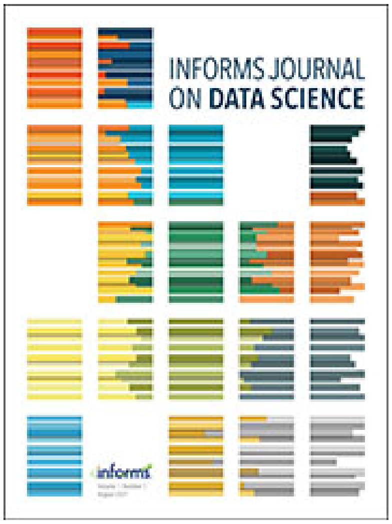


INFORMS Journal on Data Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A Nonparametric Subspace Analysis Approach with Application to Anomaly Detection Ensembles

Irad Ben-Gal, Marcelo Bacher, Morris Amara, Erez Shmueli

To cite this article:
Irad Ben-Gal, Marcelo Bacher, Morris Amara, Erez Shmueli (2023) A Nonparametric Subspace Analysis Approach with Application to Anomaly Detection Ensembles. INFORMS Journal on Data Science

Published online in Articles in Advance 13 Apr 2023

. https://doi.org/10.1287/ijds.2023.0027

informs.
https://pubsonline.informs.org/journal/ijds

INFORMS JOURNAL ON DATA SCIENCE
Articles in Advance, pp. 1–17
ISSN 2694-4022 (print), ISSN 2694-4030 (online)

A Nonparametric Subspace Analysis Approach with Application to Anomaly Detection Ensembles

Irad Ben-Gal,[a] Marcelo Bacher,[a] Morris Amara,[a] Erez Shmueli[a,*]

[a] Department of Industrial Engineering, Tel Aviv University, 69978 Tel Aviv, Israel
*Corresponding author

Contact: bengal@tauex.tau.ac.il (IB-G); marcelo.bacher@web.de (MB); morris.amara@gmail.com (MA); shmueli@tau.ac.il, https://orcid.org/0000-0003-3193-5768 (ES)

Received: February 5, 2021
Revised: June 8, 2021; August 28, 2022
Accepted: February 13, 2023
Published Online in Articles in Advance: April 13, 2023

https://doi.org/10.1287/ijds.2023.0027

Abstract. Identifying anomalies in multidimensional data sets is an important yet challenging task in many real-world applications. A special case arises when anomalies are occluded in a small subset of attributes. We propose a new subspace analysis approach, called agglomerative attribute grouping (AAG), that searches for subspaces composed of highly correlative (in the general sense) attributes. Such correlations among attributes can better reflect the behavior of normal observations and hence, can be used to improve the identification of abnormal data samples. The proposed AAG algorithm relies on a generalized multiattribute measure (derived from information theory measures over attributes' partitions) for evaluating the "information distance" among various subsets of attributes. To determine the set of subspaces, AAG applies a variation of the well-known agglomerative clustering algorithm with the proposed measure as the underlying distance function, whereas in contrast to existing methods, AAG does not require any tuning of parameters. Finally, the set of informative subspaces can be used to improve subspace-based analytical tasks, such as anomaly detection, novelty detection, forecasting, and clustering. Extensive evaluation over real-world data sets demonstrates that (i) in the vast majority of cases, AAG outperforms both classical and state-of-the-art subspace analysis methods when used in anomaly and novelty detection ensembles; (ii) it often generates fewer subspaces with fewer attributes each, thus resulting in faster training times for the anomaly and novelty detection ensemble; and (iii) the generated subspaces can also be useful in other analytical tasks, such as clustering and forecasting.

Funding: This research was partially supported by the Israeli Ministry of Economy (METRO 450 Consortium within the frame of MAGNET program) as well as by the Koret foundation grant for Smart Cities and Digital Living 2030.

Supplemental Material: The online appendix is available at https://doi.org/10.1287/ijds.2023.0027.

Keywords: subspace analysis • anomaly detection • novelty detection • Rokhlin distance

1. Introduction

Anomaly detection refers to the problem of finding patterns in data that do not conform to an expected norm behavior. These nonconforming data points or patterns are often referred to as anomalies, outliers, novelties, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants, depending on the application domain (Chandola et al. 2007). Algorithms for detecting anomalies are extensively used in a wide variety of application domains, such as machinery monitoring (Ben-Gal et al. 2003, Ge and Song 2012, Bacher et al. 2017, Kenett and Zacks 2021), sensor networks (Bajovic et al. 2011), intrusion detection in data networks (Jyothsna et al. 2011), healthcare (Tarassenko et al. 2005), and social networks (Aggarwal and Subbian 2012). A major reason for their widespread use is the fact that anomalies can often be translated directly to actionable recommendations based on either "good" or "bad" deviations from the norm (Chandola et al. 2007).

In a typical anomaly detection setting, only normal or expected observations are available, and consequently, some assumptions regarding the distribution of anomalies must be made to discriminate normal from anomalous observations (Steinwart et al. 2005). Traditional approaches for anomaly detection (see, e.g., Ben-Gal 2010, Pimentel et al. 2014) often assume that anomalies occur sporadically and are well separated from the normal data observations or that anomalies are uniformly distributed around the normal observations. However, in complex environments, such assumptions may not hold. For instance, consider the case of a complex system and a diagnosis module that continuously monitors the functionality of the system by analyzing multiattribute (we use the terms attribute, variable, and feature interchangeably) data generated from a set of sensors. If only one of the system's modules breaks down or alternatively, if only a few of the monitoring sensors fail to function normally, only some of the data attributes will be affected. Thus, from a data analysis perspective, these malfunctions can be seen as an addition of noise or deviation with respect to a *subset* of the attributes. Consequently, anomalies in the system's generated data might be noticeable only in some projections of the data into a lower-dimensional subspace and not necessarily in the entire data space, as often assumed by classical approaches. This phenomenon is also known in the literature as "sparse change." As another motivating example, consider a case where anomalies represent a new (previously unknown) class of data observations, commonly called novelties (Chandola et al. 2007). Similar to the malfunctions example, deviations from the original data observations might only be visible along a subset of attributes. However, these attributes are often correlated in some sense and therefore, cannot be treated as additive noise.

Based on these concepts, ensembles were proposed as a more effective paradigm for anomaly detection (Aggarwal and Yu 2001). Ensembles for anomaly detection typically follow three general steps (Lazarevic and Kumar 2005). First, a set of subspaces is generated—often by *randomly* selecting subsets of attributes. This step is commonly referred to as *subspace analysis*. Then, classical anomaly detection algorithms are applied on each subspace to compute local anomaly scores. Finally, these local scores are aggregated to derive a global anomaly score (e.g., using majority voting). In this work, we focus on the subspace analysis stage, which aims to find a representative set of subspaces among a very large number of possible subspace combinations, such that anomalies can be identified effectively and efficiently.

Several methods for subspace analysis have been proposed in the literature. These methods can be classified into three broad approaches. The most basic one is based on a random selection of attributes (e.g., Lazarevic and Kumar 2005). Other methods search for subspaces by giving anomality grades to data samples, thus coupling the search for meaningful subspaces with the anomaly detection algorithm (see, e.g., Müller et al. 2010, Ha et al. 2015). Recent methods search for subspaces composed of highly correlative attributes (e.g., Nguyen et al. 2014). These methods rely on the assumption that, in such subspaces, the correlations among attributes represent a systematic interaction among the attributes that can better reflect the behavior of normal observations and hence, can be used to better identify those deviating abnormal cases. However, all of the methods suffer from one or more of the following limitations. (i) Relevant attributes might not be included in the generated set of subspaces. This might impact the effectiveness of the ensemble because anomalies might occur anywhere in the data space. (ii) The set of generated subspaces might contain thousands and even millions of subspaces, which may make the training and operation phases of the ensemble computationally prohibitive. (iii) These approaches often require us, prior to their execution, to set the values of parameters, such as the number of subspaces, the maximal size of each subspace, or the number of clusters—parameters that are typically hard to predefine or tune at such a stage. For a more comprehensive review of existing subspace analysis methods, the reader is referred to Online Appendix 1.

To address the challenges mentioned, we propose the agglomerative attribute grouping method (AAG) for subspace analysis. Motivated by previous works, AAG searches for subspaces that are composed of highly correlative attributes. As a general measure for attribute association, AAG applies an information theory measure over attributes' partitions (see, e.g., Simovici 2007, Kagan and Ben-Gal 2014). In particular, AAG introduces a generalized Rokhlin distance (Rokhlin 1967) as a multiattribute measure to find subspaces with small distances (i.e., distances that reflect high information content among the attributes in those subsets). Finally, AAG applies a variation of the well-known agglomerative clustering algorithm, where subspaces are greedily searched by minimizing the multiattribute measure. AAG also contains a pruning mechanism that aims at improving the convergence time of the algorithm while limiting the size of the generated subspaces.

Several important characteristics differentiate AAG from existing state-of-the-art approaches. First, because of the used agglomerative scheme in the subspace search, none of the data attributes are discarded, and attributes are combined in an effective manner to generate the set of subspaces. Second, the set of subspaces that AAG generates is relatively "compact" in comparison with existing methods for two main reasons; the use of the agglomerative approach results in a relatively small number of subspaces, and the pruning mechanism results in a relatively small number of attributes in each subspace. Finally, as a result of combining the agglomerative approach with the minimization of the suggested measure, AAG does not require any tuning of parameters.

To evaluate the proposed AAG method, we conducted extensive experiments on 25 publicly available data sets while using eight different classical and state-of-the-art subspace analysis methods as benchmarks. The evaluation results show that an AAG-based ensemble for anomaly detection (i) outperforms the benchmark methods in cases where anomalies occur in relatively small subsets of the available attributes as well as in cases where anomalies represent a new class (i.e., novelties) and (ii) often generates fewer subspaces with a smaller (on average) number of attributes in comparison with the benchmark approaches, thus resulting in a faster training time for the anomaly detection ensemble. We also demonstrate how these subspaces can be used for forecasting based on the exogenous variables in the subsets and evaluate this setting using a real world retail data set.

It is important to note that, whereas subspace analysis for anomaly detection seems to be similar to attribute selection for supervised classification (Guyon et al. 2008) as well as to some ensemble-based classification methods (e.g., random forest in Breiman 2001), they differ greatly. The main difference between the two approaches stems from the type of data available in the training phase of the classification task versus the available data for the anomaly detection task. In the supervised classification task, information about each of the classes is usually available, whereas in anomaly detection tasks, information about abnormal data samples is often missing, and only information about the normal observations is provided. Moreover, although the goal in the case of attribute selection is to discard redundant attributes to improve accuracy and run time of the classifier, it is usually impossible to discard attributes at the training stage of anomaly detection tasks because they might be found to be extremely informative in the operational stage.

The contribution of this paper is twofold. First, it introduces a new multiattribute information-theoretic measure, which can be seen as an extension to the Rokhlin metric. The proposed measure enables us to compute the expected information gain of potential subspaces, with the aim of identifying unexpected observations. The new measure has several appealing properties. (i) Unlike many other measures, such as Pearson correlation, it can be computed over a set of more than two variables. (ii) Unlike other measures that can handle numerical attributes only, it can handle numerical as well as categorical variables. (iii) It enables us to expose high-order nonlinear dependencies among attributes, whereas simpler correlation measures often reveal linear dependencies among the variables. To the best of our knowledge, this paper is the first one to apply the multiattribute extension of the Rokhlin distance in the context of subspace analysis.

Second, this paper introduces the AAG method, which is a novel algorithm for subspace analysis. The proposed AAG algorithm is unique in the sense that (i) it is nonparametric, (ii) it outperforms other methods when used in anomaly and novelty detection ensembles, and (iii) it often generates more "compact" subspaces.

