}} - \frac{r_{li}}{r_{lv}}\right)^2}{c_{ui}} \tag{15}$$

Equation (15) is very similar to (13), and the similarity is made more obvious with some substitutions. Let $x = r_k$, $y = r_l$, $p(x_i) = r_{ki}/r_{kv}$, $p(y_i) = r_{li}/r_{lv}$, and $n = v - 1$, then apply those substitutions to (15) to result in (16).

$$d(x, y) = \sum_{i=1}^{n} \frac{(p(x_i) - p(y_i))^2}{c_{ui}} \tag{16}$$

The difference between the two is the χ^2-distance uses $p(y_i)$ as the divisor, while the modified version uses c_{ui}.

Within probability distance measures is entropy, which can be calculated from any numerical random variable. Entropy can be used as a general summary measure of features, generating entropy vectors of a dataset for selected features. Entropy of a random variable is calculated as follows: the probability of a random variable X holding the value x is $P(X = x)$ or $p(x)$, and a is the base to use, the entropy of the random variable X, $H(X)$ is calculated using the formula in (17).

$$H(X) = -\sum_{x \in X} p(x) \log_a p(x) \tag{17}$$

Many variations of entropy can be useful in NID. These variations include standardized entropy, conditional entropy, and the Jensen Distance. Standardized entropy [14] is a method which compensates for variations in entropy due to the number of values of the random variable. Using the definition of entropy in (17), standardized entropy (H_s) is defined as shown in (18), where m is the total number of values in X and a is the base in which the entropy is calculated.

$$H_s(X) = \frac{H(X)}{\log_a m} \tag{18}$$

Conditional entropy is a distance measure which allows for two random variables. Where X and Y are discrete variables, the formula for conditional entropy is in (19).

$$H(X|Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log_a p(x|y) \tag{19}$$

The Jensen distance [15] uses an entropy-like function H'. The variables within H' are two of the summation variables a, b, and c; shown in (20) where $\phi_w(x)$ is the frequency of occurrence of w in x.

$$\begin{aligned} a &= \sum_{w \in L} \min\left(\phi_w(x), \phi_w(y)\right) \\ b &= \sum_{w \in L} \left(\phi_w(x) - \min\left(\phi_w(x), \phi_w(y)\right)\right) \\ c &= \sum_{w \in L} \left(\phi_w(y) - \min\left(\phi_w(x), \phi_w(y)\right)\right) \end{aligned} \tag{20}$$

The summation variables contain the various matches and mismatches: a the positive matches, b the left mismatches, and c the right mismatches. Equation (21) defines H' in terms of a, b, and the base for which the calculations are made, o.

$$H'\left(\phi_w(x), \phi_w(y)\right) = a \log_o \frac{2a}{a + b} \tag{21}$$

Using H' the Jensen distance is formulated as shown in (22).

$$\begin{aligned} &d_{\text{jens}}(x, y) \\ &\quad = \sum_{w \in L} H'\left(\phi_w(x), \phi_w(y)\right) + H'\left(\phi_w(y), \phi_w(x)\right) \end{aligned} \tag{22}$$

Besides entropy, also in the category of probability distance measures is the popular Kullback-Leibler distance (KLD), or information gain. The KLD is formulated as shown in (23), where P_1 and P_2 are probability distributions over the domain X, $p_1(x)$ is the probability of x occurring in P_1, $p_2(x)$ is the

$$E = \begin{pmatrix} (1,1) & (1,2) & \cdots & (1,s) & \cdots & (1,t) & \cdots & (1,v-1) & (1,v) \\ (2,1) & (2,2) & \cdots & (2,s) & \cdots & (2,t) & \cdots & (2,v-1) & (2,v) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (k,1) & (k,2) & \cdots & (k,s) & \cdots & (k,t) & \cdots & (k,v-1) & (k,v) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (l,1) & (l,2) & \cdots & (l,s) & \cdots & (l,t) & \cdots & (l,v-1) & (l,v) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (u-1,1) & (u-1,2) & \cdots & (u-1,s) & \cdots & (u-1,t) & \cdots & (u-1,v-1) & (u-1,v) \\ (u,1) & (u,2) & \cdots & (u,s) & \cdots & (u,t) & \cdots & (u,v-1) & (u,v) \end{pmatrix} \tag{14}$$

probability of x occurring in P_2, and a is the base in which the KLD is calculated.

$$\mathrm{KLD}(P_1, P_2) = \sum_x p_1(x) \log_a \frac{p_1(x)}{p_2(x)} \tag{23}$$

Another probability-related distance is the Hellinger distance, which is related to the more well-known Hellinger metric. The Hellinger metric is shown in (24), where P_1 and P_2 are probability distributions over the domain X, $p_1(x)$ is the probability of x occurring in P_1, and $p_2(x)$ is the probability of x occurring in P_2.

$$H_m(P_1, P_2) = \left(2 \sum_{x \in X} \left(\sqrt{p_1(x)} - \sqrt{p_2(x)} \right)^2 \right)^{\frac{1}{2}} \tag{24}$$

The Hellinger distance, as defined by Sengar *et al.* [16], is different from the Hellinger metric in significant ways. First, the multiple is changed from 2 to 1/2. Second, the square root of the entire formula is not calculated. Equation (25) shows the Hellinger distance.

$$H_m(P_1, P_2) = \frac{1}{2} \sum_{x \in X} \left(\sqrt{p_1(x)} - \sqrt{p_2(x)} \right)^2 \tag{25}$$

Finally, we have the likelihood ratio test. This is a distance similar to the likelihood ratio in (26) where H_0 hypothesizes no anomaly, H_1 hypothesizes the presence of an anomaly, X is a discrete variable, N is the total number of observations, $p(x_k|H_0)$ is the probability of x_k given no anomaly, $p(x_k|H_1)$ is the probability of x_k given the presence of an anomaly, and $L_N(X)$ is the likelihood ratio of X.

$$L_N(X) = \prod_{k=1}^{N} \frac{p(x_k|H_1)}{p(x_k|H_0)} \tag{26}$$

The likelihood ratio can be extended to calculate the likelihood ratio of a discrete variable (X), as formulated in (27).

$$L = \frac{p_1(x)}{p_2(x)} \tag{27}$$

The likelihood ratio can be used on packet-rate and -size features to detect attacks or normal traffic. In this application, thresholds are set for upper and lower boundaries, where if the likelihood ratio exceeds an upper boundary an attack is detected, and if the likelihood ratio falls lower than a lower boundary no attack is detected.

3) Correlation Similarities: Correlation similarities and distances are measures that attempt to characterize the correlation between two datasets, and treat that as a measure of similarity or distance, rather than using the probability distributions or magnitude of vectors.

The Spearman ρ rank correlation is one such measure of similarity. Equation (28) gives the Spearman ρ rank correlation where X_r and Y_r contain the rankings of discrete variables X and Y, x_i and y_i contain the ith rank in X_r and Y_r (respectively), X_r and Y_r have the same number of elements, and n is the number of elements in X_r.

$$\rho(X_r, Y_r) = 1 - \frac{6 \sum_{i=1}^{n} (x_i - y_i)^2}{n(n^2 - 1)} \tag{28}$$

The Kendall τ rank correlation is another similarity measure of this type. The Kendall τ rank correlation is defined as in (29), where the sgn function is used to calculate the number of discordant pairs of ranks subtracted from the concordant pairs of ranks.

$$\tau(X_r, Y_r) = \frac{2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{sgn}(x_i - x_j) \cdot \mathrm{sgn}(y_i - y_j)}{n(n-1)} \tag{29}$$

The *sign*, or *signum*, function is defined in (30).

$$\mathrm{sgn}(x) = \begin{cases} -1, & \text{if } x < 0 \\ 0, & \text{if } x = 0 \\ 1, & \text{if } x > 0 \end{cases} \tag{30}$$

In order to apply the Kendall τ rank correlation to NID, a slight modification can be made to the standard formulation in (29) which allows the capturing of both similarities and differences. The result is shown in (31) where the eq function is defined as shown in (32).

$$\tau'(X_r, Y_r) = \frac{4 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{eq}(\mathrm{sgn}(x_i - x_j), \mathrm{sgn}(y_i - y_j))}{n(n-1)} - 1 \tag{31}$$

$$\mathrm{eq}(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{otherwise} \end{cases} \tag{32}$$

A third correlation coefficient method that can be applied to NID is the Pearson product-moment correlation linear coefficient, or r. The formula for r is given in (33), where $\bar{X}$ is the mean of the discrete variable X.

$$r(X, Y) = \frac{\sum_{i=1}^{n} (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{j=1}^{n} (x_i - \bar{X})^2 \sum_{j=1}^{n} (y_i - \bar{Y})^2}} \tag{33}$$

An equivalent formulation of the Pearson product-moment correlation linear coefficient that has been used for NID applications is defined in (34).

$$\begin{aligned} A &= n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i \\ B &= \sqrt{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i \right)^2} \\ C &= \sqrt{n \sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i \right)^2} \\ r(X, Y) &= \frac{A}{B \cdot C} \end{aligned} \tag{34}$$

A fourth method in this category is one that is based on Learning Vector Quantization, and uses the cosine similarity with an Artificial Neural Network. The definition of cosine similarity is shown in (35), where ϕ is the angle between vectors x and y.

$$\cos\phi = \frac{\langle x, y\rangle}{\sqrt{x^2}\cdot\sqrt{y^2}} \tag{35}$$

In application of the cosine similarity and Learning Vector Quantization to NID, a similarity is defined to determine which neuron of an Artificial Neural Network 'wins' [17]. The formulation is given in (36), where U is one of the training set of attack samples, V a neuron, W_u and W_v are the weight vectors associated with U and V (respectively), W_u and W_v have the same number of elements, and n is the total number of elements in W_u.

$$Sim(U,V) = \frac{\sum_{k=1}^{n} W_{uk}\cdot W_{vk}}{\sqrt{\sum_{k=1}^{n} W_{uk}^2}\sqrt{\sum_{k=1}^{n} W_{vk}^2}} \tag{36}$$

The cosine similarity can also be used as a measure of self-similarity [18]. In this implementation of the cosine similarity, shown in (37), $\vec{G_t}$ is a system status snapshot during time interval t, and $\vec{1}$ is a vector of all ones with the same dimensionality as $\vec{G_t}$.

$$S_t(\vec{G_t}) = \frac{\langle \vec{G_t}, \vec{1}\rangle}{\sqrt{(\vec{G_t})^2}\cdot\sqrt{(\vec{1})^2}} \tag{37}$$

The running mean of S_t is taken over the last k time periods (where k is selected using methods described in the paper), then each S_t is compared to the mean to determine whether it is within limits set by the researchers. If S_t is not within two standard deviations of the running mean, then some authors assume an attack or security violation has taken place.

4) Other Similarities and Distances: Some similarity measures that are useful for NID do not fit into the three primary categories above. For example, one method uses the Dice similarity [19], defined in (38), where X and Y are strings from Internet Relay Chat channels, an n-gram is a subsequence of n items from a given sequence, and the number of n-grams in X is given by $|\text{ngrams}(X)|$.

$$\text{Dice}(X,Y) = \frac{2\,|ngrams(X)\cap ngrams(Y)|}{|ngrams(X)| + |ngrams(Y)|} \tag{38}$$

A variation of the χ^2 distance known as squared χ^2 (also known as χ-Squared) [20] can also be used to compute distances with n-grams. The squared χ^2 measure is shown in (39), where S is the set of all possible n-grams, x and z are byte sequences taken from the packet payload, $\phi_s(x)$ is the frequency of occurrence of s in x.

$$d_{\text{sq}\chi^2}(x,z) = \sum_{s\in S}\frac{|\phi_s(x)-\phi_s(z)|^2}{\phi_s(x)+\phi_s(z)} \tag{39}$$

Some of these similarities and distances that do not fit into the three primary categories are particularly useful in application to NID that is not based upon raw network traffic or features built from the packets, but on other data characteristics. One such method is based upon a linear kernel (d_{lk}, see (40) [15].

$$d_{\text{lk}}(x,y) = \sum_{w\in L}\phi_w(x)\cdot\phi_w(y) \tag{40}$$

Note that this is effectively the dot product of the probability distributions of w within x and y.

A variation of the Geodesic distance is another such method that can be applied to NID. For this kind of application, the Geodesic distance may be formulated as $d_{\text{gd}} = \arccos(d_{\text{lk}})$, which is somewhat different than the standard definition of Geodesic distance.