This work extends two earlier conference papers (Bacher et al. 2016, 2017) by (i) expanding the selection mechanism of AAG to support a stability index (*SI*) for the selected subspaces; (ii) outlining properties of the proposed multiattribute measure and proving them (e.g., Lemma 2); (iii) providing an extensive evaluation of the proposed approach, which now includes additional settings, data sets, and benchmarks, including an analysis of a real-world forecasting use case; and (iv) elaborating on the statistical analyses of the obtained results.

The rest of the paper is organized as follows. Section 2 proposes a novel measure (based on concepts of information theory over sets of partitions) that enables to evaluate the smallest "distance" among subspaces of attributes. Section 3 describes the proposed AAG approach. Section 4 presents an experimental evaluation of AAG and the obtained results. Finally, Section 5 summarizes this paper and discusses some future research directions.

2. Information Theory Measures for Partitions

This section discusses how to apply information-theoretic measures over partitions of a generic data set in order to compute the distances among various subsets of attributes. In particular, we review the Rokhlin distance (Rokhlin 1967) and its application for attributes' association following a partitioning of a data set. We then suggest an extension of the Rokhlin distance to a multiattribute measure for any number of attributes. To that end and to maintain a self-contained text, we start this section by providing a brief review of concepts of partitions and their implementation to information theory while presenting the notation that is used throughout the paper.

2.1. Preliminaries

In this subsection, we follow Kagan and Ben-Gal (2014) and present some definitions of information-theoretic measures between partitions of a finite data set. Let D be a finite sample space composed of N observations and p attributes ($\{A_1, A_2, \dots, A_p\}$), and let χ be a set of partitions of the sample space D as defined next. Each partition $\alpha_i = \{\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iK}\}$, $K \le N$, $\alpha_{ij} \cap \alpha_{im} = \emptyset$, $\forall j$, $m = 1, 2, \dots, K$, $j \neq m$ is defined by the values of its corresponding attribute A_i, where $\alpha_{i1}, \alpha_{i2}, \dots, \alpha_{iK}$ are the sets of indices of identical values of the attribute A_i. For example, let attribute A_i contain the values $\{a_{i1}, a_{i2}, \dots, a_{iN}\}$, such that $a_{i1} = a_{i2} = a_{i3}$ and $a_{ij} \neq a_{ik}$, $\forall j, k = 3, 4, \dots, N$, $j \neq k$. Then, the partition of D generated by the attribute A_i is $\{\{a_{i1}, a_{i2}, a_{i3}\}, \{a_{i4}\}, \dots, \{a_{iN}\}\}$, which in terms of indices, is represented by $\alpha_i = \{\alpha_{i1} = \{1, 2, 3\}, \alpha_{i2} = \{4\}, \dots, \alpha_{iN-2} = \{N\}\}$. Note that, by definition, the union of the partition elements is the set of all indices (i.e., $\cup_{j=1}^{K} \alpha_{ij} = \{1, 2, \dots, N\}$, $\forall i$).

To define the entropy and the informational measures between partitions rather than with the conventional approach that defines them between random variables, it is necessary to specify a probability distribution associated with a partition. For finite sets, the empirical probability distribution induced by a partition $\alpha_i \in \chi$ is defined as follows (Simovici 2007):

$$p_{\alpha_i}(\alpha_{ij}) = \frac{|\alpha_{ij}|}{N},$$

where $|\cdot|$ represents the cardinality of the set. Note that by definition, $\sum_{j=1}^{K}(|\alpha_{ij}|/N) = 1$. Thus, the partition α_i induces a random variable $X_i \in \{\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iK}\}$, with the probabilities $p_{\alpha_i}(\alpha_{ij})$ defined over the partition elements $\alpha_{ij}, j = 1, 2, \ldots, K$.

The Shannon entropy of the random variable X_i of the corresponding partition α_i is then defined as $H(X_i) = -\sum_{\alpha_{ij}\in\alpha_i} p_{\alpha_i}(\alpha_{ij})\log_2[p_{\alpha_i}(\alpha_{ij})]$ (where by the usual convention, $0\ \log_2 0 = 0$) and is denoted for simplicity by $H(\alpha_i)$. Notice that the probabilities used in computing the entropy are obtained from the relative frequencies of the unique values of the attribute A_i regardless their order. Therefore, one can represent the Shannon entropy associated with attribute A_i, denoted by $H(\alpha_i)$, by the entropy over the distribution of the partition elements in α_i.

Let α_i and α_j be two partitions corresponding to the two attributes A_i and A_j, respectively. A new partition can be generated based the intersection between the two attributes' partitions. For example, consider attributes A_i and A_j, where attribute A_i contains the values $\{1,1,1,2,2,2\}$ and attribute A_j contains the values $\{1,1,2,2,2,3\}$. That is, $\alpha_i = \{\alpha_{i1} = \{1,2,3\}, \alpha_{i2} = \{4,5,6\}\}$ and $\alpha_j = \{\alpha_{j1} = \{1,2\}, \alpha_{j2} = \{3,4,5\}, \alpha_{j3} = \{6\}\}$. Then, the partition of D generated by the two attributes A_i and A_j is the subsets of indices of identical values of the two attributes together. In this example, the following partition is obtained: $\alpha = \{\alpha_{i1} \cap \alpha_{j1} = \{1,2\}, \{\alpha_{i1} \cap \alpha_{j2} = \{3\}, \{\alpha_{i2} \cap \alpha_{j2} = \{4,5\}, \{\alpha_{i2} \cap \alpha_{j3} = \{6\}\}$, where $\alpha_{im} \cap \alpha_{jk}$ is the intersection of the two subsets (thus, the subset of all elements that are members of both the subsets (intersections resulting in empty sets were omitted)). Such a partition, which is generated by two or more attributes, is often called a *refinement* of the partition generated by each of the individual attributes alone. It defines the joint probability distribution $p(\alpha_i, \alpha_j)$ associated with the intersection between attribute A_i and A_j.

Considering the empirical probability distributions induced by the partitions (Simovici 2007, Kagan and Ben-Gal 2013), the conditional entropy of the partition α_i with respect to the partition α_j is defined as follows:

$$H(\alpha_i|\alpha_j) = -\sum_{\alpha_{jk}\in\alpha_j}\sum_{\alpha_{im}\in\alpha_i} p(\alpha_{im}, \alpha_{jk})\log_2[p(\alpha_{im}|\alpha_{jk})],$$

where $p(\alpha_{im}, \alpha_{jk}) = p(\alpha_{im} \cap \alpha_{jk})$ and $p(\alpha_{im}|\alpha_{jk}) = p(\alpha_{im} \cap \alpha_{jk})/p(\alpha_{jk})$. In the example, $\alpha_{i1} \cap \alpha_{j1} = \{1,2\}; p(\alpha_{i1}, \alpha_{j1}) = 2/6$, whereas $p(\alpha_{i1}|\alpha_{j1}) = 2/6 : 2/6 = 1$; and $p(\alpha_{i2}, \alpha_{j2}) = 2/6$, whereas $p(\alpha_{i1}|\alpha_{j1}) = 2/6 : 3/6 = 2/3$. Similarly, $p(\alpha_{im}|\alpha_{jk})$ is the conditional probability distribution defined over the partition elements in the intersection subset, given the conditioning subset as shown.

The Rokhlin distance between two partitions α_i and α_j is defined as the sum of conditional entropies of these partitions (Rokhlin 1967): that is,

$$d_R(\alpha_i, \alpha_j) = H(\alpha_i|\alpha_j) + H(\alpha_j|\alpha_i). \tag{1}$$

For detailed consideration of the metric properties of this distance, see Sinai et al. (1976). Recall that the partitions α_i and α_j are associated with attributes A_i and A_j, respectively. The scheme presented allows us to compute the Shannon entropy of a partition α_i by using the relative frequencies of the values of attribute A_i. Similarly, the conditional entropy $H(A_i|A_j)$ can be considered as the conditional entropy of partition α_i given partition α_j as seen. It follows that the Rokhlin distance can be computed equivalently for the attributes and their corresponding partitions:

$$\begin{aligned} d_R(A_i, A_j) &= H(A_i|A_j) + H(A_j|A_i) = H(\alpha_i|\alpha_j) + H(\alpha_j|\alpha_i) \\ &= d_R(\alpha_i, \alpha_j). \end{aligned} \tag{2}$$

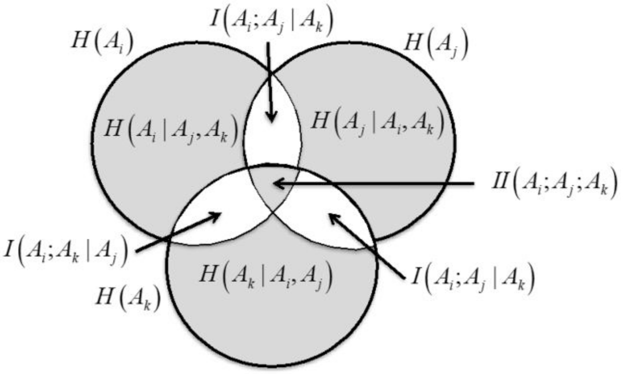
Note that the Rokhlin distance is directly related to Shannon's mutual information as a measure of entropy reduction. Recall that $I(A_i; A_j) = H(A_i) - H(A_i|A_j)$ and $H(A_i) = H(A_i, A_j) - H(A_j|A_i)$ (Cover and Thomas 2006), where $H(A_i, A_j)$ corresponds to the joint entropy of the two attributes derived by their joint probability distribution $p(\alpha_i, \alpha_j)$. Thus, $d_R(A_i, A_j) = H(A_i, A_j) - I(A_i; A_j)$.

Accordingly, the Rokhlin distance can be interpreted as a measure of mutual dependence between two attributes. A small Rokhlin distance reflects a small conditional entropy value and a high mutual information value between the attributes. A direct implementation of the Rokhlin distance as a formal informational metric between partitions has practical implications. For example, it was used in Kagan and Ben-Gal (2013) for constructing a search algorithm and in Kagan and Ben-Gal (2014) for creating various testing trees. For an illustrative example of d_R, the reader is referred to Online Appendix 2.1.

2.2. Multiattribute Measure, d_{MA}

The informational distance measures between two attributes are given in Equation (2). Subsequently, we now extend this concept to derive a similar notion of informational distance between a set of attributes while relying on the partitions associated with these attributes. Note that a partition can be generated by more than two attributes by following the same concepts presented. Once a partition is established, it can be treated following the steps; thus, an empirical distribution can be generated to define its entropy as well as its conditional entropy given another partition (that could be generated by another set of attributes). The new *multiattribute measure*, denoted by d_{MA}, is induced by sets of partitions somewhat similar to the symmetric difference between sets. The *symmetric difference* between two sets, which is also known as the *disjunctive union*, is defined as follows: $A_i \Delta A_j = (A_i \backslash A_j) \cup (A_j \backslash A_i)$. It considers the set of elements that are in either of the sets A_i and A_j but not in their intersection (see, e.g., Kuratowski 2014). Among

Figure 1. The Symmetric Difference of Three Nonempty Sets is Represented by the Gray Areas Together with the Information-Theoretic Relationships Among the Corresponding Attributes A_i, A_j, and A_k



several properties of this measure, it is known that the symmetric difference is commutative and associative. That is, let A_i, A_j, and A_k be three different sets; then, $A_i \Delta A_j \Delta A_k = (A_i \Delta A_j) \Delta A_k = A_i \Delta (A_j \Delta A_k)$. Correspondingly, the Hamming distance between sets A_i and A_j is defined as the cardinality of the set $A_i \Delta A_j$, denoted by $|A_i \Delta A_j|$ (see, e.g., Simovici 2007).

Following a similar analysis, the symmetric difference of three attributes A_i, A_j, and A_k (and their corresponding partitions α_i, β_j, and λ_k, respectively) is presented by the Venn diagram in Figure 1, where the gray areas represent the union of the attributes without their successive intersections.

Namely, $H(A_i)$, $H(A_j)$, and $H(A_k)$ denote, respectively, the Shannon entropies of the attributes A_i, A_j, and A_k, whereas $H(A_i|A_j)$, for example, denotes the conditional entropy of A_i given A_j. $I(A_i;A_j)$ is the mutual information between attributes A_i and A_j, and $I(A_i;A_j|A_k) = H(A_i|A_k) - H(A_i|A_j,A_k)$ is the conditional mutual information between attributes A_i and A_j, given attribute A_k (see Cover and Thomas 2006). Finally, $II(A_i;A_j;A_k)$ denotes the *multivariate mutual information* among the three attributes that was introduced in the seminal work of McGill (1954) as a measure of the higher-order interaction among random variables, where $II(A_i;A_j;A_k) = I(A_i;A_j) - I(A_i;A_j|A_k)$. It can be shown that the multivariate mutual information is bounded from above by $II(A_i;A_j;A_k) \le \min\{I(A_i;A_j|A_k), I(A_i;A_k|A_j), I(A_j;A_k|A_i)\}$ (McGill 1954).

We can now define the measure d_{MA} involving three attributes as follows:

$$d_{MA}(A_i,A_j,A_k) = H(A_i|A_j,A_k) + H(A_j|A_i,A_k) + H(A_k|A_i,A_j) + II(A_i;A_j;A_k), \quad (3)$$

where the first three terms on the right side of Equation (3) represent the degree of uncertainty among attributes and the last term represents the shared information among them. As seen in Figure 1, the multiattribute measure, d_{MA}, over three attributes measures how distant these attributes are from each other in a similar manner to the Rokhlin distance d_R, which is defined in Equation (2) over two attributes.

The generalized d_{MA} can be applied to a higher number of attributes. In particular, the extension of Equation (3) to p attributes is somewhat similar to the symmetric difference for p sets (see, e.g., Kuratowski 2014), and each set represents the partition of one or more attributes:

$$d_{MA}(\boldsymbol{A}) = \sum_{i=1}^{p} H(A_i|\boldsymbol{A}\backslash A_i) + II(\boldsymbol{A}), \quad (4)$$

where $\boldsymbol{A} = \{A_1, A_2, \ldots, A_p\}$ denotes a multiset of attributes in D, $A_i \in \boldsymbol{A}$. Note that $\boldsymbol{A}$ can also represent the symmetric difference of two or more sets of attributes (e.g., for $\boldsymbol{A} = \boldsymbol{A}_1 \Delta \boldsymbol{A}_2$, $d_{MA}(\boldsymbol{A}) = d_{MA}(\boldsymbol{A}_1 \Delta \boldsymbol{A}_2) = d_{MA}(\boldsymbol{A}_1, \boldsymbol{A}_2)$), where the (conditional) entropy of such union of attributes is defined by the partition associated with this multiset as seen. Recall that the term $II(\boldsymbol{A})$ is the multivariate mutual information defined for $p > 2$ in McGill (1954). In Jakulin (2005), the multivariate mutual information was extended as the recursive computation $II(A_1, A_2, \ldots, A_p) = II(A_1, A_2, \ldots, A_{p-1}) - II(A_1, A_2, \ldots, A_{p-1}|A_p)$. The latter definition reflects that the multivariate mutual information is the intersection of all partitions produced by the p attributes. This explains why for large p, the intersection of all partitions often results in a fully refined partition. Note that in the case of $p = 2$, the term $II(\cdot)$ is defined as zero. Thus, Equation (4) reduces to the Rokhlin distance between two partitions. For an illustrative example of d_{MA}, the reader is referred to Online Appendix 2.2.

There are several benefits of using the proposed measure to analyze subspaces as detailed next. First, minimizing the proposed multiattribute measure corresponds to the selection of informative subspaces that are composed of highly correlated attributes. Thus, providing an interpretable and explainable outcome.

Second, unlike classical and state-of-the-art approaches (such as ENCLUS (Cheng et al. 1999), 4S (Nguyen et al. 2014), and CMI (Nguyen et al. 2013)), the proposed subspace search algorithm minimizes the d_{MA} rather than maximizing other information measures, such as the total correlation (TC) (Watanabe 1960). This proposed procedure does not require us to select a priori some parameters (e.g., an information threshold parameter) and is shown to yield better empirical results over various data sets, as seen in later sections.

Third, the minimization of the proposed multiattribute measure tends to delegate the combination of attributes with low information content (for example, attributes with large numbers of uniformly distributed symbols) to later stages of the search, where their effects on the information measure over all the subsets structure are less critical. Consequently, the first summation term in Equation (4) approaches the sum of the Shannon entropy of the individual attributes, which by definition, yields a higher value than that of the conditional entropy (Cover and Thomas 2006). Thus, using the proposed method results in adding more informative attributes to the generated subspaces.

Finally, later sections empirically show that minimization of the proposed measure tends to generate, on average, a smaller set of subspaces than other approaches, especially in the case of data sets whose attributes have a considerably high number of unique values. A direct consequence of this characteristic is a reduced training time, on average, of the ensemble models.

2.3. Properties of d_{MA}

In this section, we describe two properties of the multiattribute measure, d_{MA}, that are used by the proposed AAG algorithm for a search after informative subsets. The first property describes an approximation of d_{MA} when the number of attributes is high, whereas the second property indicates that for a high number of independent attributes, d_{MA} can be used as a pseudometric to find informative subsets.

As the number of attributes p grows, the probability distributions induced by their partitions are becoming higher dimensional, and hence, the estimation of the multiattribute measure d_{MA} becomes more computationally demanding. To address this challenge, we make use of the following claim on d_{MA} given two sets of attributes $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$, where the latter is a subset of the first.

Lemma 1. $\boldsymbol{A}_j \subseteq \boldsymbol{A}_i \Rightarrow d_{MA}(\boldsymbol{A}_j) \geq d_{MA}(\boldsymbol{A}_i)$.

Proof. Refer to Online Appendix 3.1. □

An immediate result of Lemma 1 is the following approximation scheme. Namely, given a set of subsets of $\boldsymbol{A}$ denoted by $\tilde{\boldsymbol{A}}$, then $d_{MA}(\boldsymbol{A}) \leq \min_{\boldsymbol{A}_j \in \tilde{\boldsymbol{A}}} \{d_{MA}(\boldsymbol{A}_j)\}$. Thus, $\min_{\boldsymbol{A}_j \in \tilde{\boldsymbol{A}}} \{d_{MA}(\boldsymbol{A}_j)\}$ can be used as an upper-bound approximation of $d_{MA}(\boldsymbol{A})$, and this bound gets tighter as the subset's cardinality increases and approaches the cardinality of the entire set, as demonstrated in Online Appendix 4.2. In Section 4, an approximation of d_{MA} is applied to evaluate the information "distance" between candidate subspaces in a search for highly correlative subspaces. In our experiments, we found that calculating d_{MA} over sets with a high number of attributes can result in intractable computations. Therefore, we used an approximation based on subsets of three attributes that empirically led to informative subspaces and relatively good run time. Refer to Online Appendix 4.2 for further numerical analysis of the proposed approximation of d_{MA}.

Lemma 2. *The multiattribute measure d_{MA} is a pseudometric when the input set contains a high number of independent attributes.*

Proof. Refer to Online Appendix 3.2. □

3. Agglomerative Attribute Grouping

In this section, we present the proposed subspace analysis method, which is named the AAG. Similar to the subspace analysis methods described in Online Appendices 1.2 and 1.3, AAG generates a set of subspaces with highly correlated attributes by applying a variation of the well-known agglomerative clustering algorithm and using the proposed d_{MA} measure as the underlying distance function. The combination of this measure and the agglomerative strategy can be used to find subspace combinations without setting any parameter value (such as the number of subspaces) in advance. This is one of the differences in comparison with other conventional methods (e.g., ENCLUS (Cheng et al. 1999), FB (Lazarevic and Kumar 2005), HiCS (Keller et al. 2012), CMI (Nguyen et al. 2013), and 4S (Nguyen et al. 2014)).

The pseudocode of the proposed AAG method is shown in Algorithm 1, and a running illustrative example of it is provided in Online Appendix 2.3. The algorithm receives as input a data set D composed of N observations and p attributes. The algorithm returns as output a set of subspaces with highly correlated attributes denoted by T. The algorithm begins by initializing the result set of subspaces T to be the empty set (line 1). Then, in line 2, the algorithm generates a set of n subspaces, each of which is composed of a single attribute. This set constitutes the first agglomeration level and is denoted by $S^{(t)}$, $t = 1$ (lines 2 and 3). Then, the algorithm iteratively generates the subspaces of agglomeration level $t+1$ denoted by $S^{(t+1)}$ by combining subspaces from the previous agglomeration level, $S^{(t)}$ (lines 4–27). Each such iteration begins with updating the result set T to also contain the subspaces from the previous agglomeration level (line 5). Then, in line 6, we initialize the set of subspaces of the next agglomeration level to be the empty set. Next, in line 7, we maintain a copy of the previous agglomeration level, denoted by $S_0^{(t)}$. This is required to allow attributes to appear in different subspaces. Notice that $S_0^{(t)}$, $S^{(t)}$, and $S^{(t+1)}$, as well as T, contain the indices of the data attributes in the subspaces, whereas for example, $\boldsymbol{A}_i$ denotes the projection of data samples.

The algorithm continues by searching for two subsets in the current agglomeration level that have the lowest d_{MA} value (line 8) and adds the unified set to the next agglomeration level instead of the two individual subsets (lines 9). In lines 10–12 (and also, later in lines 18–20), the algorithm can choose not to add the resulting set; we refer to this stage as the pruning stage and describe it in detail in Section 3.1. In lines 13–25, the algorithm continues to combine subspaces iteratively until there are no more subsets left in $S^{(t)}$. However, now, the algorithm checks whether it is better to unify a subset from $S^{(t)}$ and a subset from $S^{(t+1)}$, denoted by $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$, or two subsets from $S^{(t)}$.[1] The motivation behind this stage is to avoid merging only a single pair of subspaces in each agglomeration level and to allow the merging of multiple subspaces. In doing so, we avoid the permanent selection of subspaces with a higher number of attributes to be combined.

Once all subspaces have been assigned at an agglomeration level t, the algorithm proceeds with subsequent levels of agglomeration (lines 4–27) until no subspace combination is further required (line 4). The AAG algorithm ends by returning the set of subspaces T in line 28.

The normalized multiattribute measure, denoted by $\tilde{d}(\cdot)$ as used in lines 8 and 14–16, is defined as

$$\tilde{d}(\boldsymbol{A}_i, \boldsymbol{A}_j) = \frac{d_{MA}(\boldsymbol{A}_i, \boldsymbol{A}_j)}{H(\boldsymbol{A}_i \cup \boldsymbol{A}_j)}, \tag{5}$$

where $d_{MA}(\boldsymbol{A}_i, \boldsymbol{A}_j)$ was defined in (4) and $H(\boldsymbol{A}_i \cup \boldsymbol{A}_j)$ denotes the join entropy after unifying the subspaces $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$. The normalization factor, $H(\boldsymbol{A}_i \cup \boldsymbol{A}_j)$, allows a comparison between subspaces with different numbers of attributes. We used the results from Yianilos (2002) that showed that this normalization factor does not change the measure characteristics of (4). In the general case, the computation of the measure is obtained based on Lemma 1, where we select a fixed size number of attributes (e.g., three or four) and calculate the minimum value over all subsets of this given size.

The run-time complexity of AAG is given in Lemma 3.