C. Comparison of Types of Measures

Different types of distance and similarity measures exhibit different properties that should be taken into account before applying them to a NID application. These differences make certain measures more or less useful for certain types of data, degrees of dimensionality, and other considerations. For example, while all distance measures presented here satisfy requirements one (non-negativity) and two (identity of discernables) for distance measures, not all satisfy requirements three (symmetry) and four (triangle inequality). Here, the types of distance measures and similarity measures described in Part B will be compared according to the definitions presented in Part A, and according to other key differences.

1) Power Distances: All power distances meet all four of the requirements for distance measures, and qualify as distance metrics. When applied to NID applications, Euclidean distance shows clear differences between normal and attack samples, and provides characterizations of the different types of traffic records. Euclidean distance can also be applied indirectly, by first utilizing a distribution measure, such as the Discrete Fourier Transform, then applying Euclidean or other power distance measure to the frequency domain. This approach can potentially solve some of the problems with applying Euclidean distance directly to the features (or to the traffic itself).

In NID applications, traffic data may often have a large number of features. In this high-dimensionality case, the fractional l_p method may be more effective than Euclidean in higher dimensional datasets, and the utility of fractional l_p-distance may increase as dimensionality increases. This may be the case even though fractional l_p-distance does not result in a familiar definition of "nearest" in two- and three-dimensional space, as Fig. 1 demonstrates. Notice how fractional l_p values such as 0.75 yield discontinuous regions of nearest spaces in the figure.

2) Distances on Distribution Laws: Only some of the methods for measuring distances on distribution laws qualify as distance measures and metrics. These distances are often referred to as divergences rather than distances in the case that they fail to satisfy the triangle inequality and symmetry. For example, the KLD, described in Part B of this Section, is actually a divergence, not a distance, because of its violation of

both the triangle inequality and symmetry. The Bhattacharyya distance violates the triangle inequality, but does exhibit symmetry. The Hellinger distance complies with both the triangle inequality and symmetry. Conditional Entropy also qualifies as a divergence or similarity measure, due to its violation of both symmetry and the triangle inequality.

Besides their properties relating to the definition of distances, other properties of distances on distribution laws can motivate their use. For example, the Mahalanobis distance is particularly beneficial because the distance calculated accounts for the variance within the sample data, therefore allowing outliers to be more accurately identified. Conditional entropy can be used in NID for purposes other than distance measures, such as a measure of an individual flow of traffic.

3) Correlation Similarities: Correlation similarities are different from distances on distribution laws and power laws in that they are able to operate not only on continuous or discrete types of data, as distribution law and power law methods do, but also on rankings, or ordinal types of data. Thus, the four requirements for distance measures do not necessarily apply to these methods; they are all considered to be similarity measures, rather than distance measures. Nevertheless, all of the correlation similarities presented here do satisfy at least requirement two for distance measures (identity of indiscernibles).

Correlation similarities are types of hypothesis tests which test the hypothesis of correlation between two sets of data. The Kendall τ rank correlation is the most general of the methods addressed in this paper. The Kendall τ rank correlation is a type of non-parametric test; it is used to test the degree of statistical dependence between two variables, without reliance on any assumptions on the distributions of the two variables.

The Pearson product-moment correlation and the Spearman ρ rank correlation are very similar types of measures: both are used to assess how well a variable can be described by a third reference function. The difference between these two measures is that the Pearson product-moment correlation gives a measure of the linear correlation between two variables, while the Spearman ρ rank correlation assesses how well the relationship between two variables can be described using a monotonic function.

The cosine similarity is a more simplistic type of correlation similarity which measures the degree of similarity between two vectors in terms of orientation, rather than magnitude.

Within NID applications, it is sometimes useful to make use of multiple correlation coefficients within a single distance measure, in order to detect similarities a single measure might not catch.

This section compared the three different types of distance measures available: power distances, distances on distribution laws, and correlation similarities. In the next section, we identify the uses of these distance measures within the NID field of research.

V. Distance Measure Use

Examination of the current state of distance and similarity measure use requires a review of recent work. The goal is to examine how research is using measures within the Anomaly Detection (AD) field. We analyze research published since 2005 to find articles which provided both names and formulas for the measures used. In addition to the discovery that many authors do not provide sufficient information about the measures used to replicate their work, we also discovered that while measures are used ubiquitously in this field, they are only used in a subset of the phases outlined in Section III.

This review found measures used within the phases of feature selection (Section V-A), classification (Section V-B), and clustering (Section V-C). In the explanation of the measures used the works are separated by the phase of use, and only exemplar articles providing excellent description and identification of the measure(s) used are examined. To provide consistency in labels all measures named within this work are referred to by the standard names described in, "Encyclopedia of Distances," by Deza and Deza [4].

A. Feature Selection

The first AD phase to show explicit use of measures within the literature is feature selection. The measures used in feature selection tend to be those related to probability, as the probable occurrence of a feature is a useful mechanism by which to reduce dimensionality of large datasets.

Eid *et al.* [21] develop a feature selection method for Network Intrusion Detection (NID) in which the first layer uses Kullback-Leibler distance (KLD) to rank the features in the dataset. KLD, or information gain, is formulated as shown in (23), where P_1 and P_2 are probability distributions over the domain X, $p_1(x)$ is the probability of x occurring in P_1, $p_2(x)$ is the probability of x occurring in P_2, and a is the base in which the KLD is calculated.

The features are ranked by KLD, then the dataset is classified using the J48 classifier (an open source variation of the C4.5 decision-tree algorithm). After classification local and global maxima accuracy are identified, and features are selected for inclusion when performing the final classification based on the identified maxima. The resulting reduced set of features led to an increased classification F-measure of 99.2% using only 20 features on the NSL-KDD dataset. The contribution of the paper is not limited to a new framework for feature selection, but is also the repeatability of the work. Eid *et al.* [21] are precise in their definitions of the measures used, and the methods used to achieve results, such that confirming their experiments would be relatively simple.

In another notable recent work Hancock and Lamont [12] perform feature selection in a multi agent network attack classification system using the Bhattacharyya coefficient (12) to rank features according to the ability of each feature to distinguish one class from the others.

The three features with the largest overlap (largest $\rho(P_1, P_2)$ value) are selected after rejecting any feature which is strongly correlated to a higher ranked feature to reduce redundancy among selected features. The feature selection is part of each agent within their system. The agents are then distributed throughout the network in an attempt to provide an effective multi-agent Network Intrusion Detection System (NIDS) using reputation.

Other works use distances related to probability in feature selection problems, and in ways which may not be obviously distance related. Wang *et al.* [14] use an entropy feature vector calculated using standardized entropy as defined in (18). Entropy is then used to calculate a value for each feature before training a particular type of classifier.

In feature selection the use of measures is limited to those related to probability. However, the way in which the measures are used within feature selection varies from simple entropy calculations to rank ordering features based on KLD along with multiple classification runs. Use of the measures in feature calculation is limited to calculation using a single feature (measurement for future comparison), or calculation using two or more features (direct comparison), but those two can be used with measures to provide a myriad of different methods for selecting features.

B. Classification

After feature selection, classification of the dataset using the selected features is the next step in the process. Some prior research has very effectively compared classification performance results when using multiple distance measures. Chmielewski and Wierzchoń [2] examine the use of the l_p-metric ((5), with $p = r \geq 1$), fractional l_p-distance ((5), with $0 < p = r < 1$), and cosine similarity (35) to measure the distance between different samples of high-dimensional data. Through experimentation using differing values of p on the l_p-metric, and using the resulting distance in an application of negative selection to a Network Intrusion (NI) dataset, they conclude that values of p on the interval [0.5, 1.0] should provide an improvement in detection rate compared to other values when applied to high-dimensional datasets.

In another work, Chmielewski and Wierzchoń [22] suggest using both binary and real-valued detectors to detect non-self samples. They demonstrate that the results are both consistent with the theory that the utility of fractional l_p-distance increases as dimensionality increases, and that using two different types of detectors significantly increases the coverage of the non-self region by generated detectors. Both works demonstrate applications of the power (p, r)-distance in ways that challenge common intuition about distance.

Tan *et al.* [23] generate Euclidean Distance Maps for the features of each sample in the dataset, then convert the distances in each cell of the map to a color for purposes of rapid visual comparison. The visualization showed clear differences between normal and attack samples, and provide novel characterizations of the different types of computer network traffic records. While Euclidean distance is not used as the singular difference between two datasets, the generation of a feature distance matrix is a novel method of classification.

Gu *et al.* [19] developed a method of detecting Botnets that uses the Dice similarity (see (38)) where X and Y are strings from Internet Relay Chat channels, an n-gram is a subsequence of n items from a given sequence, and the number of n-grams in X is given by $|\text{ngrams}(X)|$.

Most of the analysis encountered in this review is based upon raw network traffic or features generated from the network packets. The use of data from a higher-level abstraction than that usually addressed by NID (character streams) gives a glimpse of an entire area of the field in which little work is done in comparing the efficacy of different measures.

Zhao *et al.* [24] use a single distance measure which incorporates one of three correlation coefficients to detect stepping-stone attacks, where one computer is used by the attacker to reach another. The authors use the alternative to the Kendall τ rank correlation previously described in (29), (31), (33) and (34). Each measure is applied to the two traffic streams, and each result is subtracted from the number one, to calculate the minimum distance between the two streams $(\sigma(X, Y))$, as shown in (41).

$$\sigma(X, Y) = \min\left(1 - \rho(X_r, Y_r), 1 - \tau'(X_r, Y_r), 1 - r(X, Y)\right) \quad (41)$$

If $\sigma(X, Y)$ is less than a given threshold (set by the researcher), then the compared pair is similar enough to be considered relayed traffic. The use of multiple correlation coefficients within a single distance measure is a good example of using multiple measures to detect similarities a single measure might not catch.

Classification is useful, but it assumes *a priori* knowledge of the classes to which anomalies will belong as well as the unique characteristics which define that class. Clustering, on the other hand, can be helpful in determining how many classes there are and identifying any unique characteristics of a given class when those two pieces of information are unknown.

C. Clustering

Some published works effectively use more than one family of distance measures to cluster data. One such work is by Lakhina *et al.* [25] which uses measures from both the power (p, r)-distance and probability families. The first measure used is the entropy of the traffic, as shown in (17).

The authors use entropy as a general summary measure of features, generating entropy vectors of the sample set for selected features. The second measure used is the squared Euclidean distance (power (p, r)-distance where $p = 2$ and $r = 1$), which the authors use to calculate the magnitude of the anomalous component of the entropy vector. The authors demonstrate that entropy is an effective method of detecting unusual traffic feature distribution changes caused by anomalies, and that their multiway subspace method is an effective method of extracting anomalous traffic changes.

The clear explanation of methodology for both the reason for the use of multiple measures from different families, and the qualities needed for the problem being solved, is extremely helpful in understanding how and why each measure was used. The authors did not provide a formula for the squared Euclidean distance, but they did identify it as the ℓ_2 norm, and later as $\|\tilde{\mathbf{x}}\|^2$, where $\tilde{\mathbf{x}}$ is a vector of the anomalous components. Although the formula is not provided, the level of clarity was sufficient to allow understanding and repetition of the experiment.

Rieck and Laskov [15] utilize a myriad of distance measures in evaluating the effect of measure choice on anomaly detection accuracy. The first defined measure is a linear kernel (d_{lk}, see (40) where L is the language corresponding to all the sequences of length n possible in incoming connection payloads, w represents a sequence which is part of L, x and y are two incoming connection payloads, and $\phi_w(x)$ is the frequency of occurrence of w in x. This is effectively the dot product of the probability distributions of w within x and y.

The second defined measure is identified by the authors as Geodesic distance, which the authors formulate as $d_{\mathrm{gd}} = \arccos(d_{\mathrm{lk}})$. It should be noted this does not correspond to the standard definition of Geodesic distance. The third defined measure is Canberra distance (d_{can}) as shown in (42).

$$d_{\mathrm{can}}(x,y) = \sum_{w \in L} \frac{|\phi_w(x) - \phi_w(y)|}{\phi_w(x) + \phi_w(y)} \tag{42}$$

The fourth distance measure defined by the authors is Jensen Distance, another measure not defined in the Encyclopedia of Distances, which uses an entropy-like function H'. The variables within H' are two of the summation variables a, b, and c; shown in (20)–(22).