Lemma 3. *The run-time complexity of AAG is* $O(Np^3 \log p)$, *where* N *is the number of instances and* p *is the number of attributes.*

Proof. Refer to Online Appendix 3.3. □

Algorithm 1 (Agglomerative Attribute Grouping)

```
Input: A data set $D$ with $N$ observations and $p$ attributes
Output: A set of subspaces $T$
1.  $T \leftarrow \emptyset$
2.  $S^{(1)} \leftarrow \{\{A_1\}, \{A_2\}, \ldots, \{A_p\}\}$
3.  $t \leftarrow 1$
4.  while $(S^{(t)} \neq \emptyset)$ do
5.      $T \leftarrow T \cup S^{(t)}$
6.      $S^{(t+1)} \leftarrow \emptyset$
7.      $S_0^{(t)} \leftarrow S^{(t)}$
8.      $\{\mathbf{A}_i, \mathbf{A}_j\} = \arg\min_{\mathbf{A}_i, \mathbf{A}_j \in S^{(t)}} \tilde{d}(\mathbf{A}_i, \mathbf{A}_j)$
9.      $S^{(t)} \leftarrow S^{(t)} \setminus \{\mathbf{A}_i, \mathbf{A}_j\}$
10.     if $t \leq 2$ OR $(TC(\boldsymbol{A}_i \cup \boldsymbol{A}_j) \geq \nu_i TC(\boldsymbol{A}_i) + \nu_j TC(\boldsymbol{A}_j))$ then
11.         $S^{(t+1)} \leftarrow S^{(t+1)} \cup \{\mathbf{A}_i \cup \mathbf{A}_j\}$
12.     end if
13.     while $S^{(t)} \neq \emptyset$ do
14.         $\{\mathbf{A}_i, \mathbf{A}_j\} = \arg\min_{\mathbf{A}_i \in S^{(t)}, \mathbf{A}_j \in S^{(t+1)}} \tilde{d}(\mathbf{A}_i, \mathbf{A}_j)$
15.         $S_k = \arg\min_{\mathbf{A}_k \in S_0^{(t)} \setminus \mathbf{A}_i} \tilde{d}(\mathbf{A}_k, \mathbf{A}_i)$
16.         if $(\tilde{d}(\mathbf{A}_i, \mathbf{A}_k) \leq \tilde{d}(\mathbf{A}_i, \mathbf{A}_j))$ then
17.             $S^{(t)} \leftarrow S^{(t)} \setminus \{\mathbf{A}_i, \mathbf{A}_k\}$
18.             if $t \leq 2$ OR $(TC(A_i \cup A_j) \geq \nu_i TC(A_i) + \nu_j TC(A_j))$ then
19.                 $S^{(t+1)} \leftarrow S^{(t+1)} \cup \{\mathbf{A}_i \cup \mathbf{A}_k\}$
20.             end if
21.         else
22.             $S^{(t)} \leftarrow S^{(t)} \setminus \mathbf{A}_i$
23.             $S_j \leftarrow \{\mathbf{A}_i \cup \mathbf{A}_j\}$
24.         end if
25.     end while
26.     $t \leftarrow t + 1$
27. end while
28. return $T$
```

3.1. Pruning Stage

The agglomerative approach used in the previous section has an inherent property that the number of attributes in subspaces grows with the agglomeration level. This property has two major limitations. (i) It may have a great impact on the efficiency of the anomaly detection ensemble, and (ii) recall that the Lemma 1 approximation to Equation (4) becomes less accurate when the number of attributes grows considerably. To overcome these limitations, we propose a simple rule to determine whether to proceed with unifying two subspaces or not. This rule is embedded in the AAG algorithm in lines 10–12 and 18–20. According to this rule, two candidate subspaces are unified only if their union does not considerably reduce the subspace's quality with respect to the two individual subspace candidates. More specifically, we evaluate the TC (Watanabe 1960) of the two individual subspaces $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ and compare their sum with the TC of their union $\boldsymbol{A}_i \cup \boldsymbol{A}_j$:

$$TC(\boldsymbol{A}_i \cup \boldsymbol{A}_j) \geq \nu_i TC(\boldsymbol{A}_i) + \nu_j TC(\boldsymbol{A}_j), \tag{6}$$

where $\nu_i = J(\boldsymbol{A}_i; \boldsymbol{A}_i \cup \boldsymbol{A}_j)$ and $\nu_j = J(\boldsymbol{A}_j; \boldsymbol{A}_i \cup \boldsymbol{A}_j)$ serve as soft thresholds and $J(\cdot)$ is the well-known Jaccard index. If the condition is satisfied (the sum of individual TCs is lower than the TC of their union), the two subspaces are combined. Note that the proposed rule does not require any tuning of parameters. Moreover, its usage by AAG does not lead to discarded attributes because all attributes are already combined in the previous level of agglomeration. As noted, this is an important property in anomaly (and novelty) detection applications where all attributes are required.

In some special cases, it is possible to speed up the evaluation of the rule by avoiding the computation of the different TCs. For example, if $\boldsymbol{A}_i \cap \boldsymbol{A}_j = \emptyset$, the following lemma indicates that it is legitimate to unify the two subspaces.

Lemma 4. *Given two subspaces* $\boldsymbol{A}_i$ *and* $\boldsymbol{A}_j$ *such that* $|\boldsymbol{A}_i| \geq 2$, $|\boldsymbol{A}_j| \geq 2$, *and* $\boldsymbol{A}_i \cap \boldsymbol{A}_j = \emptyset$, *then necessarily,* $TC(\boldsymbol{A}_i \cup \boldsymbol{A}_j) \geq TC(\boldsymbol{A}_i) + TC(\boldsymbol{A}_j)$.

Proof. Refer to Online Appendix 3.4. □

Note that, for $\boldsymbol{A}_i \cap \boldsymbol{A}_j = \emptyset$, the soft thresholds result in $\nu_i \leq \delta$ and $\nu_j \leq 1 - \delta$, where $\delta \in (0, 0.5)$ is to be computed. Furthermore, it can be shown that, if $\boldsymbol{A}_i \subseteq \boldsymbol{A}_j$ (or $\boldsymbol{A}_j \subseteq \boldsymbol{A}_i$), then $TC(\boldsymbol{A}_i \cup \boldsymbol{A}_j) = TC(\boldsymbol{A}_j) \leq \nu_i TC(\boldsymbol{A}_i) + TC(\boldsymbol{A}_j)$ (because $\nu_j = 1$ and $\nu_i > 0$). Therefore, in such cases, the two subspaces should not be unified.

Also, note that although TC is not a formal metric, it can still be used for comparison (i.e., testing whether one set is "better" than the other), as implemented in Equation (6).

4. Evaluation

In this section, we compare the quality of the subspaces generated by AAG against eight other benchmark algorithms when used in ensembles for anomaly and novelty detection. Additionally, we include a demonstration of how subspaces generated by AAG can improve forecasting accuracy and clustering inner information in Online Appendix 4.5.

4.1. Experimental Settings

Our empirical study is based on the experimental settings used in Cheng et al. (1999), Keller et al. (2012), and Nguyen et al. (2013). All of our experiments were conducted on 25 real-world data sets (see Table 4 in Online Appendix 4.1) taken from the UCI repository (Bache and Lichman 2013). Although these data sets are usually used in the context of classification tasks, previous studies (Aggarwal and Yu 2001; Lazarevic and Kumar 2005; Keller et al. 2012; Nguyen et al. 2013, 2014) have also used them in the context of anomaly and novelty detection. In Section 4.1.1, we describe in detail how normal and abnormal observations for each data set were generated and how the training and test sets were obtained. In general, we considered four different settings. We focus on two of them; one is related to anomaly detection, and the second is related to novelty detection. A third setting, which is also related to anomaly detection, is described in Online Appendix 4.3. A fourth setting, which deals with forecasting and clustering, is presented in Online Appendix 4.5.

The following evaluation procedure was used for AAG as well as for the eight benchmark algorithms (see Section 4.1.2). Stated differently, the only difference between the evaluation procedure of the various methods was the subspace analysis algorithm used.

First, each subspace analysis algorithm was learned over the training set. Then, the same training set was used to train the anomaly detection algorithm (we used minimum volume set (MV set); more details are provided in Section 4.1.3) in each one of the obtained subspaces.

The missing values in each attribute of the training data set were replaced by the mean value in case of noncategorical values attributes and by the most frequent symbol in case of categorical values attributes (see, e.g., Bishop and Nasrabadi 2006). The missing values in the test data set were accordingly replaced by the mean and most frequent values computed from the training data set. Because AAG, ENCLUS, and 4S make use of elements of information theory to combine subspaces, we discretized the continuous-valued attributes in the training set using the *equally frequency* technique following the recommendations by Garcia et al. (2012).

After training the anomaly detection algorithm over each subspace, a weighting factor was computed to aggregate the ensemble elements at the test stage. To this purpose, we followed the recommendations of Menahem et al. (2013). More specifically, the training data were split randomly into a new training data set, which was used to generate the subspaces as well as to train the MV set model in each subspace, and into a validation data set, which was used to estimate the generalization error of each trained model. That is, the validation data (i.e., majority class) were used to compute the weighting factors as the average error of the MV set in each subspace to be used as a "belief factor" of how good each trained model represents the normal data in each subspace. Note that because the validation data contain only normal observations, only one type of error is considered (i.e., normal observations that were classified as anomalies). The aggregation of the ensemble elements was incorporated by summing up the weighted factors of the subspaces as follows. Given an observation $\boldsymbol{x}$ from the validation data set, we computed $\hat{y} = \sum_{i=1}^{M} w_i g_i(\boldsymbol{x}) \geq \rho$, where $\hat{y} \in \{0, 1\}$ denotes if the observation is normal (i.e., $\hat{y} = 1$); w_i denotes the weighting factor of subspace $i = 1, 2, \ldots, |T|$; $|T|$ denotes the total number of subspaces; $g_i(\boldsymbol{x})$ represents the MV set model trained on subspace I; and ρ denotes a threshold computed as the weighted number of subspaces that guaranteed at maximum α error rate on the validation data set. As default, we used $\alpha = 0.05$, as it is typically used in many academic and industrial applications. It is important to emphasize that, in all of our experiments, we only used the normal observations to find subspaces and to train the ensemble for anomaly detection because only this information is assumed to be available at the training stage. In other words, our training set did not contain any abnormal observations at all.

Finally, the trained ensemble for anomaly detection was evaluated over the test set (containing both normal and abnormal observations).

As measures of performance, we examined the F1 score, the run time in seconds, and the SI (see Section 4.1.4).

The F1 score is calculated as $F1 = 2TP/(2TP + FN + FP)$, where TP, FP, and FN denote, respectively, the number of true positives (true anomaly samples), the number of false positives (the number of normal samples classified as anomalies), and the number of false negatives (the number of anomalies classified as normal samples).

All experiments were executed 20 times, where in each repetition, the data set was resplit randomly into training and test sets. The reported results are averages over the 20 different repetitions.

All of the experiments were conducted on a standard MacBook Pro running Mac OS X version 10.6.8 with a 2.53-GHz Intel Core 2 Duo processor and 8 GB of DRAM.

4.1.1. The Considered Settings.

As explained, we considered four different experimental settings. Two of them are described, a third setting is described in Online Appendix 4.3, and a fourth setting is given in Online Appendix 4.5.