Of the four measures defined, two of them are not found in the Encyclopedia of Distances: d_{gd} and d_{jens}. In addition to the distance measures, the authors describe four similarity coefficients also used to compare two input connection payloads, which use the summation variables defined in (20). The similarity coefficients described are the Jaccard (s_j), Czekanowski (s_c), Sokal-Sneath (s_s), and Kulszynski (s_k), as shown in (43).

$$\begin{aligned} s_{\mathrm{j}} &= \frac{a}{a+b+c} \\ s_{\mathrm{c}} &= \frac{2a}{2a+b+c} \\ s_{\mathrm{s}} &= \frac{a}{a+2(b+c)} \\ s_{\mathrm{k}} &= \frac{1}{2}\left(\frac{a}{a+b} + \frac{a}{a+c}\right) \end{aligned} \tag{43}$$

Rieck and Laskov [15] also use a simplified version of Mahalanobis distance developed by Wang and Stolfo [10] for use in high-throughput environments. The simplified measure, $m_{\mu,\sigma}$, is defined in (7).

Two aspects of the authors' treatment of measures are notable. First, the authors purposefully include the distance and similarity measures in the portion of the experiment to be changed, examining the effect different measures have on their results. This is especially notable as only five papers are discovered in this review which focused on the impact of measurement selection on the results. Second, the simplified Mahalanobis distance is used as an anomaly detection mechanism by the authors: it is not included in the list of distance measures.

In this section, we presented exemplary research, showing how the authors effectively identified and presented clear parameterization of their distance measure choices. We also showed how several different techniques are used to compare system performance when the distance measures or parameters are altered. Finally, it is important to note that several authors identified novel uses of distance measures with success. Despite these few good examples of proper treatment of distance measures we are able to find, it is unfortunate to note, as will be described in the next section, that the large majority of research in this field does not share a similar rigorous approach to identifying and explaining their use of distance measures in NID.

TABLE I
QUANTITY OF EXPLICITLY NAMED DISTANCE OR SIMILARITY MEASURES WITHIN SAMPLED WORKS

Measure identified	Count	References
Standard	29	[26]–[54]
Novel	4	[55]–[58]
Not given	67	[59]–[125]

VI. AN ANALYSIS OF THE USE OF DISTANCE MEASURES IN THE FIELD

To provide an objective view of the state of distance measure identification and explanation in the literature, we randomly sampled a cross-section of the publications in Network Intrusion Detection (NID) research. First, we identified 556 papers published between 2008 and 2012 (inclusive) containing the terms "network intrusion" in the title, abstract, or keywords using Google Scholar's reverse citation lookup function to a depth of three. Of these papers, we randomly selected a sample of 100 papers to manually review.

Our survey uses the, "Encyclopedia of Distances," [4] as the standard listing of measure definitions. Any names and formulas found in the sample are translated to match those within the Encyclopedia. To quantify how well the name and formulation of each distance measure matched the standard, we categorized papers into three groups:

- **Standard**: The measure name or formula is explicit in the paper, and identifiable in the Encyclopedia of Distances.
- **Novel**: The measure name or formula is explicit in the paper, and is not identifiable in the Encyclopedia of Distances.
- **Not Provided**: The measure name or formula is not explicit in the paper.

The categorical information from the 100 papers in this sample is depicted in two tables: Table I shows the quantity of papers which explicitly named the distance measures used, and Table II categorizes papers based on whether specific distance or similarity measure formulas are provided.

Among the work sampled for this review, 65 of the papers did not provide a measure name, and 74 of the papers did not provide an explicit formulation.

It is useful to understand which types of measures are being used in the field of NID, and which are not. Focusing on only the work with specified distance measures, there are 33 articles with distance measures that are named, some with more than

TABLE II
QUANTITY OF DISTANCE OR SIMILARITY MEASURES WITH EXPLICIT FORMULATIONS WITHIN SAMPLED WORKS

Formula provided	Count	References
Standard	21	[28]–[31], [33]–[36], [38], [39], [41], [43]–[47], [49]–[51], [53], [54]
Novel	4	[55]–[58]
Not provided	75	[26], [27], [32], [37], [40], [42], [48], [52], [59]–[125]

TABLE III
FREQUENCY OF DISTANCE OR SIMILARITY MEASURES TYPES USED WITHIN SAMPLED WORK

Measure type	Count
Power (p, r)-distance	20
Distances on distribution laws	11
Distances on strings and permutations	4

TABLE IV
FREQUENCY OF DISTANCE MEASURE COMPARISON

Measures compared	Count	References
1	98	[26]–[34], [36], [38]–[125]
2	1	[37]
3	1	[35]

one measure, and three types of distances identified within the sampled work. The three different types of measures observed are all defined in the Encyclopedia of Distances: power (p, r)-distance, distances on distribution laws, and distances on strings and permutations. The distribution of distance measures within the 33 papers is listed in Table III.

The majority (57% of works with measure specified) of the distance measures used within the sample are based on the power (p, r)-distance and distances on distribution laws. The sample shows that there is little exploration of the possible use of other measures within the field, as most of the measures are based upon the power (p, r)-distance or distances on distribution laws.

This survey of 100 papers in the field also revealed that the vast majority (98%) of papers do not examine more than one distance measure during any single phase of NID. Table IV shows that only two papers compared more than one distance measure during a phase of research. This striking finding reveals that there is more to explore in the trade space of distance measure choice in research.

Given these findings, we can see clearly that there is work to be done regarding identification, specification, providing rationale, and comparing the use of distance measures. The contents of the 100-paper sample categorized in Table IV are summarized in the following entries. These entries are organized primarily by whether the measure(s) used are defined in the work, and then subcategorized by the phase of NID in which the measure is used.

TABLE V
MEASURES USED IN SELECTION

Type	Measure	Count	References
Power (p, r)	Euclidean	2	[27], [43]
	Mahalanobis	1	[46]
Distances on distribution laws	Kullback-Leibler distance (KLD)	4	[28], [35], [50], [51]
	Entropy	2	[33], [48]

A. Measure Defined

In an Intrusion Detection System (IDS) there are three phases where distance or similarity measures are observed within the sample: selection, clustering (unsupervised classification), and supervised classification. One of the first possible uses of a distance or similarity measure in an IDS is the selection phase. A breakdown of the different measures identified in the sample is provided in Table V. The two categories of measure stand out in their use in selection: power (p, r) distance and distances on distribution laws.

1) Selection: There are three examples that use power (p, r) distance as the basis for feature selection. Li *et al.* [27] propose the use of k-means clustering to reduce the number of observations evaluated, and the authors discuss using the *distance* between data and clusters. Although the authors never define *distance* precisely, the use of k-means clustering suggests Euclidean Distance is a reasonable assumption. Tan *et al.* [46] generate Mahalanobis Distance Maps to perform feature selection, then analyze the results of the comparisons between normal and attack packet feature maps. The map analysis is then used to select the features for classification. Lu *et al.* [43] select features based on the Isomap algorithm, which is used for dimensionality reduction while maintaining the *nearness* of data points doesn't change while dimensionality is reduced.

Power-(p, r) distances are very useful tools, and are used often in almost all works, but there are difficulties to be considered if using power (p, r) distance. For example, setting $p = r = 2$ results in Euclidean distance, which, when used on high-dimensional data can have counter-intuitive results.

There are six examples of the use of distances on distribution laws during the feature selection phase. The first four make use of KLD. Sindhu *et al.* [28] use KLD to calculate the gain ratio of each attribute during the feature selection process. The gain ratio is then used in calculating the attribute farthest from a gain ratio of zero (and thus most useful for classification), and is implicitly used as a similarity measure. Wang *et al.* [51] and Singh and Silakari [50] use raw KLD to determine the features selected: features with high values of KLD are retained. In all three examples KLD is used as a basic measure of similarity with a threshold set by the research used to determine the gain. Lima *et al.* [35] present a comparison of different entropy measures (Rènyi, Tsallis, and Shannon) for feature selection. The three definitions of entropy are each used for feature

TABLE VI
MEASURES USED IN CLUSTERING

Type	Measure	Count	References
Power (p, r)	Euclidean	10	[32], [33], [37], [39], [40], [42], [45], [47], [50], [53]
	Euclidean^2	2	[41], [44]
Distances on strings and permutations	Identical attributes	1	[36]
Unknown	$d_{i,j}$	1	[52]

reduction based upon the gain ratio, then the 10 attributes with the highest gain ratios are selected.

Rather than KLD, the last two papers makes use of entropy in the selection phase. Devarakonda *et al.* [33] use entropy as a method of feature selection. The features are ranked in descending order by entropy, then the top 15 features are selected for use in classification. Chen *et al.* [48] use the entropy of network traffic flows to train a Support Vector Machine (SVM). They compared the attack detection rate and accuracy of SVM classifiers trained on the full feature-set of the Knowledge Discovery and Data Mining Cup 1999 (KDD99) dataset, the features selected by rough set theory, and entropy. Entropy is used as a measure of each flow, but is not used as a distance between flows, and is a good example of the distances on distribution laws category.

In the entire sample only one paper examined the effect different measures had on the experiment when applied to selection, and most just used information gain. The measures used within the sample clearly indicate a lack of published exploration in this area as well as an implied suitability of KLD as a feature selection measure.

2) Clustering: Unsupervised classification, or clustering, provides a method of discovery for groups that are unknown to the user. Those groupings depend in part on the method of clustering, and the measure used to determine distance between the instances. In particular, the type of measure is critical in determining the efficacy of clustering. A breakdown of the different measures identified in the sample is provided in Table VI. Three different measure types are used in the clustering phase among the papers in the sample: power (p, r) distance, distances on strings and permutations, and a *distance* variable.

Among those clustering methods which use power (p, r) distance the most common clustering method is the k-means and its derivatives. There are a wide variety of implementations among the literature when examining the use of k-means clustering: Brahmi *et al.* [40] use the standard k-means methodology, Borah *et al.* [32] hash the inputs before clustering, Yang and Mi [47] use a fitness function to maximize the inter-cluster distance and minimize the intra-cluster distance, and Bharti *et al.* [45] propose a modification to the k-means algorithm to solve the no class and class dominance problems.

Aside from k-means clustering, some other unsupervised classification methods are implemented using power (p, r) distance as the measure of choice. Devarakonda *et al.* [33] use the k-nearest neighbor method of clustering as one of many in a voting ensemble. Singh and Silakari [50] also make use of k-nearest neighbor as the classification method. The use of distance is in the context of the k-nearest neighbor classifier, and authors specify the use of Euclidean distance, but note that other distances (such as the Manhattan distance) could be used instead. Chou *et al.* [53] classify with a fuzzy c-means clustering technique. This method makes use of an optimization function that is specified as based upon the l_2-norm (Euclidean distance). Hayat and Hashemi [42] demonstrate clustering with the constraint of limited memory, and introduce a clustering method based on Discrete Cosine Transform. Cheng and Wen [41] utilize a Self-Organizing Map for classification. Palmieri and Fiore [44] view traffic flows as a dynamic system, and therefore use recurrence quantification analysis as an unsupervised classification method. However, the authors recognize that transformation between dimensionalities can cause the distances between points to become distorted, and attempt to compensate by identifying, "false nearest neighbors," and use the squared Euclidean distance as a metric. Finally, Obimbo *et al.* [37] investigate vote-based use of self-organized feature maps. Several self-organized feature maps are trained, and each votes on the resulting data shape. In experimentation, the authors compare use of Euclidean distance to use of a customized distance measurement. Zheng and Wang [39] investigate a clustering approach which consists of two phases: clustering phase which is specified as using Euclidean distance, and Particle Swarm Optimization phase which uses the clusters from the clustering phase as the initial particles. The Euclidean distance of a point from a cluster center is utilized in the Particle Swarm Optimization fitness function.

The distance or similarity measures used are not always based on physical distance. Gogoi *et al.* [36] demonstrate a clustering approach to anomaly detection in which the similarity between two objects is defined as the number of attributes which have identical values between the two objects. In another work, Zhuang *et al.* [52] introduce a proximity-assisted IDS approach to identify worms. A clustering approach is used based on, "proximity". The algorithm presented makes extensive use of a variable representing distance, but this variable is not defined.

3) Supervised Classification: Supervised classification is a broad field with many possible algorithms and measure to be used. A breakdown of the different measures identified in the sample is provided in Table VII. The following works focus on possible improvements to classification using supervised methods, and use variations on the following measure categories: power (p, r) distance, distances on distribution laws, distances on strings and permutations, and novel measures not previously defined.

Among the supervised classification methods using power (p, r) distance, there is a wider variety of classification types used. Gong, *et al.* [26] study a modification of the Negative Selection Algorithm using an additional training phase to minimize the number of self-samples required to cover the self-region, and reduce the false alarm rate greatly and the detection rate slightly. Ferreira *et al.* [34] investigate the use of wavelet analysis and Artificial Neural Network (ANN) for an IDS.

TABLE VII
MEASURES USED IN SUPERVISED CLASSIFICATION

Type	Measure	Count	References
Power (p, r)	Euclidean	3	[26], [34], [38], [49]
Distances on distribution laws	Probability	1	[29]
	Entropy	1	[31]
Distances on strings and permutations	Affinity $(LCS(x, y))$	1	[30]
	Swap metric	1	[54]
Novel	Anomaly Metric S_G	1	[58]
	Matching Degree	1	[55]
	c_j	1	[56]
	$sim(S_1, S_2)$	1	[57]

Yi *et al.* [38] introduce an improved incremental Support Vector Machine algorithm with a modified kernel function that weights data points according to their distance to the hyperplane, to indicate the likelihood that the point will be a support vector.

However, not all uses of power (p, r) distance are strictly Euclidean distance. Kou *et al.* [49] address the case of classifying multiple classes (more than two). A kernel method called multi-criteria mathematical programming is used to classify data with nonlinearities. The concept of distance to the hyperplane is required for the kernel method, and is specified as a formula that contains the l_2-norm.

There are two examples of distances on distribution laws among supervised classification methods. Altwaijry and Algarny [29] consider a Naïve Bayes classifier in which training data is processed to find the probability of each feature value occurring in the normal data. The calculated probability is then used as a threshold to determine whether new data is an attack or normal. In this case, probability calculations are utilized as the "distance measure" even though distance is not explicitly discussed. Arshadi and Jahangir [31] propose using the entropy of packet inter-arrival times within a sliding window to detect SYN flooding attacks, based on the concept that attack packets have lower entropy than normal packets. If the entropy in the current window is less than the mean entropy minus three standard deviations, then an attack is identified. Again, entropy is defined and used as a distance measure, but distance is not explicitly discussed.

Among distances on strings and permutations, there are a variety of uses. Most identified by this sample to fall into those techniques loosely labeled as artificial immune system. Antunes and Correia [30] investigate the immunological concept of Tunable Activation Thresholds. This is as opposed to the commonly-studied immunological concepts of negative selection and danger theory. In the study of Artificial Immune Systems, the distance concept is known as *affinity*. In this study, *affinity* is defined as the maximal length of the substring in common between the T-cell receptor (the detector element) and the peptide (the element of the data to be classified), or the similarity longest common substring between x and y $(LCS(x, y))$ where x and y are strings. Zhang *et al.* [54] study artificial immune systems and specify *affinity* as allowing for "any kind of Hamming, r-continuous matching, etc.". This study uses the Hamming matching algorithm to determine affinity. He and Parameswaran [58] work from the premise that anomalous connections from a single attacker are similar to each other. They develop a system that tests multiple connections for similarity within clustered groups, compare it to a set threshold, and mark everything above the threshold as anomalous. The devised novel similarity measure is called the Anomaly Metric S_G, and is related to distance measures on permutations or strings.

Some of the supervised classification examples are not able to use previously defined measures of distance and similarity, and devise new methods of measurements specific to the topic of the study. Mabu *et al.* [55] propose an IDS framework of generating class-association rules using fuzzy set theory combined with genetic network programming. In either misuse or anomaly detection two pools are generated to hold the association rules for normal and intrusion connections, and those rules are then applied to the classification of the dataset. A novel similarity measure is defined by the author called Matching Degree, which is used to determine whether a newly generated rule matches the rules known to be effective. Shyu and Sainani [56] propose a framework to be used in the development of IDSs using multiple classification techniques and multiple agents. The intent is to reduce the complexity involved in building IDSs and distribute the load throughout the network, rather than having all load at a single point. The distance measure is defined but only referenced as a "distance measure," and uses the eigenvectors and eigenvalues to generate a measure value, which is then compared to a threshold to determine abnormality or normality. Su *et al.* [57] propose a method of comparing fuzzy association rule sets generated using incremental data mining. A rule set is generated from incoming traffic, and another from attack-free training traffic. The similarity between the rule sets is used to determine the abnormality of traffic from which the rule sets are generated, with a decision made every two seconds. The similarity measure is a novel formulation defined for the purpose of evaluating the rules in this study.

B. Measure Undefined

Those works that did not provide specific identification or formulation of the measure used are covered below. The papers will be grouped as in Section VI-A, with the addition of one further category: works that focus on the entire system, presenting a new structure or framework instead of focusing on one phase.

1) System Focused: Roughly speaking, the works that focus on the entire system rather than one phase of the system can be grouped as follows: ensemble, multi-agent, and comparison of existing systems using some novel method to compare. A number of works focus on ensemble techniques in which the strengths of multiple methods of classification, whether unsupervised or supervised, are combined to produce

a (hopefully) more effective method of intrusion detection. At times the effectiveness is measured by detection rate, false positive rate, or in other ways. The main focus, regardless of how the improvement is measured, is to improve the results of classification, and reduce the number of falsely classified instances. The ensemble methodologies can be categorized into two general types: layered and voting.

Among the layered methods, there are a number that use two or more classification methods to improve results. Ali and Jantan [63] uses two layers to detect attacks: the first recognizes undesirable characteristics, then feeds any non-attacks to the second layer that recognizes desirable characteristics. After the number of detections made by the second layer pass a threshold they are clustered using k-means and the first layer is trained on the results. Xu *et al.* [116] make use of a "perceptron tree", which is a decision tree in which each node is a neural network. It is proposed in this study that this kind of combination might display the advantages of both symbolic and non-symbolic models. Mohammed and Awadelkarim [71] propose a NID framework using a decision tree to detect known attacks and two-step clustering to detect new attacks. Both KDD99 and real network data are used in the evaluation, and the framework is compared to the Minnesota Intrusion Detection System which uses Snort and local outlier factor clustering. Salama *et al.* [75] implement a method of intrusion detection using a Deep Belief Network (DBN) to perform feature reduction, then classifying the remaining features with a SVM. The authors also compare the DBN-SVM method with each as a standalone classifier, and compare DBN with other common feature reduction methods. The results are than DBN-SVM is more accurate than either alone, and DBN is more effective than Principal Component Analysis (PCA), Gain-Ratio, and Chi-Square as a feature reduction method. Zhang [84] proposes a neural network in which feature selection, network structure, and weight adaptation are all evolved in conjunction using a genetic algorithm as part of an improved evolutionary neural network. A heuristic mutation operator is used to prevent the search from ending in local optima, and allow full coverage of the search space. Sarvari and Keikha [92] propose the use of multiple machine learning methods to detect attacks in the Intrusion Detection System combinatory of Machine Learning Methods (IDSCML). The proposed method uses k-nearest neighbor, decision trees, neural networks and SVMs to take advantage of the benefits of both anomaly and misuse detection methods. The most useful method is the combination of decision tree, 1 nearest neighbor, 2 nearest neighbor, 3 nearest neighbor, and SVM run in parallel on the data, then combined using a neural network. Wang *et al.* [97] present a use of the Artificial Bee Colony algorithm to select both the free parameters of a SVM classifier and the features to use in classifying the KDD99 dataset. In every category of attack used in the experiments, the artificial bee colony method of free parameter and feature selection resulted in greater accuracy than either particle swarm optimization or genetic algorithm. Folino *et al.* [90] take an ensemble approach based on genetic programming. The approach is applied in a distributed manner in order to build a network profile. The approach is tested on the KDD99 dataset, and shows performance similar to the winning entry in the KDD99 competition.

Despite the prevalence of layered systems, there are a few voting systems which focused on the entire system rather than one phase. Panda and Patra [111] assemble the classifiers AdaBoost, MultiBoosting, and Bagging each combined with a decision tree pruned using reduced-error-pruning, and the results are then compared to other classifiers used within recent research. Zainal *et al.* [117] combine individual classifiers designed for detecting a single class, and each utilizing a different learning method. The classifiers then vote to determine the final classification. Feature selection is done using Rough Set Technique and Binary Particle Swarm in a 2-tier process. It is shown that the system performs better than the best-performing classifier alone. Farid *et al.* [67] merges Boosting (AdaBoost) with a Naïve Bayesian classifier. One classifier is created for each feature in the data, then all of the classifiers are used to calculate the probability of occurrence of an unseen data point and the classifiers vote to determine if the point is normal or anomalous.

Although layering may be a common solution to combining multiple Network Intrusion Anomaly Detection (NIAD) methods, some have tried multiple agents. Joldos and Muntean [69] introduce the idea of comparing the distances between feature vectors of datasets formed from subsets of large standard datasets: this study focuses on an investigation of the benefits of performing feature reduction and classification tasks for intrusion detection using grid computing. Zeng *et al.* [101] incorporates three primary components: intrusion detection node, intrusion detection coordinator, and snooper agent. The coordinator and agent are generally located on the same host, while there is at least one node on each Local Area Network (LAN) segment. The node captures and parses traffic, and passes new information back to the coordinator. The coordinator manages the information and signature databases, and the alert function. The agent is launched when new information is received from the node by the coordinator, and gathers information which may be needed about new attacks. Barika *et al.* [105] and [106] propose an architecture for an IDS using distributed mobile agents throughout the network with four types of agents: sniffer, filter, analyzer, and decision. The performance of the MA_IDS is tested using both port scan and SYN flood attacks, and by passing messages through all the agents. MA_IDS is also compared to the centralized detection system, and the agent detection system had less packet loss and shorter detection delays as the number of packets per second and overall packets increased. Rehak *et al.* [124] uses a multi-agent approach to detect anomalies and build a trust model. Each detection agent utilizes a single anomaly detection method and contributes to a trust model built collectively by the multiple agents. Attacks are determined by thresholding a "trust score", which in this context can be considered the distance measure. Yu *et al.* [83] present an intrusion detection model based on modular mobile agents. They use the Aglets environment to construct a mobile agent simulated environment, and use a Markov chain model as the intrusion classifier. A positive correlation is found between both the number of detectors and the length of data, and the True Positive Rate (TPR): increases in either the number of detectors or length of data resulted in an increase in TPR. Gao *et al.* [109] propose a distributed IDS framework

using both local and global detectors. The local detector is the Multiple Adaboost algorithm, in which the expectation maximization method is used to update the parameters of the detector. The global detector is a combination of particle swarm optimization and SVM. The results indicate parameter selection for the local detector is important in minimizing False Positive Rate (FPR), and show the global detector results in a detection rate of 99.99% and false positive rate of 0.3713%.

There are also works that focus on the comparison between different IDSs, or different studies of particular datasets. Engen *et al.* [66] compare the results of research into intrusion detection using decision trees and Naïve Bayes methods to classify the KDD99 dataset. The comparison is used to investigate discrepancies in the subsets of KDD99 used in research, and determine whether KDD99 is useful for further research or too flawed to be of use. Eljadi and Othman [65] demonstrate the use of three separate data mining techniques to detect anomalous traffic in a real world network traffic dataset. The results use a threshold value to determine whether segments of the examined dataset contain intrusion attempts. Each method is evaluated for suitability in intrusion detection by evaluating the time required to evaluate the dataset segment, and the ability to generate rules with which to update the Network Intrusion Detection System (NIDS) in use. Pastrana *et al.* [74] propose a framework to use in modeling existing NIDS. Genetic Programming is used to generate an instance that behaves like the modeled NIDS, but is much simpler, for the purpose of determining weak points. The created model allows researchers to discover new evasion methods, and provides a new method with which to audit the performance of commercial NIDS. Gogoi *et al.* [91] present a limited review of anomaly based NIDS, a few numeric and categorical proximity measures, and performance comparison of some supervised and unsupervised NIDS. This work discusses different proximity measures and their formulations, but does not mention the measure description or definition. Shafi and Abbass [77] propose a method of generating NIDS datasets with real background traffic and simulated attacks. They then compare a number of different intrusion detection algorithms with a dataset built by the authors according to the proposed method. Day and Burns [64] compare the performance of Snort and Suricata NIDS on multi-core computer systems. The performance of each is evaluated based on accuracy, false positives and negatives, system utilization, and dropped traffic. The conclusion is that while Suricata performs marginally better than Snort, the increased resources required by Suricata make Snort the better choice for an open source NIDS. Zhengbing *et al.* [125] propose an algorithm to find new attack signatures based on known attack signatures, using a variation of the Apriori algorithm (Signature Apriori) to find frequently-occurring flow patterns. The proposed algorithm is tested against Signature Apriori using SNORT. It is found that the proposed algorithm is more efficient than Signature Apriori with equal detection rates, in the case that the new attack is derived from an earlier attack. Fanelli [88] experiments on an immune system inspired IDS by comparing the detection results of Snort and Network Threat Recognition with Immune Inspired Anomaly Detection (NetTRIIAD). The two NIDS are compared using the KDD99 dataset in the categories of known attacks, unknown attacks, and ablation tests.