4.1.1.1. Setting 1—Anomaly Detection (Adding Gaussian Noise).

In this setting, we simulated a case where anomalies were generated by adding zero-mean Gaussian noise to normal observations, but only over a subset of the attributes and not over the entire data space. More specifically, we first identified the majority class for each one of the data sets. Then, we sampled 70% of the observations associated with the majority class. These observations were considered as normal observations and served as the training set. The remaining 30% of the observations associated with the majority class were split into two equally sized data sets. One of the newly split sets was kept as is, representing normal observations in the test set. For the other split, we randomly selected K attributes from the entire data space and added zero-mean Gaussian noise on the projected subspace of these attributes, representing anomalies in the test set. The variance-covariance matrix of the Gaussian noise was set to be diagonal, where the diagonal elements are the variances of the K attributes in the selected subspace. The described procedure was repeated with different percentages of perturbed attributes (i.e., 1%, 3%, 5%, 7%, and 10%–100% with steps of 10%).

4.1.1.2. Setting 2—Novelty Detection.

In this setting, we simulated a case where the abnormal observations represent a previously unseen class (i.e., novelties as defined in the literature). For this purpose, we used the approach that was applied in several previous studies (see, e.g., Aggarwal and Yu 2001; Lazarevic and Kumar 2005; Keller et al. 2012; Nguyen et al. 2013, 2014). Similar to the first setting described, we first sampled 70% of the observations associated with the majority class. These observations represented normal observations and served as the training set. The remaining 30% of the observations associated with the majority class represented normal observations in the test set. Finally, 10% of the observations associated with the remaining classes (i.e., not with the majority class) represented novelties in the test.

Table 5 in Online Appendix 4.1 shows the number of normal and abnormal instances for each one of the data sets for each one of the settings.

4.1.2. Benchmark Algorithms for Subspace Analysis.

As benchmark methods against the proposed AAG method, we selected eight classical and state-of-the-art algorithms, representing a wide range of techniques. Specifically, FB (Lazarevic and Kumar 2005) and isolation forest (iForest) (Liu et al. 2008) were selected to represent the random selection of attributes. HiCS was selected to represent the a priori-based technique (Keller et al. 2012). ENCLUS (Cheng et al. 1999), EWKM (Jing et al. 2007), and AFG k-means (Gan and Ng 2015) were selected to represent the clustering-based techniques. Finally, CMI (Nguyen et al. 2013) and 4S (Nguyen et al. 2014) were selected to represent a category of algorithms that search for subspaces based on information-theoretic measures.

With regard to AAG, subsets of three attributes were used to approximate the evaluation of Equation (4) and appear to generate a good trade-off between highly informative subspaces and a reasonable run time. Our implementation of FB sampled attributes from a uniform distribution over the range $[p/2, p]$ as suggested in Lazarevic and Kumar (2005). The total number of subspaces (i.e., ensemble size) was set to 20 according to the authors' suggestion.

Our implementation of the iForest algorithm tightly followed the work published in Liu et al. (2008). iForest generates and ensembles decision trees, where attributes and splits are randomly selected. Each tree, denoted as an isolation tree, is built recursively by partitioning the given feature space until samples are isolated. The height (node depth) of each sample is mapped to a score. Normal samples are then expected to be associated leaves of average height, whereas abnormal samples are expected to be associated leaves with lower height.

The HiCS algorithm was executed with its default parameters, and we selected the first 400 subspaces obtained by the algorithm according to Keller et al. (2012). As for ENCLUS, we implemented the version ENCLUS_SIG as described in Cheng et al. (1999) because it is the faster variant of the algorithm. We also included the pruning option described by the authors to speed up the subspace analysis. The tuning of the parameters required in ENCLUS resulted in an extensive grid search over the parameter space for each data set used in the experiments. Regarding the clustering algorithms EWKM and AFG k-means, we applied the well-known technique proposed in Sugar and James (2003) to set the number of clusters. In particular, for AFG k-means, we used the default parameters recommended in Gan and Ng (2015) and the group of features per cluster delivered by the algorithm as the set of subspaces. For EWKM, we selected the attributes in each cluster with the highest weighting factor, generating as many subspaces as the number of clusters. Finally, for 4S and CMI, we followed the default parameterization suggested in the original articles.

All algorithms, with the exception of HiCS, CMI, and 4S, were implemented in MATLAB R2009b, whereas for HiCS, CMI, and 4S, we made use of the publicly available code.

4.1.3. The Anomaly Detection Algorithm. As explained, after executing the subspace analysis algorithm, an anomaly detection algorithm was trained on each one of the obtained subspaces. We used MV set as presented in Park et al. (2010) as the anomaly detection algorithm.

MV set, which is based on the plug-in estimator, provides asymptotically the smallest type II error (false-negative error) for a given fixed type I error (false-positive error). More specifically, the MV set aims at finding the minimal support of a distribution for which the probability of each element of the support is at least as high as a predefined minimal threshold. Accordingly, the anomaly detection rule reduces to the following principle; if a new sample belongs to the minimum volume, then the new sample is considered as normal observation. Otherwise, it is labeled as abnormal. In our experiments, we used a fixed type I error of 0.05.

Park et al. (2010) used PCA to reduce the data dimensionality before applying kernel density estimation (see, e.g., Bishop and Nasrabadi 2006) to compute the empirical probability that was later used to find the minimum volume set. In the experiments, Park et al. (2010) selected two principal components as it is well known that higher dimensionality often worsens the performance of kernel density estimators (Scott 2015). We followed this approach but selected the principal components that describe 90% of the variance. If the mapped dimension of the data was found to be larger than two, then we used a Gaussian mixture model (GMM) (see, e.g., Bishop and Nasrabadi 2006) to compute the supporting empirical distribution and applied it to estimate the MV set. The number of components in the GMM was set to obtain a minimal Akaike information criterion.

We also applied another anomaly detection algorithm in our experiments, namely OC-SVM (Schölkopf et al. 2005), as a classical anomaly detection algorithm, yet we found that in most cases, MV set achieved better performance, required fewer parameters to be tuned, and was faster to train on the same data.

4.1.4. The Stability Index *SI*. Often, domain experts prefer subspace analysis methods that obtain not only acceptable performance values in detecting anomalies but also, show stability in the set of subspaces. Although low stability does not necessarily imply low performance rates, in many cases, low stability follows from fundamental problems in the subspace search process (Somol and Novovičová 2010).

Derived from the work presented in García-Torres et al. (2016), we propose a way to compute the stability index of subspace analysis methods. We denote a set of subspaces from one run of a subspace analysis method as $T_i = \{S_{i,m}\}_{m=1}^{M_i}$, where i symbolizes the run index, M_i is the number of subspaces in the run i, and $S_{i,m}$ symbolizes one of the M_i subspaces in the set T_i. We further denote the set of all subspaces from L algorithm runs of a subspace analysis method by $\boldsymbol{S} = \{S_{i,m} \in T_i,\ \forall m = 1, 2, \ldots, M_i \text{ and } i = 1, 2, \ldots, L\}$. Additionally, we denote allsubspaces in $\boldsymbol{S}$ that contain k attributes by Λ_k (i.e., $\Lambda_k = \{S_i, S_j \in \boldsymbol{S} : |S_i| = |S_j| = k,\ \forall i, j = 1, 2, \ldots, |\boldsymbol{S}| \text{ and } i \neq j\}$, where $|\cdot|$ denotes the cardinality of a set). Thus, $|\boldsymbol{S}|$ denotes the total number of subspaces obtained after L executions of the algorithm.

The approach for estimating the stability index $SI(\boldsymbol{S})$ for the set $\boldsymbol{S}$ consists of assessing the stability index for each set of equally sized subspaces (i.e., Λ_k) and then, averaging the latter values. Assuming that there are L sets of equally sized subspaces and each set $l = 1,\ 2 \ldots L$ is denoted as $\Lambda_{k(l)}$, where $k(l)$ refers to the number of attributes in the set l, then the stability index is defined as

$$SI(\boldsymbol{S}) = \frac{1}{L} \sum_{l=1}^{L} \frac{2}{N_l(N_l - 1)} \sum_{i=1}^{N_l - 1} \sum_{j=i+1}^{N_l} J(S_{i,l}; S_{j,l}), \tag{7}$$

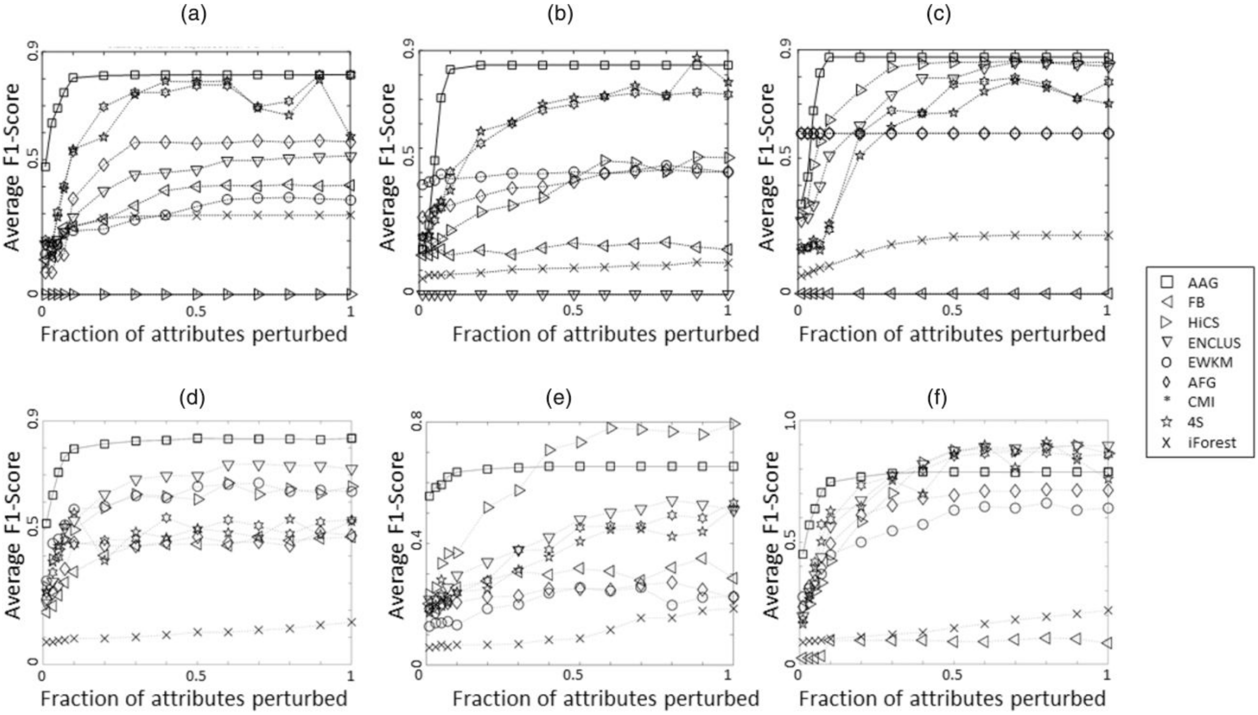
where J is the Jaccard index, N_l is the number of subspaces in the set $\Lambda_{k(l)}$, and $S_{i,l}, S_{j,l} \in \Lambda_{k(l)}$. It is easy to see that $0 \leq SI(\boldsymbol{S}) \leq 1.0$, where values closer to 1.0 correspond to more stable solutions. Indeed, if all subspaces $S_{i,l}$ and $S_{j,l}$ have the same result, the double-sum term on the right side of Equation (7) is equal to $N_l(N_l - 1)/2$. Therefore, the first sum results in L, computing $SI(\boldsymbol{S}) = 1$.

4.2. Results

The following subsections report the detection performance results under the settings described in Section 4.1.1. Based on the experimental evaluation, we provide a detailed comparison of the proposed AAG method versus the different benchmark methods. Finally, we report the run time that was required to train the various subspace analysis methods.

4.2.1. Setting 1—Anomaly Detection (Adding Gaussian Noise). Figure 2 shows the resulting averaged F1 scores as a function of the fraction of attributes synthetically perturbed by additive zero-mean Gaussian noise on 6 of the 25 considered data sets. In all cases, the maximum error rate α was set to 0.05 (see Section 4.1 for more details). The x axis indicates the fraction of perturbed attributes with respect to the total number of attributes, and the y axis shows the averaged F1 scores

Figure 2. Setting 1—Averaged F1 Score as a Function of the Fraction of Attributes Synthetically Perturbed by Additive Zero-Mean Gaussian Noise for Different Subspace Analysis Methods



Notes. (a) Thyroid data set. (b) Features Kar data set. (c) Arrhythmia data set. (d) Breast data set. (e) Fourier data set. (f) Faults data set.

over 20 repetitions of the experiment. As seen in the figure, the proposed AAG method considerably outperforms the other methods when the fraction of perturbed attributes is lower than ~ 0.3. When the fraction of perturbed attributes is higher than 0.3, AAG performance remains stable and becomes comparable with that of HiCS. Furthermore, it seems that AAG's performance is less affected by the fraction of perturbed attributes (note the lower variance in its F1 score values), whereas the other methods are more affected by these percentages.

Table 1 shows the averaged F1 scores obtained by the different subspace analysis methods, for all 25 data sets, when zero-mean Gaussian noise is added to 10% of the attributes. In each row (i.e., data set), the two highest average F1 score results, obtained by the two best-performing subspace analysis methods, are indicated by bold numbers.

As seen from Table 1, in 18 of the 25 data sets, AAG is included in the two best-performing subspace analysis methods. In most of these cases, when AAG is the second best, the difference from the best method is marginal and nonsignificant. On the other hand, in many of the cases that AAG is ranked as the best method, the difference from the second-best method is significant. In eight cases, ENCLUS is included in the two best-performing methods; in four of these cases, it outperforms AAG marginally, whereas in two of these cases, AAG outperforms it significantly. In eight cases, HiCS is included in the two best-performing methods; in four of these cases, it outperforms AAG marginally, whereas AAG outperforms it in most of the cases significantly. CMI is included four times in the two best-performing methods, outperforming AAG in a single data set only (the cover data set), and outperformed by AAG all other cases. FB is included five times among the two best-performing methods, outperforming AAG in four of these cases, especially when the data set dimensionality is relatively small. All other methods are left way behind in terms of their performance.

Our evaluation shows that AAG performs well at detecting anomalies when they occur in relatively small subspaces. The superiority of AAG in such cases can be explained by three main directions. First, the use of the proposed multiattribute distance allows AAG to identify highly qualitative subspaces. Second, during the subspace combination process, AAG does not discard even a single attribute—attributes that might be necessary in the testing phase to identify anomalies that are not available for training. Third, all other benchmark methods require some tuning of parameters, where among them, one can find the number of subspaces to generate that is extremely critical. Determining the right number of subspaces is, in general, a nontrivial task, which is usually achieved by validating the framework on test data. Such a procedure may result in

Table 1. Setting 1—Averaged F1 Scores of the Nine Anomaly Detection Ensembles over the 25 UCI Repository Data Sets

Data set	AAG	FB	HiCS	ENCLUS	EWKM	AFG *k*-means	CMI	4S	iForest
KDDCup99 (http)	0.482	0.499	0.422	0.399	0.000	**0.517**	0.441	0.442	**0.529**
KDDCup99 (smpt)	**0.044**	0.036	0.041	0.029	0.000	**0.045**	0.038	0.034	0.039
Thyroid	**0.803**	0.252	0.000	0.289	0.236	0.591	**0.663**	0.603	0.254
Mammography	**0.594**	0.579	0.488	**0.598**	0.489	0.212	0.501	0.473	0.240
Glass	**0.541**	0.376	0.409	**0.553**	0.514	0.375	0.324	0.324	0.000
Breast cancer	**0.797**	0.344	0.498	**0.532**	0.573	0.449	0.441	0.445	0.096
Zoo	0.537	**0.572**	0.473	**0.605**	0.433	0.445	0.336	0.342	0.000
Cover	0.531	**0.556**	0.123	0.197	0.317	0.551	**0.668**	0.497	0.218
Wine	**0.478**	0.379	0.359	0.428	0.401	**0.440**	0.397	0.377	0.000
Pen digits	**0.747**	0.402	0.293	**0.627**	0.543	0.524	0.241	0.341	0.091
Letter	0.523	0.289	**0.564**	**0.640**	0.425	0.337	0.415	0.511	0.182
Waveform 1	0.228	**0.468**	**0.548**	0.000	0.433	0.431	0.442	0.440	0.139
Faults	**0.747**	0.484	0.424	0.564	0.448	0.550	**0.594**	0.494	0.104
Dermatology	**0.702**	0.401	0.580	**0.610**	0.436	0.564	0.566	0.568	0.094
Satimage	**0.346**	0.186	0.314	**0.365**	0.323	0.303	0.234	0.239	0.098
Waveform 2	0.268	**0.637**	0.513	0.573	**0.585**	0.513	0.398	0.387	0.135
Segmentation	**0.720**	0.577	**0.733**	0.658	0.608	0.514	0.605	0.625	0.000
Lung cancer	**0.753**	0.421	**0.704**	0.660	0.448	0.378	0.627	0.627	0.000
Sonar	**0.430**	0.246	**0.499**	0.373	0.232	0.299	0.391	0.390	0.059
Features Pix	0.432	0.497	0.381	0.452	**0.564**	**0.595**	0.327	0.387	0.057
Audiology	**0.712**	0.485	**0.674**	0.000	0.370	0.000	0.492	0.397	0.000
Feature Fourier	**0.635**	0.256	**0.370**	0.294	0.147	0.207	0.238	0.230	0.067
MNIST	**0.873**	**0.865**	0.579	0.778	0.833	0.836	0.682	0.668	0.477
Features Kar	**0.923**	0.162	0.264	0.000	0.474	0.365	**0.504**	0.412	0.083
Arrhythmia	**0.873**	0.000	**0.643**	0.510	0.592	0.592	0.239	0.339	0.103

Note. The two highest averaged F1 scores are indicated by bold numbers.

discarding subspaces as a result of some criterion during the training stage that can impact the performance of the anomaly detection ensemble during the testing phase, when new unseen data samples arrive.

Our findings were found to be statistically significant as described in Online Appendix 4.4.1.

4.2.2. Setting 2—Novelty Detection.

Table 2 shows the averaged F1 scores obtained in the novelty detection setting. As seen from Table 2, in 15 of the 25 data sets, AAG is included among the two best-performing subspace analysis methods; in six cases, it achieves the best performance. CMI seems to be the second-best method in the novelty detection setting, being included nine times among the two best-performing methods, whereas in six of these cases, it is either very close to AAG or underperforms it. AFG *k*-means comes next, being included six times among the two best-performing methods, with a relatively close performance of AAG in three of these cases. HiCS follows next, being included five times among the two best-performing methods, with a relatively close performance of AAG in three of these cases. ENCLUS, iForest, and EWKM were found to be less effective in detecting novelties and were included among the two best-performing methods five, four, and four times, respectively.

Our findings were found to be statistically significant as described in Online Appendix 4.4.2.

4.2.3. Detailed Comparison.

The rest of this subsection provides a more detailed comparison of AAG against the other benchmark methods.

With respect to the FB subspace method, the results obtained in all three settings were of relatively low performance with respect to AAG. In the two anomaly detection settings, FB's selection of subspaces obtained a better performance in detecting random perturbations on the attribute space as well as in detecting samples when these came from combined anomaly classes. Nevertheless, in the novelty detection setting, FB's performance was even lower. A possible reason for this might be that random combinations, as done in FB, are less prone to detect inherent correlations among different attributes that usually exhibit different data classes.

Unlike HiCS, AAG succeeds in finding a smaller number of subspaces that can be directly applied. The reason for this lies in the search strategy of HiCS, which is based on the a priori approach and on randomly permuted attributes to reduce the algorithm complexity. HiCS retrieves several hundreds of subspaces that afterward have to be filtered in some way. This can be observed from the obtained results in the anomaly detection evaluation where on average, the HiCS method misses finding moderate deviations in the data set.

With respect to ENCLUS, although it does not require to set the number of generated subspaces in advance, it does require three other parameters as input, such that their tuning requires an extensive grid search over the

Table 2. Setting 2—Averaged F1 Scores of the Nine Anomaly Detection Ensembles over the 25 UCI Repository Data Sets

Data set	AAG	FB	HiCS	ENCLUS	EWKM	AFG *k*-means	CMI	4S	iForest
KDDCup99 (http)	**0.492**	0.301	0.301	0.330	0.000	0.407	0.291	0.288	**0.495**
KDDCup99 (smtp)	**0.041**	0.020	**0.044**	0.029	0.000	0.041	0.024	0.018	0.038
Thyroid	**0.687**	0.339	0.587	0.357	0.501	0.201	**0.597**	0.537	0.566
Mammography	**0.522**	0.379	0.404	0.505	0.389	0.218	0.330	0.395	**0.610**
Glass	**0.550**	0.441	0.333	0.504	0.283	**0.575**	0.457	0.412	0.160
Breast cancer	**0.902**	0.396	0.616	0.655	0.857	0.891	**0.904**	0.901	0.229
Zoo	**0.581**	0.526	0.460	**0.576**	0.527	0.522	0.161	0.361	0.167
Cover	0.514	0.442	0.091	0.122	0.290	0.219	**0.598**	0.480	**0.588**
Wine	**0.570**	0.424	0.400	0.456	0.561	**0.583**	0.523	0.353	0.192
Pen digits	**0.827**	0.387	0.637	0.579	0.743	0.770	**0.875**	0.340	0.249
Letter	0.173	0.337	**0.553**	**0.630**	0.275	0.181	0.169	0.435	0.407
Waveform 1	0.634	0.508	0.602	0.533	**0.746**	0.712	**0.728**	0.449	0.299
Faults	0.377	**0.573**	0.448	**0.488**	0.394	0.247	0.236	0.595	0.291
Dermatology	**0.834**	0.578	0.517	0.460	**0.812**	0.770	0.782	0.619	0.262
Satimage	**0.810**	0.337	0.363	0.411	**0.804**	0.797	0.801	0.272	0.236
Waveform 2	0.201	0.455	0.516	**0.663**	0.516	**0.538**	0.298	0.426	0.297
Segmentation	0.813	0.758	0.599	0.631	**0.845**	**0.826**	0.746	0.561	0.000
Lung cancer	0.694	0.529	**0.705**	0.659	0.385	0.270	**0.736**	0.625	0.000
Sonar	0.236	0.305	**0.417**	0.349	0.385	**0.453**	0.221	0.393	0.215
Features Pix	**0.855**	0.378	0.432	0.572	0.531	0.474	**0.792**	0.415	0.233
Audiology	**0.743**	0.675	**0.698**	0.378	0.410	0.387	0.521	0.469	0.229
Feature Fourier	**0.846**	0.413	0.375	0.277	0.692	0.715	**0.844**	0.278	0.196
MNIST	0.661	0.677	0.420	0.441	0.722	**0.735**	0.595	0.600	**0.744**
Features Kar	**0.846**	0.277	0.204	0.484	0.577	0.731	**0.794**	0.503	0.262
Arrhythmia	0.468	**0.572**	0.495	**0.592**	0.495	0.495	0.431	0.289	0.240

Note. The two highest averaged F1 scores are indicated by bold numbers.

support of the parameters. In contrast to FB, one can see that ENCLUS often performs better in the case of anomaly detection applications, but its performance degrades as a subspace method for novelty detection ensembles. On the other hand, ENCLUS manages to generate a stable set of subspaces, mainly because of the a priori search mechanism. Such a search strategy enables ENCLUS to find thousands of subspaces with a relatively small number of attributes, where several subspaces might have redundant results. Yet, because of the high number of subspaces that ENCLUS generates, potential subspaces that may find abnormal data samples are downgraded in the averaging computation of the scores. An interesting research direction might be then to evaluate different subspace combinations when the number of ensemble subspaces is high.