2) Selection: Many researchers focus on the selection phase as the primary method of reducing complexity in the field of NIAD: If the clustering or classification method has fewer features or observations to process, then the overall result is reached faster. Khor *et al.* [59] propose a method of splitting NID datasets based on the frequency of attack. Normal records are included in both, but one only includes rare attacks, while the other includes only non-rare attacks. In both datasets the authors reduced the number of records included to prevent a particular class from overwhelming the other purely by virtue of the number of records included. The experimental results suggest that splitting the datasets can improve the classification accuracy of some classes, but not all. Farid *et al.* [108] investigate a Naïve Bayesian classifier and Iterative Dichotomiser 3 algorithm for feature reduction, where KLD is used to select the best attributes. Ahmad *et al.* [61] and [62] propose two feature selection methods that apply genetic algorithms to the result of PCA. The fitness function used for the genetic algorithm uses multilayer perceptron accuracy in one paper, SVM accuracy in the other, and the number of features not selected for the particular subset of features in both papers. An accuracy of 0.99 is achieved with 12 of the 38 KDD99 features selected using multilayer perceptron, and 99.6% accuracy is achieved with SVM. Sen and Clark [76] examine the use of evolutionary computation to develop intrusion detection programs for mobile *ad hoc* networks. Genetic programming and grammatical evolution are used to evolve intrusion detection programs with the programs doing their own feature selection (all features are provided) for individual attacks, multiple attack, and finally cooperative attack detection. The fitness of any given solution is defined detection rate—false positive rate, and security versus power-used trade-offs are considered in the experiments because of the inherent limitations of mobile devices. Das *et al.* [86] propose the use of multiple machine learning methods to classify intrusion attempts on a network. The preprocessor extracts 14 features from network traffic every 4 seconds. The features extracted are then examined for use in classification by two methods: PCA and rough set theory. Based on a comparative study within the work, rough set theory is chosen as the more effective method of feature selection. The selected features are sent to the SVM to learn and classify. Al-Sharafat and Naoum [104] examine the importance of feature selection in detecting network attacks, and perform experiments to determine the most effective combination of features used in research. The set of features selected in four different works are used to classify network attacks. The classifiers used are generated by a steady state genetic based machine learning algorithm. Out of the four classes of features, one resulted in a detection rate higher than the other three: 97.5%. Shanmugam and Idris [112] introduce a variation of the data mining algorithm by Kuok and Apriori using fuzzy logic to create rules expressed as logic implications. This approach is used for feature reduction to determine the features which provide maximum KLD for classification of attacks. Singh and Silakari [114] introduce a Generalized Discriminant Analysis approach to feature reduction, then use an ANN for

classification. Generalized Discriminant Analysis (GDA) is a form of Linear Discriminant Analysis (LDA) in which the dataset is first transformed into a higher-dimensional space prior to the feature reduction, in order to more successfully process nonlinear datasets. The ANN approach is compared to a C4.5 decision-tree classification approach, and both are considered using both LDA and GDA for feature reduction. The GDA feature-reducer performs slightly better than the LDA with both the C4.5 and the ANN classifiers. Zaman and Karray [118] present a feature reduction study, in which attempt is made to improve the Enhancing Support Vector Decision Function (ESVDF) method by integrating it with a fuzzy inferencing model. This approach simplifies design complexity and reduces execution time. The approach is tested on KDD99, comparing fuzzy ESVDF to other types of ESVDF as feature selectors. Neural Network and Support Vector Machine classifiers are used to classify the data post-feature reduction. Accuracy is found to be best with the proposed method, with decreased training times. Zargar and Kabiri [120] apply Principal Component Analysis to reduce features in the KDD99 dataset in order to detect smurf-type attacks. The k-nearest neighbor method is used as the classifier for feature-reduced data. Abdulla *et al.* [85] suggest a set of features to use when applying artificial neural networks to the problem of intrusion detection. The affect caused by using different types and numbers of vectors as input to the neural network is explored, as is the effect of changing the number of epochs for the run. The authors demonstrate that the detection rate increases as the number of epochs or vectors increase.

3) Clustering: Clustering is usually an efficient method of discovering groups among data that the researcher may not know about, and it lends itself to the use of explicit measures of distance and similarity. That may be the reason so few (relatively) of the sampled works focus on clustering but do not specify a distance measure. Niemalä [72] classifies using a k-means clustering approach. Song *et al.* [78] present a clustering method followed by use of a support vector machine. In this study, a previously-unseen data point is determined to be anomalous if it is "inside the hypersphere". Zhenying [121] compares a single-layer to a multiple layer self-organizing map, and shows that the multiple layer map is unable to improve performance. It is proposed that the reason for the lack of improvement is due to the overlap between different classes of data. Tarannum and Lamble [79] simulate a hybrid IDS running on mobile devices. A combination of on demand clustering (using the Ad Hock On Demand Vector routing protocol) and neighbor information collection (using the Destination Sequenced Distance Vector routing protocol) are used in addition to host-based NIDS to detect intrusions. The Mobile Ad Hoc Networks (MANET) is clustered with each cluster having a *head* that is responsible for the intrusion detection for that cluster. In the case where the *head* suspects an intrusion, but needs more evidence, it engages the nodes in the cluster to perform cooperative intrusion detection. The feasibility of a hybrid cooperative approach was demonstrated by the simulation.

4) Supervised Classification: Supervised classification is the most reliable method of classification, as it allows the researcher or operator to determine what is normal or anomalous, and train the classifier on the results of those decisions. One type of method used by researchers is the tree. Farid *et al.* [89] attempt to reduce the rate of false positives by using a decision tree-based attribute weighting for feature selection and an adaptive Naïve Bayesian tree for classification. Srinivasulu *et al.* [95] utilize a frequent pattern tree rule-learning algorithm to learn normal customer behavior in a transaction database. The study defines two measures for association rules: *support* and *confidence*, which serve the purpose of distance measures. *Support* is the ratio of transactions containing all specified feature values to the total number of transactions analyzed, and *Confidence* is the ratio of transactions containing one specified feature value to the total number of transactions analyzed. Hu *et al.* [123] apply Adaboost, a learning algorithm that combines weak classifiers, to intrusion detection. They use "decision stumps", decision trees with one root and two leaf nodes, as the weak classifiers. This approach is shown to be less computationally complex compared to ANN and SVM classifiers, while the detection rate and false-alarm rate is comparable. Visumathi and Shunmuganathan [82] propose a new architecture for IDS which uses misuse detection (effectively signature detection) to identify attacks, uses the Apriori algorithm on the known attack signatures to generate a set of patterns, compares the known attack patterns to the probable attack patterns, then adds the attack signature for any probable attack pattern that has a similarity of greater than 0.9 to any known attack pattern. The authors obtained a set of results using a myriad of machine learning methods including SVM, and found that the random forest algorithm gave the highest accuracy with 99.97, but do not provide any information about the false alarm rate.

Another method is modeled after the immune system, sometimes labeled as an artificial immune system. Fanelli [122] presents a hybrid immune inspired IDS: NetTRIIAD. The system uses two layers to allow for misuse and anomaly based detection. The innate layer performs misuse detection for existing threats, ensuring that known attacks will be caught. The adaptive layer performs self-nonself discrimination by imitating the body's response to intruders using T cells. When compared with Snort, the NetTRIIAD system has equivalent TPR, and a significantly lower FPR, thus its positive predictive value (0.65) is almost double Snort's (0.38). Xiao-Pei and Hou-Xiang [99] propose and demonstrate an experimental IDS that uses an immune system inspired detection model, with detectors generated by both randomly and using a genetic algorithm to produce immature detectors. When attacks are detected features are extracted and codified for use within a vaccine detector. A fitness function using the TPR and FPR is used to determine the probability of a particular detector becoming a parent. The proposed method is shown to be more effective in both TPR and FPR at detection of all types of attacks in the KDD99 dataset when compared to a classic immunity detection model. Zamani *et al.* [119] use the concept of "danger theory" from biological immunology in which cells identify foreign cells which need to be attacked based upon signals sent out by dying self-cells. They define a "costimulation concentration level" as a weighted sum of "sends" and "receives" from other

network nodes. This approach is used to detect Distributed Denial of Service (DDoS) attacks and is tested on a simulated network.

The imitation of human systems to detect attacks is not new, as neural networks have been simulated for many years. The use of ANN has resulted in advances in the intrusion detection field and many others. Seliya and Khoshgoftaar [93] present an active learning procedure using neural networks. The performance of the new procedure is then compared with the results of applying C4.5 decision tree to the same dataset (KDD99). The overall results indicate that the active learning method is better able to generalize when detecting intrusions than the C4.5 method. Norouzian and Merati [73] use a two layer neural network to classify attacks as separate types based on network traffic. Many works treat classification as a two-class problem where a record is either an attack, or normal. Instead the attack classes are used as recorded by the creators of the KDD99 dataset. Sheikhan *et al.* [94] experiment with both multi-layer perceptrons and Elman neural networks to classify a portion of the KDD99 dataset. The classifiers are evaluated according to detection rate and false alarm rate, and the authors also propose a Cost Per Example formula to use in evaluating classifiers. It is shown that a multi-layer perceptron with 15 inputs performs better than an equivalent Elman neural network and the top two performers in the 2000 KDD competition. Wang *et al.* [98] experiment with the use of a fuzzy based feed forward neural network for Denial of Service (DoS) attack detection. Fuzzy sets are used to define the eight features to be used in detecting DoS attacks, then to convert a portion of the KDD99 dataset to the form of the new fuzzy features. A portion of the converted features is used to train a two-layer neural network over 276 epochs. Ahmad *et al.* [103] demonstrate the use of a back propagation neural network as an IDS for DoS attacks. Root mean-squared-error is used to evaluate the performance of a variety of hidden layer configurations, with the lowest value chosen as the configuration for the experiment. Sheikhan and Sha'bani [113] utilize a neural-network (multi-layer perceptron) approach, in which they attempt to improve the speed of training by using an "output weight optimization-hidden weight optimization" training algorithm. The results show an increase in training speed without loss of classification accuracy relative to the year 2000 winner of the KDD99 competition. Tian and Gao [115] apply a Genetic Algorithm to the back propagation process in a multilayer perceptron to improve the speed of convergence of the network. The approach is applied to a dataset based on the MIT Lincoln Laboratory dataset. The back propagation with genetic algorithm approach results in a reduced mean squared error of anomaly detection compared to a standard back propagation algorithm. Orfila *et al.* [110] explore the application of genetic programming to the problem of NID, in particular to the automatic creation of rules and patterns to use in the detection of attacks. The genetic programming approach to rule building is compared to the use of C4.5, and found to be both simpler and to require fewer operations per Transmission Control Protocol (TCP) packet.

SVMs are another useful tool in the supervised classification toolbox. SVM allows low dimensional data to be extrapolated into higher dimensions for the purpose of classifying each observation, by using a hyperplane as the separator for the groups of data. Abdulla *et al.* [60] propose the use of SVM to classify NetFlow Data and provide warning of worm attacks. They demonstrate false positive and negative rates from 0.09 to 0.00 after refining the data to remove non-existent nodes, whereas the rates are between 0.10 and 0.28 without refinement. He [68] proposes the use of the Relevance Vector Machine (RVM) using a logistic chaotic map instead of the standard Gaussian as an estimate of the noise in the output signal. The results of classifying the KDD99 dataset with both RVM and SVM are compared using receiver operating characteristic curves and required number of vectors. The results indicate that the RVM has lower false alarm rates at a given detection probability, and generates fewer vectors to handle a similar number of records.