CMI resulted in lower performance than the proposed AAG algorithm both for novelty detection and for anomaly detection. Nevertheless, CMI showed better results in the novelty detection setting. It seems that its subspace generation managed to combine relevant subspaces that captured the correlation among important attributes. On the other hand, 4S was not included among the two best-performing subspace methods. The 4S method requires us to a priori set the maximal number of attributes, which turns out to be critical for finding highly qualitative subspaces. This is manifested in the obtained results for all three examined settings.

The subspace clustering methods EWKM and AFG *k*-means follow AAG, FB, HiCS, and ENCLUS in terms of their performance. The poorer performance with respect to all other methods is because of the fact that attributes are discarded from the set of subspaces. Consequently, neither novel nor abnormal samples can be efficiently identified. Additionally, we found that it was not trivial to set the number of clusters—a critical parameter for both methods. In both subspace-clustering methods, the number of clusters has a major impact on the selected subspaces when optimizing the extended *k*-means cost objective.

Finally, the iForest method achieved a poorer performance than the proposed AAG method in settings where the model is trained using only normal data and then applied to abnormal samples. Often, the iForest method is applied to outlier detection problems: that is, when abnormal and normal data samples coexist in the training data set. It seems that only in cases where the unexpected data samples are well separated from the normal data, iForest manages to obtain a good representation of the normal data. This may be the case when abnormal samples are almost homogeneously distributed among the subspaces that were obtained during the random training of the iForest ensemble. Nevertheless, in common real-world cases, the tree depth used to compute the threshold as anomaly score is not significant enough to generalize to unseen abnormal samples.

Nevertheless, it is important to acknowledge AAG's limitations. First, as noted in Lemma 3, AAG's run-time complexity is proportional to p^3, where p is the number of attributes. This property can impose a serious limitation for data sets with a very large number of attributes. Second, AAG's superiority stems from its inherent assumption that anomalies are occluded around a relatively small number of attributes, and in some cases, this assumption may not hold.

4.2.4. Run-Time Evaluation. We evaluated the time taken to train each of the ensemble methods over the 25 data sets considered in this study. Because the run times obtained did not differ significantly among the three settings, we show in Table 3 only the run times for setting 1.

As seen in Table 3, in none of the 25 studied cases, AAG's run time was the lowest one among the nine compared methods. HiCS and ENCLUS were found to be faster than AAG in 60% and 80% of the cases, respectively. A possible reason for this is that HiCS uses random selection of attributes to cope with the run-time requirement of the original a priori strategy. ENCLUS requires as a parameter a limit to the number of attributes in each subspace, and therefore, it finishes the execution even if the selected subspaces are far from optimal. As expected, FB and iForest were found to be faster than AAG in 80% and 92% of the cases, respectively, mainly because of their random selection of attributes. Additionally, iForest does not require building an anomaly detection model over the selected subspaces, as FB does. Therefore, in most cases, iForest outperformed FB in terms of run time.

Additional run-time analyses are reported in Online Appendix 4.6.

4.2.5. Stability Analysis. We computed the stability index $SI(S)$ for the proposed AAG method as well as for all benchmark subspaces analysis methods using the 25 data sets considered in this study. The results shown in Table 4 are obtained for setting 1 (similar results were obtained for settings 2 and 3) after executing the corresponding subspace analysis method 20 times, where the best two results are indicated with bold numbers.

From Table 4, we can see that, on average, the proposed AAG method, as well as the benchmark methods HiCS, ENCLUS, CMI, and 4S, achieve relatively stable solutions, whereas FB, EWKM, and AFG *k*-means achieved less robust sets of subspaces.

A possible explanation for the lower stability of FB is in the fact that subspaces are randomly selected. Therefore, for each algorithm run, a different set of subspaces is generated, leading to a poorer stability index. EWKM

Table 3. Averaged Run Times (in Seconds) for Executing the Subspace Analysis Method and Training the Ensembles for Each One of the Nine Subspace Analysis Methods over the 25 Studied UCI Repository Data Sets

Data set	AAG	FB	HiCS	ENCLUS	EWKM	AFG *k*-means	CMI	4S	iForest
KDDCup99 (http)	131.41	**29.31**	304.63	92.25	111.13	102.02	142.81	256.52	**56.53**
KDDCup99 (smtp)	14.88	**5.42**	142.76	11.32	12.03	**10.65**	87.03	131.92	22.40
Thyroid	6.82	2.55	68.75	**0.23**	2.06	**1.16**	13.45	22.21	2.27
Mammography	25.18	21.77	43.94	**0.11**	**1.48**	4.25	22.03	39.38	4.97
Glass	1.62	4.16	1.13	**0.35**	**0.33**	0.47	0.42	0.71	0.40
Breast cancer	0.45	4.02	3.42	**0.07**	**0.33**	0.50	0.68	0.90	0.53
Zoo	0.48	**0.06**	0.80	3.31	0.32	0.58	0.26	0.28	**0.22**
Cover	3,999.70	**0.11**	52.86	103.58	244.67	402.39	**31.43**	36.41	124.21
Wine	1.67	3.35	4.60	2.98	**0.56**	0.63	0.77	0.86	**0.27**
Pen digits	115.22	23.92	32.98	**0.88**	**1.72**	3.47	8.26	11.98	7.92
Letter	12.44	4.35	42.47	3.91	**0.82**	**1.56**	7.13	7.25	6.81
Waveform 1	148.51	8.25	23.99	30.00	**1.74**	**2.53**	5.24	5.73	3.62
Faults	31.83	2.78	4.58	36.23	**0.51**	**1.25**	2.99	4.07	2.52
Dermatology	7.14	3.43	3.57	**0.46**	**0.38**	0.55	1.14	1.68	0.83
Satimage	84.55	26.74	28.35	10.01	**3.59**	5.77	19.24	31.35	**4.80**
Waveform 2	545.14	15.57	29.39	179.17	**3.10**	4.40	9.41	16.33	**3.63**
Segmentation	0.83	2.07	94.07	0.69	**0.47**	0.56	23.08	37.33	**0.27**
Lung cancer	6.79	2.71	1.17	1.03	0.66	**0.57**	0.65	0.67	**0.25**
Sonar	37.60	2.49	11.88	84.55	**0.25**	0.41	2.24	3.41	**0.27**
Features Pix	215.06	7.43	18.98	18.41	**1.28**	**1.86**	13.13	20.91	3.19
Audiology	1.09	2.77	19.74	**0.26**	0.33	0.60	10.42	14.71	**0.19**
Feature Fourier	225.56	3.40	35.78	291.33	**0.79**	**1.12**	5.78	9.21	2.84
MNIST	704.04	72.77	135.85	37.21	**12.93**	21.94	23.10	28.06	**5.12**
Kar	334.53	5.43	57.29	100.25	**2.85**	**3.35**	19.49	31.48	6.25
Arrhythmia	2,638.30	293.42	43.52	1,086.12	2.27	**2.05**	8.76	10.62	**0.40**

Note. The two best (lowest) run times are indicated with bold numbers.

Table 4. Averaged Stability Index $SI(S)$ for Each One of the Eight Subspace Analysis Methods over the 25 Studied UCI Repository Data Sets

Data set	AAG	FB	HiCS	ENCLUS	EWKM	AFG *k*-means	CMI	4S
KDDCup99 (http)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
KDDCup99 (smtp)	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Thyroid	**0.651**	0.369	0.367	0.537	0.423	0.466	**0.611**	0.601
Mammography	0.821	0.766	0.801	0.799	**1.000**	**1.000**	**0.876**	0.812
Glass	0.653	0.478	**0.667**	**0.655**	0.309	0.291	0.622	0.631
Breast cancer	**0.501**	0.166	0.449	0.477	0.422	0.466	**0.498**	0.487
Zoo	**0.713**	0.685	**0.691**	0.685	0.408	0.419	0.644	0.635
Cover	0.823	0.732	0.804	0.813	**0.987**	**0.998**	0.798	0.809
Wine	0.617	**0.693**	0.583	**0.621**	0.289	0.313	0.587	0.601
Pen digits	0.618	0.580	**0.694**	**0.622**	0.422	0.458	0.602	0.611
Letter	0.549	0.467	0.488	0.550	0.340	0.411	**0.610**	**0.590**
Waveform 1	**0.526**	0.423	0.490	0.443	0.211	0.190	0.429	**0.511**
Faults	**0.595**	0.269	0.475	0.521	0.201	0.383	**0.588**	0.570
Dermatology	**0.652**	0.269	0.521	**0.589**	0.267	0.390	0.499	0.511
Satimage	**0.581**	0.289	0.510	**0.577**	0.402	0.431	0.544	0.561
Waveform 2	**0.579**	0.353	0.504	**0.561**	0.166	0.207	0.522	0.535
Segmentation	**0.598**	0.152	0.447	0.554	0.338	0.298	**0.590**	0.578
Lung cancer	**0.507**	0.303	0.407	0.487	**0.479**	0.471	0.402	0.446
Sonar	0.527	0.297	0.388	0.601	0.332	0.378	**0.611**	**0.612**
Features Pix	**0.691**	0.106	0.359	**0.609**	0.231	0.233	0.579	0.591
Audiology	**0.477**	0.290	0.391	**0.522**	0.112	0.134	0.378	0.401
Feature Fourier	**0.541**	0.210	0.466	0.476	0.129	0.142	0.489	**0.493**
MNIST	0.609	0.123	0.434	**0.655**	0.589	0.609	0.590	**0.612**
Features Kar	**0.509**	0.151	0.472	**0.510**	0.148	0.201	0.465	0.490
Arrhythmia	0.573	0.237	**0.583**	0.576	0.281	0.229	**0.579**	0.565

Note. The two best results are indicated with bold numbers.

and AFG *k*-means select subspaces by minimizing a distortion function that involves the Euclidean distance. Thus, changes in the data set produced by the shuffling process have higher impact than their competitors, leading to a relatively lower stability in the generated subspaces.

Methods based on inherent information within the data set suffer less from variations in the data set. In particular, for the methods HiCS and ENCLUS, we found that the high number of selected subspaces contributes to the stability index. Specifically, both methods are based on the a priori mechanism, and henceforth, both methods tend to select several hundred subspaces, where a small portion of attributes differs among subspaces.

Nevertheless, HiCS results are less robust than ENCLUS because of two reasons. First, it includes a random permutation of attributes to overcome the time-consuming a priori search. Second, only the first few hundred generated subspaces are usually selected, negatively impacting the overall stability index. CMI and 4S were more robust to changes in the data set with respect to the previously mentioned algorithms but still fall behind the proposed AAG method in stability. Recall that CMI applies the *k*-means clustering to compute the conditional mutual information, and therefore, the random data set shuffling produces deterioration in the stability index. The 4S method, for its part, selects a specific number of attributes after computing the total correlation, and henceforth, the stability index shrinks. The pseudometric used in the search for subspaces in AAG was less influenced by the shuffling mechanism, leading to subspaces comprising almost the same attributes.

5. Summary and Future Work

In this paper, we introduced the AAG subspace analysis algorithm that aims at finding highly informative subspaces for anomaly detection ensembles as well as other analytics tasks. Similar to other state-of-the-art methods for subspace analysis, AAG searches for subspaces with highly correlated attributes. In order to assess how correlative a subset of attributes is, AAG proposes a new informational measure, which was derived from previous information theory measures over sets of partitions. We then suggest a method to approximate the proposed measure in cases where the number of attributes is large. Relying on the newly suggested measure, AAG applies a variation of the well-known agglomerative algorithm to search for highly correlated subspaces. Our variation of the agglomerative algorithm also applies a pruning rule that reduces the potential redundancy in the final set of subspaces.

As a result of combining the agglomerative approach with the suggested measure, AAG avoids any tuning of parameters when generating the subspaces. Moreover, based on an extensive empirical study, we show that AAG outperforms other classical and state-of-the-art subspace analysis algorithms, specifically when it was used for ensemble-based anomaly detection (settings 1 and 3). In this case, we found that AAG training time is lower and that it can better distinguish between normal and abnormal observations. AAG also outperformed other subspace analysis methods when it was used for ensemble-based novelty detection (setting 2): that is, when new classes that were not present during the training stage of the ensemble arise in the testing stage. Finally, we demonstrated how the obtained subspaces can be used in other analytical tasks, such as forecasting based on exogenous variables and clustering by analyzing a real-world retail data set (setting 4). Thus, the subspaces generated by AAG can be used in various applications, such as anomaly detection, novelty detection, forecasting, and clustering.

Although in some cases, AAG demonstrated a faster training time than other state-of-the-art algorithms, its run-time complexity is proportional to p^3, where p is the number of attributes. In principle, this property can impose a serious limitation for data sets with a very large number of attributes. Nevertheless, it is important to note that (i) executing AAG is performed once (and typically, in an offline procedure); (ii) the run-time complexity can be improved considerably using parallelization, as explained in Online Appendix 3.3; (iii) the run-time complexity can further be improved by predecomposition of the feature set (e.g., by applying simpler correlation measures or even randomization); and (iv) using AAG can reduce considerably the time spent during the training and/or inference phases of the learning algorithm used on top of it (see Online Appendix 4.6).

In the first anomaly detection setting (setting 1), where random noise was added to normal observations, AAG obtained considerably better results than the other benchmark methods, specifically when noise was added to a relatively small number of attributes. However, when the noise was added to the entire data space, AAG lost its superiority. Thus, in cases where noise is spread sporadically over all attributes, it could be better to use simpler anomaly detection algorithms (not necessarily ensembles) to gain faster run times.

Recall that AAG searches for highly correlated subspaces, but it does not necessarily find the optimal set of subspaces for two main reasons. (i) The computed measure for a subset is approximated, and (ii) the agglomerative algorithm is inherently a greedy one. It would be interesting to analyze the optimality boundaries obtained by AAG and explore whether certain variations of it may result in better performance boundaries.

When preparing the data sets for the novelty detection task (setting 2), we randomly sampled 10% of the minority classes that were only added to the test set. It would be interesting to experiment with other sample percentages as well as their relative quantity compared with the majority class in order to analyze their impact on the detection performance of the trained ensembles.

AAG addresses the case where no separation is made between normal observations (i.e., there exists only one normal class). More specifically, in settings 1 and 2, all normal observations are taken from a single class. In setting 3, although the normal observations can be taken from multiple classes, they are unified into a single normal class, and the separation between the underlying classes is not transparent to the algorithm. In future work, we aim to extend AAG's usage to data sets with multiclass normal observations. Although the trivial way of doing so is to apply AAG on each one of the normal classes separately (and unify the sets of subspaces), we would like to utilize jointly the information available in the different classes to find higher-quality subspaces.

Finally, another research direction is extending AAG to find subspaces in dynamic environments, where the probability distribution of the normal observations may change over time. Under such a scenario, we intend to first find a base set of subspaces and then, to update this set incrementally when new normal observations become available.

Endnote

[1] Note that in order to reduce run-time complexity considerably, we do not iterate over all pairs of subsets in $S^{(i)}$ but only on pairs that include A_i and another subset (i.e., A_k) from S_0, denoted by $\boldsymbol{A}_i$ and $\boldsymbol{A}_k$.

References

Aggarwal CC, Subbian K (2012) Event detection in social streams. Ghosh J, Liu H, Davidson I, Domeniconi C, Kamath C, eds. *Proc. 2012 SIAM Internat. Conf. Data Mining* (SIAM, Philadelphia), 624–635.

Aggarwal CC, Yu PS (2001) Outlier detection for high dimensional data. Sellis T, Mehrotra S, eds. *Proc. 2001 ACM SIGMOD Internat. Conf. Management Data* (ACM, New York), 37–46.

Bache K, Lichman M (2013) *UCI Machine Learning Repository* (University of California School of Information and Computer Science, Irvine, CA).

Bacher M, Ben-Gal I, Shmueli E (2016) Subspace selection for anomaly detection: An information theory approach. *2016 IEEE Internat. Conf. Sci. Electr. Engrg. (ICSEE)* (IEEE, Piscataway, NJ), 1–5.

Bacher M, Ben-Gal I, Shmueli E (2017) An information theory subspace analysis approach with application to anomaly detection ensembles. Fred ALN, Filipe J, eds. *Proc. 9th Internat. Joint Conf. Knowledge Discovery Knowledge Engrg Knowledge Management—KDIR* (SciTePress, Setúbal, Portugal), 27–39.

Bajovic D, Sinopoli B, Xavier J (2011) Sensor selection for event detection in wireless sensor networks. *IEEE Trans. Signal Processing* 59(10):4938–4953.

Ben-Gal I (2010) Outlier detection. Maimon O, Rokach L, eds. *Data Mining and Knowledge Discovery Handbook* (Springer, Boston), 117–130.

Ben-Gal I, Morag G, Shmilovici A (2003) Context-based statistical process control: A monitoring procedure for state-dependent processes. *Technometrics* 45(4):293–311.

Bishop CM, Nasrabadi NM (2006) *Pattern Recognition and Machine Learning*, vol. 4 (Springer, New York), 738.

Breiman L (2001) Random forests. *Machine Learn.* 45(1):5–32.

Chandola V, Banerjee A, Kumar V (2007) Outlier detection: A survey. *ACM Comput. Surveys* 14:15.

Cheng CH, Fu AW, Zhang Y (1999) Entropy-based subspace clustering for mining numerical data. Fayyad U, Chaudhuri S, Madigan D, eds. *Proc. Fifth ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 84–93.

Cover TM, Thomas JA (2006) *Elements of Information Theory*, 2nd ed. (John Wiley & Sons, Hoboken, NJ).

Gan G, Ng MKP (2015) Subspace clustering with automatic feature grouping. *Pattern Recognition* 48(11):3703–3713.

Garcia S, Luengo J, Sáez JA, Lopez V, Herrera F (2012) A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Trans. Knowledge Data Engrg.* 25(4):734–750.

García-Torres M, Gómez-Vela F, Melián-Batista B, Moreno-Vega JM (2016) High-dimensional feature selection via feature grouping: A variable neighborhood search approach. *Inform. Sci.* 326:102–118.

Ge Z, Song Z (2012) *Multivariate Statistical Process Control: Process Monitoring Methods and Applications* (Springer Science & Business Media, New York).

Guyon I, Gunn S, Nikravesh M, Zadeh LA, eds. (2008) *Feature Extraction: Foundations and Applications*, vol. 207 (Springer, Berlin).

Ha J, Seok S, Lee JS (2015) A precise ranking method for outlier detection. *Inform. Sci.* 324:88–107.

Jakulin A (2005) Machine learning based on attribute interactions. Doctoral dissertation, University of Ljubljana, Ljubljana, Slovenia.

Jing L, Ng MK, Huang JZ (2007) An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data. *IEEE Trans. Knowledge Data Engrg.* 19(8):1026–1041.

Jyothsna VVRPV, Prasad R, Prasad KM (2011) A review of anomaly based intrusion detection systems. *Internat. J. Comput. Appl.* 28(7): 26–35.

Kagan E, Ben-Gal I (2013) *Probabilistic Search for Tracking Targets* (John Wiley & Sons, Hoboken, NJ).

Kagan E, Ben-Gal I (2014) A group testing algorithm with online informational learning. *IIE Trans.* 46(2):164–184.

Keller F, Muller E, Bohm K (2012) HICS: High contrast subspaces for density-based outlier ranking. Kementsietsidis A, Vaz Salles MA, eds. *Proc. 2012 IEEE 28th Internat. Conf. Data Engrg.* (IEEE, Piscataway, NJ), 1037–1048.

Kenett RS, Zacks S (2021) *Modern Industrial Statistics: With Applications in R, MINITAB and JMP* (John Wiley & Sons, Hoboken, NJ).

Kuratowski K (2014) *Introduction to Set Theory and Topology* (Elsevier, Amsterdam).

Lazarevic A, Kumar V (2005) Feature bagging for outlier detection. *Proc. Eleventh ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (ACM, New York), 157–166.

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. Giannotti F, Gunopulos D, Turini F, Zaniolo C, Ramakrishnan N, Wu X, eds. *Proc. 2008 Eighth IEEE Internat. Conf. Data Mining* (IEEE, Piscataway, NJ), 413–422.

McGill W (1954) Multivariate information transmission. *Trans. IRE Professional Group Inform. Theory* 4(4):93–111.

Menahem E, Rokach L, Elovici Y (2013) Combining one-class classifiers via meta learning. He Q, Iyengar A, Nejdl W, Pei J, Rastogi R, eds. *Proc. 22nd ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 2435–2440.

Müller E, Schiffer M, Seidl T (2010) Adaptive outlierness for subspace outlier ranking. *Proc. 19th ACM Internat. Conf. Inform. Knowledge Management* (ACM, New York), 1629–1632.

Nguyen HV, Müller E, Böhm K (2014) A near-linear time subspace search scheme for unsupervised selection of correlated features. *Big Data Res.* 1:37–51.

Nguyen HV, Müller E, Vreeken J, Keller F, Böhm K (2013) CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection. *Proc. 2013 SIAM Internat. Conf. Data Mining* (SIAM, Philadelphia), 198–206.

Park C, Huang JZ, Ding Y (2010) A computable plug-in estimator of minimum volume sets for novelty detection. *Oper. Res.* 58(5): 1469–1480.

Pimentel MA, Clifton DA, Clifton L, Tarassenko L (2014) A review of novelty detection. *Signal Processing* 99:215–249.

Rokhlin VA (1967) Lectures on the entropy theory of measure-preserving transformations. *Russian Math. Surveys* 22(5):1–52.

Schölkopf B, Smola A, Müller KR (2005) Kernel principal component analysis. *Artificial Neural Networks—ICANN'97: 7th Internat. Conf. Proc.* (Springer, Berlin), 583–588.

Scott DW (2015) *Multivariate Density Estimation: Theory, Practice, and Visualization* (John Wiley & Sons, Hoboken, NJ).

Simovici D (2007) On generalized entropy and entropic metrics. *J. Multiple-Valued Logic Soft Comput.* 13(4/6):295–320.

Sinai IG, Sinaj JG, Sinai YG (1976) *Introduction to Ergodic Theory*, vol. 18 (Princeton University Press, Princeton, NJ).

Somol P, Novovičová J (2010) Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. *IEEE Trans. Pattern Anal. Machine Intelligence* 32(11):1921–1939.

Steinwart I, Hush D, Scovel C (2005) A classification framework for anomaly detection. *J. Machine Learn. Res.* 6(2):211–232.

Sugar CA, James GM (2003) Finding the number of clusters in a data set: An information-theoretic approach. *J. Amer. Statist. Assoc.* 98(463):750–763.

Tarassenko L, Hann A, Patterson A, Braithwaite E, Davidson K, Barber V, Young D (2005) BiosignTM: Multi-parameter monitoring for early warning of patient deterioration. *Proc. 3rd IEE Internat. Seminar Medical Appl. Signal Processing 2005* (IEEE, Piscataway, NJ), 71–76.

Watanabe S (1960) Information theoretical analysis of multivariate correlation. *IBM J. Res. Development* 4(1):66–82.

Yianilos PN (2002) Normalized forms for two common metrics. Report No. 91–082, NEC Research Institute, Princeton, NJ.