Those that do not fit into one of the neat categories above are also worth examination. Kahn and Burney [70] propose an intrusion detection system consisting of a Finite State Machine (FSM) that uses Push Down Automata (PDA) to perform attack-instance storage. The results of using FSM as a NID are not clearly demonstrated, as the accuracy and false positive rates are not specified. Vijayasarathy *et al.* [81] propose a lightweight DoS classifier framework to operate on both the TCP and User Datagram Protocol (UDP) protocols. The framework uses packet windowing to split input traffic into subsets, uses the TCP flags to define six categories T_1 through T_6, and cross-validation to determine the threshold beyond which an attack is assumed to be taking place. Accuracy increased and false alarm rate decreased as the window size increased, however the authors note the threshold will have to be set by an experienced network administrator. Faizal *et al.* [87] proposes a method of detecting anomalies based on the number of connections during a one second period. The results of this detection are compared assuming all traffic is normal. The detection rate increases to 85.9%, while the false positive rate increases to 3.2%. Torrano-Gimenez *et al.* [96] propose a new web application firewall to detect attacks on a web-based application. The firewall is provided with an Extensible Markup Language (XML) file containing a thorough description of the web application's normal behavior, and thresholds providing some flexibility in the definition of *normal* behavior for web applications. Any traffic exceeding the thresholds are considered to be an attack. The performance of the firewall using an XML normal operation description is excellent, detecting all attempted web attacks. However, the authors note that automated description of the normal operation descriptions would be necessary for implementation on a large scale. Ye *et al.* [100] propose an anomaly detection system using a simple Hidden Markov Model (HMM). The assumption is made that all network behaviors are normal within a given time window. If the network behavior deviates from the normal behaviors, then the behavior is assumed to be an attack. The HMM is found to be capable of detecting attacks, although not the type of attack. Tuncer and Tatar [80] propose an embedded system to detect DoS attacks in real time. The embedded system uses a programmable system on a chip to train on traffic patterns and generate alarms during the test phase. Boolean association

rules are derived during the training phase based on five traffic attributes, and the chip is then programmed to recognize traffic matching those rules as DoS attacks. Changguo *et al.* [107] modify the standard Apriori algorithm with fuzzy association rules mining, and apply the technique to wireless network intrusion detection. Experimentation shows a reduction in the number of candidate "itemsets" with the proposed method. Zhu *et al.* [102] present an attack on semi-supervised classifiers by injecting fake unlabeled instances. An algorithm to generate misleading instances is provided, influence on the classifier is demonstrated, and a possible defense involving self-training by comparing labeled and unlabeled instances is proposed. Misleading instances are found to reduce the accuracy of the Naïve Bayes and self-trained-based Naïve Bayes classifiers, but self-training attenuated the decrease in accuracy.

Section VII presents the pitfalls and lessons learned during this review of the field. It also provides exemplary papers which the reader can use as templates for improving their own future work.

VII. Lessons Learned

Trends in publication of research in the Network Intrusion Detection (NID) field explored in Section VI reveal several areas for improvement. The four common pitfalls uncovered are:

1) Failure to identify distance measures selected
2) Failure to provide adequate details (such as mode or parameterization) on selected distance measures
3) Failure to explain why the distance measure(s) are chosen
4) Failure to treat distance measure and parameterization as another experimental factor in performance evaluation

When authors fall prey to these pitfalls, experiment repeatability and validation is compromised. Furthermore, benefits discovered about the distance measures during the research cannot be fully realized by the field. In the remainder of this section, these pitfalls are described in detail and mitigation strategies are discussed. For each category we also suggest exemplars in the field which demonstrate the recommended guidance for future research.

The first pitfall is that identification of distance measures are often missing. In the field sample of 100 papers, only 40% identified the name of the distance measure used. Since distance measure choice is central to algorithm performance in anomaly detection, researchers should clearly identify which distance measures they use, and identify how they are using them. Researchers should also use standardized names for distance measures, and we recommend the Encyclopedia of Distances [4] which will assist this endeavor. Examples of papers which do this very well are Gong's paper with a description of a sum of squares distance used for thresholding [26] and Borah's paper describing the use of a power (p, r)-distance for a K-nearest neighbors algorithm [28]. In each of these papers, the authors clearly identify the distance measures they used and provide formulas.

Second, while some authors indicate which distance measures are used, far fewer give the implementation details such as formulas or parameter settings. In the field sample, only 32% gave a specific mathematical formulation for their distance measure. Parameter settings are also important in some of the distance measures, but are often overlooked when authors present their research. For example, when using power (p, r)-distance, p and r can be selected independently, but are often assumed to be the same, as in the Euclidean distance formulation $(p = r = 2)$. Another example of potential ambiguity is revealed when calculating the various Entropy-related distances, where distance depends on the selected log base. In most cases, a default log base of 2 is assumed, but it is possible to select another base. Another pitfall of in the category is the use of "default settings" of an off-the-shelf algorithm with the assumption that those default settings will remain static in perpetuity. Machine learning libraries are frequently updated, and default settings used during the time of research may have been changed by the time the research is published. An even more dangerous practice occurs when researchers do not realize that when they choose an off-the-shelf algorithm without fully understanding the implementation details, they may not realize that default settings have been selected for them—these settings may be even be sub-optimal for the phenomenon they are studying.

An exemplar paper in this category is by Arshadi and Jahangir [31], which identified the distance measure, provided a formula, and provided detailed parameter and variable descriptions for the use of entropy as a distance measure to determine randomness of the inter-arrival rate of packets. This level of documentation greatly facilitates recreating the experiment for validation and comparison with future research.

Third, even when implementation details are known and provided, researchers often fail to indicate why the choices are made. A failure to explain these choices leads to a missed opportunity to pass on important learning opportunities to the reader. This behavior can slow the general progress of advancement in the field.

Palmieri provides an exemplar in this category: he explains why he uses squared-Euclidean distance as the measure for determining nearest- and false-nearest neighbors as feature-space dimensions are increased one-by-one [44].

Finally, unlike the exemplary articles described in Section V, the sample indicates that in the vast majority of research a single distance or similarity measure is selected, and the authors do not explore the tradespace of distance measure alternatives or even alternative parameter settings. In our sample of the field, only two authors (2%) explored distance measure choice as a factor in experiment design.

While most research didn't evaluate more than one distance measure for a single anomaly detection phase, the two treatments we reviewed seem promising, since they provided a framework that could be repeated with distance measures the authors did not use. One article by Obimbo explores the performance of classification using two measures - Euclidean distance, and a custom measure based on a voting system [37]. Another exemplar comparison study reviewed performance of three measures of entropy for determining the best features to use [35]. These authors are paving the way by using distance measure as another factor in experiment design. We recommend future research follow this lead.

VIII. Conclusion

Every experiment which utilizes Anomaly Detection (AD) in the Network Intrusion Detection (NID) field uses distance measures, most without much thought as to which distance measure would be most appropriate. However, it is clear that sometimes similarity and distance measures are used with care in research. There are great examples to provide guidance and to be expanded upon. We recommend adopting the following habits to improve the quality of the research:

- Clearly name and describe all distance measures and parameters used throughout the research endeavor. When present in the Encyclopedia of Distances [4] standardize the measure's name.
- Borrow distance measurement methods from other fields which have similar challenges, such as natural language processing, and examine solutions that are unused in the NID field.
- Consider exploring the use of different distance and similarity measures as part of the experiment to determine how they affect detection rate.
- Incorporate flexible methods for capturing and expressing data values to make distance measures comparable. For example, build a graph distance matrix (discussed in Section VII), as used by Tan *et al.* [23] as a method of comparison.
- Develop techniques to compare graph distance matrices without visualization, and determine of which thresholds are most useful under certain conditions.
- Gain a better understanding of how distance measures challenge the intuitive understanding of the term *closest*, and develop visualizations and simulations to aid in that understanding how to set parameters appropriate for the problem space.

Acknowledgment

The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[2] A. Chmielewski and S. Wierzchoń, "On the distance norms for detecting anomalies in multidimensional datasets," *Zeszyty Naukowe Politechniki Białostockiej*, vol. 2, pp. 39–49, 2007.

[3] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.

[4] M. Deza and E. Deza, *Encyclopedia of Distances*. Berlin, Germany: Springer-Verlag, 2009.

[5] S. Staab, "Ontologies and similarity," in *Case-Based Reasoning Research and Development*, vol. 6880. Berlin, Germany: Springer-Verlag, 2011, ser. Lecture Notes in Computer Science, pp. 11–16.

[6] P. Cunningham, "A taxonomy of similarity mechanisms for case-based reasoning," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 11, pp. 1532–1543, Nov. 2009.

[7] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 198–211, Jul./Aug. 2013.

[8] F. Aurenhammer, "Voronoi diagrams—A survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, no. 3, pp. 345–405, Sep. 1991.

[9] P. C. Mahalanobis, "On the generalised distance in statistics," *Proc. Nat. Inst. Sci. India*, vol. 2, no. 1, pp. 49–55, Jan. 1936.

[10] K. Wang and S. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, vol. 3224. Berlin, Germany: Springer-Verlag, 2004, ser. Lecture Notes in Computer Science, pp. 203–222.

[11] S. Teng, H. Du, W. Zhang, X. Fu, and X. Li, "A cooperative network intrusion detection based on heterogeneous distance function clustering," in *Proc. 14th Int. Conf. Comput. Supported Coop. Work Des.*, Apr. 2010, pp. 140–145.

[12] D. Hancock and G. Lamont, "Multi agent system for network attack classification using flow-based intrusion detection," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2011, pp. 1535–1542.

[13] J. Cho, K. Choi, T. Shon, and J. Moon, "A network data abstraction method for data set verification," in *Secure and Trust Computing, Data Management and Applications*, vol. 186. Berlin, Germany: Springer-Verlag, Jun. 2011, ser. Communications in Computer and Information Science, pp. 54–62.

[14] Y. Wang, Z. Zhang, L. Guo, and S. Li, "Using entropy to classify traffic more deeply," in *Proc. 6th IEEE Int. Conf. Netw., Architecture Storage*, Jul. 2011, pp. 45–52.

[15] K. Rieck and P. Laskov, "Language models for detection of unknown attacks in network traffic," *J. Comput. Virol.*, vol. 2, no. 4, pp. 243–256, Feb. 2007.

[16] H. Sengar, X. Wang, H. Wang, D. Wijesekera, and S. Jajodia, "Online detection of network traffic anomalies using behavioral distance," in *Proc. 17th Int. Workshop Qual. Serv.*, Jul. 2009, pp. 1–9.

[17] Z. Lu and T. Peng, "The VoIP intrusion detection through a LVQ-based neural network," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, Nov. 2009, pp. 1–6.

[18] H. Kwon, T. Kim, S. Yu, and H. Kim, "Self-similarity based lightweight intrusion detection method for cloud computing," in *Intelligent Information and Database Systems*, vol. 6592. Berlin, Germany: Springer-Verlag, 2011, ser. Lecture Notes in Computer Science, pp. 353–362.

[19] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *Proc. 15th Annu. Netw. Distrib. Syst. Security Symp.*, 2008, pp. 1–18.

[20] T. Krueger, C. Gehl, K. Rieck, and P. Laskov, "An architecture for inline anomaly detection," in *Proc. Eur. Conf. Comput. Netw. Defense*, Dec. 2008, pp. 11–18.

[21] H. F. Eid, M. A. Salama, A. E. Hassanien, and T.-H. Kim, "Bi-layer behavioral-based feature selection approach for network intrusion classification," in *Security Technology*, vol. 259. Berlin, Germany: Springer-Verlag, Dec. 2011, ser. Communications in Computer and Information Science, pp. 195–203.

[22] A. Chmielewski and S. T. Wierzchoń, "An immune approach to classifying the high-dimensional datasets," in *Proc. Int. Multiconf. Comput. Sci. Inf. Technol.*, Oct. 2008, pp. 91–96.

[23] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. Liu, "Multivariate correlation analysis technique based on Euclidean distance map for network traffic characterization," in *Information and Communications Security*, vol. 7043. Berlin, Germany: Springer-Verlag, Nov. 2011, ser. Lecture Notes in Computer Science, pp. 388–398.

[24] G. Zhao, J. Yang, G. Hura, L. Ni, and S.-H. Huang, "Correlating TCP/IP interactive sessions with correlation coefficient to detect stepping-stone intrusion," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, May 2009, pp. 546–551.

[25] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. Conf. Appl., Technol., Architectures, Protocols Comput. Commun.*, Aug. 2005, pp. 217–228.

[26] M. Gong, J. Zhang, J. Ma, and L. Jiao, "An efficient negative selection algorithm with further training for anomaly detection," *Knowl.-Based Syst.*, vol. 30, pp. 185–191, Jun. 2012.

[27] Y. Li *et al.*, "An efficient intrusion detection system based on support vector machines and gradually feature removal method," *Exp. Syst. Appl.*, vol. 39, no. 1, pp. 424–430, Jan. 2012.

[28] S. S. S. Sindhu, S. Geetha, and A. Kannan, "Decision tree based light weight intrusion detection using a wrapper approach," *Exp. Syst. Appl.*, vol. 39, no. 1, pp. 129–141, Jan. 2012.

[29] H. Altwaijry and S. Algarny, "Multi-layer Bayesian based intrusion detection system," in *Proc. WCECS*, 2011, vol. II, pp. 918–922.

[30] M. Antunes and M. Correia, "Tunable immune detectors for behaviour-based network intrusion detection," in *Artificial Immune Systems*,

vol. 6825. Berlin, Germany: Springer-Verlag, 2011, ser. Lecture Notes in Computer Science, pp. 334–347.
[31] L. Arshadi and A. Jahangir, "Entropy based SYN flooding detection," in *Proc. IEEE 36th Conf. LCN*, Oct. 2011, pp. 139–142.
[32] S. Borah, S. P. K. Chetry, and P. K. Singh, "Hashed-K-means: A proposed intrusion detection algorithm," in *Computational Intelligence and Information Technology*, vol. 250. Berlin, Germany: Springer-Verlag, 2011, ser. Communications in Computer and Information Science, pp. 855–860.
[33] N. Devarakonda, S. Pamidi, V. Valli Kumari, and A. Govardhan, "Outliers detection as network intrusion detection system using multi layered framework," in *Advances in Computer Science and Information Technology*, vol. 131. Berlin, Germany: Springer-Verlag, 2011, ser. Communications in Computer and Information Science, pp. 101–111.
[34] E. Ferreira, G. Carrijo, R. de Oliveira, and N. de Souza Araujo, "Intrusion detection system with wavelet and neural artificial network approach for networks computers," *IEEE Latin America Trans. (Revista IEEE America Latina)*, vol. 9, no. 5, pp. 832–837, Sep. 2011.
[35] C. Ferreira Lemos Lima, F. Assis, and C. de Souza, "A comparative study of use of Shannon, Rényi and Tsallis entropy for attribute selecting in network intrusion detection," in *Proc. IEEE Int. Workshop Meas. Netw.*, Oct. 2011, pp. 77–82.
[36] P. Gogoi, B. Borah, and D. Bhattacharyya, "Network anomaly detection using unsupervised model," *Int. J. Comput. Appl.—(Special Issue Netw. Security Cryptogr.)*, no. 1, pp. 19–30, Dec. 2011.
[37] C. Obimbo, H. Zhou, and R. Wilson, "Multiple SOFMs working cooperatively in a vote-based ranking system for network intrusion detection," *Proc. Comput. Sci.—(Special Issue Complex Adaptive Syst.)*, vol. 6, pp. 219–224, 2011.
[38] Y. Yi, J. Wu, and W. Xu, "Incremental SVM based on reserved set for network intrusion detection," *Exp. Syst. Appl.*, vol. 38, no. 6, pp. 7698–7707, Jun. 2011.
[39] H. Zheng, M. Hou, and Y. Wang, "An efficient hybrid clustering-PSO algorithm for anomaly intrusion detection," *J. Softw.*, vol. 6, no. 12, pp. 2350–2360, Dec. 2011.
[40] I. Brahmi, S. Yahia, and P. Poncelet, "MAD-IDS: Novel intrusion detection system using mobile agents and data mining approaches," in *Intelligence and Security Informatics*, vol. 6122. Berlin, Germany: Springer-Verlag, 2010, ser. Lecture Notes in Computer Science, pp. 73–76.
[41] X. Cheng and S. Wen, "A real-time hybrid intrusion detection system based on principle component analysis and self organizing maps," in *Proc. 6th Int. Conf. Natural Comput.*, Aug. 2010, vol. 3, pp. 1182–1185.
[42] M. Hayat and M. Hashemi, "An adaptive DCT based intrusion detection system," in *Proc. Int. Symp. Comput. Netw. Distrib. Syst.*, 2010, pp. 1–6.
[43] L. F. Lu, M. L. Huang, M. Orgun, and J. W. Zhang, "An improved wavelet analysis method for detecting DDoS attacks," in *Proc. 4th Int. Conf. Netw. Syst. Security*, Sep. 2010, pp. 318–322.
[44] F. Palmieri and U. Fiore, "Network anomaly detection through nonlinear analysis," *Comput. Security*, vol. 29, no. 7, pp. 737–755, Oct. 2010.
[45] K. Bharti, S. Shukla, and S. Jain, "Intrusion detection using unsupervised learning," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 5, pp. 1865–1870, 2010.
[46] Z. Tan, A. Jamdagni, X. He, and P. Nanda, "Network intrusion detection based on LDA for payload feature selection," in *Proc. IEEE GLOBECOM Workshops*, Dec. 2010, pp. 1545–1549.
[47] Y. Yang and J. Mi, "Design and implementation of distributed intrusion detection system based on honeypot," in *Proc. 2nd Int. Conf. Comput. Eng. Technol.*, Apr. 2010, vol. 6, pp. V6-260–V6-263.
[48] R. C. Chen, K.-F. Cheng, and C.-F. Hsieh, "Using rough set and support vector machine for network intrusion detection," *Int. J. Netw. Security Appl.*, vol. 1, no. 1, pp. 1–13, Apr. 2009.
[49] G. Kou, Y. Peng, Z. Chen, and Y. Shi, "Multiple criteria mathematical programming for multi-class classification and application in network intrusion detection," *Inf. Sci.*, vol. 179, no. 4, pp. 371–381, Feb. 2009.
[50] S. Singh and S. Silakari, "An ensemble approach for feature selection of cyber attack dataset," *Int. J. Comput. Sci. Inf. Security*, vol. 6, no. 2, pp. 297–302, 2009.
[51] J. Wang, Q. Yang, and D. Ren, "An intrusion detection algorithm based on decision tree technology," in *Proc. Asia-Pac. Conf. Inf. Process.*, Jul. 2009, vol. 2, pp. 333–335.
[52] Z. Zhuang, Y. Li, and Z. Chen, "PAIDS: A proximity-assisted intrusion detection system for unidentified worms," in *Proc. 33rd Annu. IEEE Int. Comput. Softw. Appl. Conf.*, Jul. 2009, vol. 1, pp. 392–399.
[53] T. Chou, K. Yen, and J. Luo, "Network intrusion detection design using feature selection of soft computing paradigms," *Int. J. Comput. Intell.*, vol. 4, no. 3, pp. 196–208, Jul. 2008.
[54] C. Zhang, J. Zhang, S. Liu, and Y. Liu, "Network intrusion active defense model based on artificial immune system," in *Proc. 4th Int. Conf. Natural Comput.*, Oct. 2008, vol. 1, pp. 97–100.
[55] S. Mabu, C. Chen, N. Lu, K. Shimada, and K. Hirasawa, "An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 1, pp. 130–139, Jan. 2011.
[56] M.-L. Shyu and V. Sainani, "A multiagent-based intrusion detection system with the support of multi-class supervised classification," in *Data Mining and Multiagent Integration*. New York, NY, USA: Springer-Verlag, 2009, pp. 127–142.
[57] M.-Y. Su, G.-J. Yu, and C.-Y. Lin, "A real-time network intrusion detection system for large-scale attacks based on an incremental mining approach," *Comput. Security*, vol. 28, no. 5, pp. 301–309, Jul. 2009.
[58] X. He and S. Parameswaran, "MCAD: Multiple connection based anomaly detection," in *Proc. 11th IEEE Int. Conf. Commun. Syst.*, Nov. 2008, pp. 999–1004.
[59] K.-C. Khor, C.-Y. Ting, and S. Phon-Amnuaisuk, "A cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection," *Appl. Intell.*, vol. 36, no. 2, pp. 320–329, Mar. 2012.
[60] S. A. Abdulla, S. Ramadass, A. Altaher, and A. A. Nassiri, "Setting a worm attack warning by using machine learning to classify NetFlow data," *Int. J. Comput. Appl.*, vol. 36, no. 2, pp. 49–56, Dec. 2011.
[61] I. Ahmad, A. Abdullah, A. Alghamdi, K. Alnfajan, and M. Hussain, "Intrusion detection using feature subset selection based on MLP," *Sci. Res. Essays*, vol. 6, no. 34, pp. 6804–6810, Dec. 2011.
[62] I. Ahmad, A. Abdullah, A. Alghamdi, and M. Hussain, "Optimized intrusion detection mechanism using soft computing techniques," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2187–2195, Apr. 2013.
[63] G. A. Ali and A. Jantan, "A new approach based on honeybee to improve intrusion detection system using neural network and bees algorithm," in *Software Engineering and Computer Systems*, vol. 181. Berlin, Germany: Springer-Verlag, 2011, ser. Communications in Computer and Information Science, pp. 777–792.
[64] D. Day and B. Burns, "A performance analysis of Snort and Suricata network intrusion detection and prevention engines," in *Proc. 5th Int. Conf. Digit. Society*, Feb. 2011, pp. 187–192.
[65] E. Eljadi and Z. Othman, "Anomaly detection for PTM's network traffic using association rule," in *Proc. 3rd Conf. Data Mining Optim.*, Jun. 2011, pp. 63–69.
[66] V. Engen, J. Vincent, and K. Phalp, "Exploring discrepancies in findings obtained with the KDD cup '99 data set," *Intell. Data Anal.*, vol. 15, no. 2, pp. 251–276, Apr. 2011.
[67] D. M. Farid, M. Z. Rahman, and C. M. Rahman, "Adaptive intrusion detection based on boosting and naïve Bayesian classifier," *Int. J. Comput. Appl.*, vol. 24, no. 3, pp. 12–19, Jun. 2011.
[68] D. He, "Improving the computer network intrusion detection performance using the relevance vector machine with Chebyshev chaotic map," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 1584–1587.
[69] M. Joldos and I. Muntean, "Distributed investigations of intrusion detection data on the grid," in *Proc. 10th RoEduNet Int. Conf.*, Jun. 2011, pp. 1–4.
[70] D. A. Khan and D. Burney, "Efficient FSM techniques for IDS to reduce the system attacks," *Int. J. Comput. Appl.*, vol. 29, no. 11, pp. 42–47, Sep. 2011.
[71] R. Mohammed and A. Awadelkarim, "Design and implementation of a data mining-based network intrusion detection scheme," *Asian J. Inf. Technol.*, vol. 10, no. 4, pp. 136–141, 2011.
[72] A. Niemelä, "Traffic analysis for intrusion detection in telecommunications networks," M.S. thesis, Tampere Univ. Technol., Tampre, Finland, Mar., 2011.
[73] M. Norouzian and S. Merati, "Classifying attacks in a network intrusion detection system based on artificial neural networks," in *Proc. 13th Int. Conf. Adv. Commun. Technol.*, Feb. 2011, pp. 868–873.
[74] S. Pastrana, A. Orfila, and A. Ribagorda, "A functional framework to evade network IDS," in *Proc. 44th Hawaii Int. Conf. Syst. Sci.*, Jan. 2011, pp. 1–10.
[75] M. Salama, H. Eid, R. Ramadan, A. Darwish, and A. Hassanien, "Hybrid intelligent intrusion detection scheme," in *Soft Computing in Industrial Applications*, vol. 96. Berlin, Germany: Springer-Verlag, 2011, ser. Advances in Intelligent and Soft Computing, pp. 293–303.
[76] S. Sen and J. A. Clark, "Evolutionary computation techniques for intrusion detection in mobile ad hoc networks," *Comput. Netw.*, vol. 55, no. 15, pp. 3441–3457, Oct. 2011.

[77] K. Shafi and H. Abbass, "Evaluation of an adaptive genetic-based signature extraction system for network intrusion detection," *Pattern Anal. Appl.*, vol. 16, no. 4, pp. 549–566, Nov. 2013.

[78] J. Song, H. Takakura, Y. Okabe, and K. Nakao, "Toward a more practical unsupervised anomaly detection system," *Inf. Sci.*, vol. 231, pp. 4–14, May 2011.

[79] R. Tarannum and M. Lamble, "Hybrid approach: Detection of intrusion in MANET," *IJCA Proc. Innov. Conf. Embedded Syst., Mobile Commun. Comput.*, vol. ICEMC2, no. 1, pp. 24–28, Sep. 2011.

[80] T. Tuncer and Y. Tatar, "Detection DoS attack on FPGA using fuzzy association rules," in *Proc. IEEE 10th Int. Conf. Trust, Security Privacy Comput. Commun.*, Nov. 2011, pp. 1271–1276.

[81] R. Vijayasarathy, S. Raghavan, and B. Ravindran, "A system approach to network modeling for DDoS detection using a naïve Bayesian classifier," in *Proc. 3rd Int. Conf. Commun. Syst. Netw.*, Jan. 2011, pp. 1–10.

[82] J. Visumathi and K. Shunmuganathan, "A computational intelligence for evaluation of intrusion detection system," *Indian J. Sci. Technol.*, vol. 4, no. 1, pp. 40–45, Jan. 2011.

[83] W. Yu, C. Xiaohui, and W. Sheng, "Anomaly network detection model based on mobile agent," in *Proc. 3rd Int. Conf. Meas. Technol. Mechatron. Autom.*, Jan. 2011, vol. 1, pp. 504–507.

[84] B. Zhang, "A heuristic genetic neural network for intrusion detection," in *Proc. Int. Conf. Internet Comput. Inf. Serv.*, Sep. 2011, pp. 510–513.

[85] S. Abdulla, N. Al-Dabagh, and O. Zakaria, "Identify features and parameters to devise an accurate intrusion detection system using artificial neural network," *World Acad. Sci., Eng. Technol.*, vol. 46, no. 70, pp. 627–631, Oct. 2010.

[86] V. Das *et al.*, "Network intrusion detection system based on machine learning algorithms," *Int. J. Comput. Sci. Inf. Technol.*, vol. 2, no. 6, pp. 138–151, Dec. 2010.

[87] M. Faizal *et al.*, "Time based intrusion detection on fast attack for network intrusion detection system," in *Proc. 2nd Int. Conf. Netw. Appl. Protocols Serv.*, Sep. 2010, pp. 148–152.

[88] R. Fanelli, "Further experimentation with hybrid immune inspired network intrusion detection," in *Artificial Immune Systems*, vol. 6209. Berlin, Germany: Springer-Verlag, Jul. 2010, ser. Lecture Notes in Computer Science, pp. 264–275.

[89] D. M. Farid and M. Z. Rahman, "Attribute weighting with adaptive NBTree for reducing false positives in intrusion detection," *Int. J. Comput. Sci. Inf. Security*, vol. 8, no. 1, pp. 19–26, 2010.

[90] G. Folino, C. Pizzuti, and G. Spezzano, "An ensemble-based evolutionary framework for coping with distributed intrusion detection," *Genetic Programm. Evol. Mach.*, vol. 11, no. 2, pp. 131–146, Jun. 2010.

[91] P. Gogoi, B. Borah, and D. Bhattacharyya, "Anomaly detection analysis of intrusion data using supervised & unsupervised approach," *J. Convergence Inf. Technol.*, vol. 5, no. 1, pp. 95–110, Feb. 2010.

[92] H. Sarvari and M. Keikha, "Improving the accuracy of intrusion detection systems by using the combination of machine learning approaches," in *Proc. Int. Conf. Soft Compu. Pattern Recog.*, Dec. 2010, pp. 334–337.

[93] N. Seliya and T. Khoshgoftaar, "Active learning with neural networks for intrusion detection," in *Proc. IEEE Int. Conf. Inf. Reuse Integr.*, Aug. 2010, pp. 49–54.

[94] M. Sheikhan, Z. Jadidi, and M. Beheshti, "Effects of feature reduction on the performance of attack recognition by static and dynamic neural networks," *World Appl. Sci. J.*, vol. 8, no. 3, pp. 302–308, 2010.

[95] P. Srinivasulu, J. R. Rao, and I. R. Babu, "Network intrusion detection using FP tree rules," *J. Adv. Netw. Appl.*, vol. 1, no. 1, pp. 30–39, 2009.

[96] C. Torrano-Giménez, A. Pérez-Villegas, and G. Álvarez Marañón, "An anomaly-based approach for intrusion detection in web traffic," *J. Inf. Assur. Security*, vol. 5, no. 4, pp. 446–454, 2010.

[97] J. Wang, T. Li, and R. Ren, "A real time IDSs based on artificial bee colony-support vector machine algorithm," in *Proc. 3rd Int. Workshop Adv. Comput. Intell.*, Aug. 2010, pp. 91–96.

[98] Y. Wang, D. Gu, M. Wen, J. Xu, and H. Li, "Denial of service detection with hybrid fuzzy set based feed forward neural network," in *Advances in Neural Networks—ISNN 2010*, vol. 6064. Berlin, Germany: Springer-Verlag, Jun. 2010, ser. Lecture Notes in Computer Science, pp. 576–585.

[99] J. Xiao-Pei and W. Hou-Xiang, "A new immunity intrusion detection model based on genetic algorithm and vaccine mechanism," *Int. J. Comput. Netw. Inf. Security*, vol. 2, no. 2, pp. 33–39, Dec. 2010.

[100] X. Ye, J. Li, and Y. Li, "An anomaly detection system based on hide Markov model for MANET," in *Proc. 6th Int. Conf. Wireless Commun. Netw. Mobile Comput.*, Sep. 2010, pp. 1–4.

[101] B. Zeng, L. Yao, and Z. Chen, "A network intrusion detection system with the snooping agents," in *Proc. Int. Conf. Comput. Appl. Syst. Modeling*, Oct. 2010, vol. 3, pp. V3-232–V3-236.

[102] F. Zhu, J. Long, W. Zhao, and Z. Cai, "A misleading attack against semi-supervised learning for intrusion detection," in *Modeling Decisions for Artificial Intelligence*, vol. 6408, V. Torra, Y. Narukawa, and M. Daumas, Eds. Berlin, Germany: Springer-Verlag, 2010, ser. Lecture Notes in Computer Science, pp. 287–298.

[103] I. Ahmad, A. B. Abdullah, and A. S. Alghamdi, "Application of artificial neural network in detection of DOS attacks," in *Proc. 2nd Int. Conf. Security Inf. Netw.*, Oct. 2009, pp. 229–234.

[104] W. Al-Sharafat and R. Naoum, "Significant of features selection for detecting network intrusions," in *Proc. Int. Conf. Internet Technol. Secured Trans.*, Nov. 2009, pp. 1–6.

[105] F. Barika, N. El Kadhi, and K. Ghedira, "MA IDS: Mobile agents for intrusion detection system," in *Proc. IEEE Int. Adv. Comput. Conf.*, Mar. 2009, pp. 900–910.

[106] F. A. Barika, N. E. Kadhi, and K. Ghédira, "Agent IDS based on misuse approach," *J. Softw.*, vol. 4, no. 6, pp. 495–507, Aug. 2009.

[107] Y. Changguo *et al.*, "Improvement of association rules mining algorithm in wireless network intrusion detection," in *Proc. Int. Conf. Comput. Intell. Natural Comput.*, Jun. 2009, vol. 2, pp. 413–416.

[108] D. Md. Farid, J. Darmont, N. Harbi, H. H. Nguyen, and M. Z. Rahman, "Adaptive network intrusion detection learning: Attribute selection and classification," in *Proc. Int. Conf. Comput. Syst. Eng.*, Bangkok, Thailand, Jul. 2009, pp. 154–158.

[109] J. Gao, W. Hu, X. Zhang, and X. Li, "Adaptive distributed intrusion detection using parametric model," in *Proc. IEEE/WIC/ACM Int. Joint Conf. Web Intell. Intell. Agent Technol.*, Sep. 2009, vol. 1, pp. 675–678.

[110] A. Orfila, J. Estevez-Tapiador, and A. Ribagorda, "Evolving high-speed, easy-to-understand network intrusion detection rules with genetic programming," in *Applications of Evolutionary Computing*, vol. 5484. Berlin, Germany: Springer-Verlag, 2009, ser. Lecture Notes in Computer Science, pp. 93–98.

[111] M. Panda and M. R. Patra, "Ensemble of classifiers for detecting network intrusion," in *Proc. Int. Conf. Adv. Comput., Commun. Control*, Jan. 2009, pp. 510–515.

[112] B. Shanmugam and N. Idris, "Improved intrusion detection system using fuzzy logic for detecting anamoly and misuse type of attacks," in *Proc. Int. Conf. Soft Comput. Pattern Recog.*, Dec. 2009, pp. 212–217.

[113] M. Sheikhan and A. Sha'bani, "Fast neural intrusion detection system based on hidden weight optimization algorithm and feature selection," *World Appl. Sci. J.—(Special Issue Comput. IT)*, vol. 7, pp. 45–53, 2009.

[114] S. Singh and S. Silakari, "Generalized discriminant analysis algorithm for feature reduction in cyber attack detection system," *Int. J. Comput. Sci. Inf. Security*, vol. 6, no. 1, pp. 173–180, Oct. 2009, abs/0911.0787.

[115] J. Tian and M. Gao, "Network intrusion detection method based on high speed and precise genetic algorithm neural network," in *Proc. Int. Conf. Netw. Security, Wireless Commun. Trusted Comput.*, Apr. 2009, vol. 2, pp. 619–622.

[116] Q. Xu, Z. Bai, and L. Yang, "An improved perceptron tree learning model based intrusion detection approach," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, Nov. 2009, vol. 4, pp. 307–311.

[117] A. Zainal, M. Maarof, and S. Shamsuddin, "Ensemble classifiers for network intrusion detection system," *J. Inf. Assur. Security*, vol. 4, pp. 217–225, Jul. 2009.

[118] S. Zaman and F. Karray, "Fuzzy ESVDF approach for intrusion detection systems," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, May 2009, pp. 539–545.

[119] M. Zamani, M. Movahedi, M. Ebadzadeh, and H. Pedram, "A DDoS-aware IDS model based on danger theory and mobile agents," in *Proc. Int. Conf. Comput. Intell. Security*, Dec. 2009, vol. 1, pp. 516–520.

[120] G. Zargar and P. Kabiri, "Identification of effective network features to detect Smurf attacks," in *Proc. IEEE Student Conf. Res. Develop.*, Nov. 2009, pp. 49–52.

[121] M. Zhenying, "Reason for hierarchical self organized map-based intrusion detection system incapable of increasing detection rate," in *Proc. Int. Symp. Inf. Eng. Electron. Commerce*, May 2009, pp. 150–154.

[122] R. Fanelli, "A hybrid model for immune inspired network intrusion detection," in *Artificial Immune Systems*, vol. 5132. Berlin, Germany: Springer-Verlag, 2008, ser. Lecture Notes in Computer Science, pp. 107–118.

[123] W. Hu, W. Hu, and S. Maybank, "AdaBoost-based algorithm for network intrusion detection," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 577–583, Apr. 2008.

[124] M. Rehak, M. Pechoucek, P. Celeda, J. Novotny, and P. Minarik, "CAMNEP: Agent-based network intrusion detection system," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst., Ind. Track*, 2008, pp. 133–136.

[125] H. Zhengbing, L. Zhitang, and W. Junqi, "A novel network intrusion detection system (NIDS) based on signatures search of data mining," in *Proc. 1st Int. Workshop Knowl. Discov. Data Mining*, Jan. 2008, pp. 10–16.

David J. Weller-Fahy (S'11–M'12) received the B.S. degree in computer science from the University of Illinois at Springfield, Springfield, IL, USA, in 2010 and the M.S. degree in cyber operations from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, OH, USA, in 2013. He is a Senior Master Sergeant with the Air Force. His research interests include network intrusion detection, machine learning, artificial intelligence, network characterization, and data visualization.

Brett J. Borghetti received the B.S. degree in electrical engineering from the Worcester Polytechnic Institute, Worcester, MA, USA, in 1992; the M.S. degree in computer systems from the Air Force Institute of Technology (AFIT), Wright-Patterson Air Force Base, Dayton, OH, USA, in 1996; and the Ph.D. degree in computer science from the University of Minnesota, Twin Cities, MN, USA, in 2008. He is an Assistant Professor of computer science at AFIT. His research interests focus on anomaly detection, artificial intelligence, multi-agent systems, human–machine teams, game theory, and machine learning.

Angela A. Sodemann (M'12) received the M.S. degree in mechanical engineering from the University of Wisconsin–Milwaukee, Milwaukee, WI, USA, in 2006 and the Ph.D. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2009. She is currently an Assistant Professor of mechanical engineering with the Department of Engineering, College of Technology and Innovation, Arizona State University, Mesa, AZ, USA. Her current research interests include applications of artificial intelligence and machine learning in mechatronics, robotics, and manufacturing.