Data Imputation Through the Identification of Local Anomalies

Huseyin Ozkan, Ozgun Soner Pelvan, and Suleyman S. Kozat, *Senior Member, IEEE*

Abstract— We introduce a comprehensive and statistical framework in a model free setting for a complete treatment of localized data corruptions due to severe noise sources, e.g., an occluder in the case of a visual recording. Within this framework, we propose: 1) a novel algorithm to efficiently separate, i.e., detect and localize, possible corruptions from a given suspicious data instance and 2) a maximum *a posteriori* estimator to impute the corrupted data. As a generalization to Euclidean distance, we also propose a novel distance measure, which is based on the ranked deviations among the data attributes and empirically shown to be superior in separating the corruptions. Our algorithm first splits the suspicious instance into parts through a binary partitioning tree in the space of data attributes and iteratively tests those parts to detect local anomalies using the nominal statistics extracted from an uncorrupted (clean) reference data set. Once each part is labeled as anomalous versus normal, the corresponding binary patterns over this tree that characterize corruptions are identified and the affected attributes are imputed. Under a certain conditional independency structure assumed for the binary patterns, we analytically show that the false alarm rate of the introduced algorithm in detecting the corruptions is independent of the data and can be directly set without any parameter tuning. The proposed framework is tested over several well-known machine learning data sets with synthetically generated corruptions and experimentally shown to produce remarkable improvements in terms of classification purposes with strong corruption separation capabilities. Our experiments also indicate that the proposed algorithms outperform the typical approaches and are robust to varying training phase conditions.

Index Terms— Anomaly detection, localized corruption, maximum *a posteriori* (MAP)-based imputation, occlusion.

I. INTRODUCTION

IN MANY applications from a wide variety of fields, the data to be processed can partially (or even almost completely) be affected by severe noise in several phases, e.g., occlusions during a visual recording or packet losses during transmission in a communication channel. Such partial, i.e., localized, data corruptions often severely degrade the performance of the target application; for instance, face recognition or pedestrian detection under occlusion [1]–[4].

To reduce the impact of this adverse effect, we develop a complete and novel framework, which efficiently detects, localizes, and imputes corruptions by identifying the local anomalies in a given suspicious data instance. We emphasize that neither the existence nor, if exists, the location of a corruption is known in our framework. Moreover, the proposed algorithms do not assume a model but operate in a data-driven manner.

We consider the local corruptions as statistical deviations from the nominal distribution of the uncorrupted (clean) observations. To detect and localize corruptions, i.e., such statistical deviations, we model a corruption as an anomaly due to an external factor (communication failure in a channel or occluder object in an image), which locally overwrites a data instance and moves it outside the support of the nominal distribution. However, corruptions that we consider as examples of anomalies have further specific properties such that: 1) the corruptions in an instance are confined to unknown intervals along the data attributes, i.e., localized and 2) not only a corrupted part but also all of its subparts are anomalous. Thus, a corruption does not provide an anomaly due to an incompatible combination of normal subparts. Based on these properties that accurately model a wide variety of real life applications, we characterize the event of corruption and formulate the corresponding detection/localization as an anomaly detection problem [5]–[11].

The introduced algorithm applies a series of statistical tests with a prespecified false alarm rate to the parts of the suspicious instance after extracting the nominal statistics from a reference (training) data set of uncorrupted (clean) observations. As a result, each part is labeled as anomalous/normal and the local anomalies are identified. These parts are generated and organized through a binary tree partitioning of the data attributes, each node of which corresponds to a part of the suspicious instance (Fig. 1). Once the nodes (or parts) are labeled as anomalous/normal on this tree, the patterns of corruption are identified using the aforementioned characterization to detect and localize corruptions (Fig. 2). We point out that this localization procedure transforms the nominal distribution into a multivariate Bernoulli distribution with a success probability that precisely coincides with the constant false alarm rate of the local anomaly tests. Considering the hierarchy among the binary labels implied by the tree as a directed acyclic graph, the resulting multivariate Bernoulli distribution achieves a certain dependency structure. Under this condition, we derive the false alarm rate of the proposed framework in detecting the corruptions and show that it is a constant rate, that is, no parameter tuning is required to achieve the desired/specified false alarm rate even if the data change.

If a corruption is localized, then we impute/replace the affected attributes with the estimates of the underlying unknown true attributes. For this purpose, we additionally develop a novel maximum *a posteriori* (MAP) estimator using the score function defined in [8]. Our estimator exploits the local dependencies among the data attributes, where the locality is encoded in the binary partitioning tree. We point out that the implementation of this MAP estimator does not load extra computational cost since it utilizes the outputs of our anomaly detection approach, which are computed prior to the imputation phase. Furthermore, we also propose a novel distance measure named ranked Euclidian distance a generalization to the standard Euclidean distance, which is used in the course of the labeling of each part as anomalous/normal. The proposed distance measure is compared with the standard Euclidean distance in the experiments and shown to be superior in terms of detecting and localizing corruptions.

We conduct tests over several well-known machine learning data sets [12], [13], which are exposed to severe data corruptions. Our experiments indicate that the proposed framework achieves significant improvements after imputation up to 80% in terms of the classification purposes and outperforms the typical approaches. The proposed algorithms are also empirically shown to be robust to varying training phase conditions with strong corruption separation capabilities.

A. Related Work

In this paper, the corrupted attributes are considered to be statistically independent with the underlying unobserved true data, i.e., corrupted attributes are of no use in estimation of the uncorrupted counterparts. Hence, if one knows which attributes are corrupted in an instance, then those attributes can readily be treated as missing data [14]–[19]. For example, classification and clustering with missing data is a well-studied problem in the machine learning literature. The corresponding studies such as [16]–[18], [20], and [21] are related to inference with incomplete data [17] and generative models [20], where Bayesian frameworks [18] are used for inference under missing data conditions. Alternatively, pseudolikelihood [22] and dependency network [23] approaches solve the data completion problem by learning conditional distributions. In [24], the probability density of the missing data is modeled conditioned on a set of introduced latent variables and thereafter, a MAP-based inference is used. However, all of the studies [14]–[18], [20]–[24] either assume the knowledge of the location of the missing attributes or impose strong modeling constraints, as opposed to the model free solutions in this paper.

On the other hand, imputation is commonly used as a preprocessing tool [18]. The mixture of factor analyzers [25] approach replaces the missing attributes with samples drawn from a parametric density, which models the distribution of the underlying true data, whereas the proposed imputation techniques in [26] and [27] are both nonparametric and based on the inference of the posterior densities via certain kernel expansions. On the contrary, the MAP estimator in this paper does not even attempt to estimate the posterior density either in a parametric or nonparametric manner. Instead, the introduced method is only based on the sufficient

rank statistics. We emphasize that unlike our approach, the incomplete data approaches generally assume the knowledge of the missing attributes, i.e., they are precisely localized and provided beforehand. For example, the occluded pixels in the event of occlusion of a target object in an image cannot be known *a priori*, which requires a detection and localization step. Since the existing studies do not have such a step, an exhaustive list of the occluded pixels as the result of a manual inspection of the missing attributes is required as an input to the algorithms proposed in the corresponding literature. In this regard, our study is the first to jointly handle the issues of detecting/localizing missing attributes, i.e., corruptions, as well as their imputation in a single, complete, and comprehensive framework. Hence, the generic local corruption detection and imputation algorithm of our framework complements the missing data imputation approaches as an additional merit.

Data imputation and completion is also essential in image processing for handling corrupted images [28], [29]. In general, a corrupted image is restored by explicitly learning the image statistics [30], [31] or using neural networks [32]–[34]. These denoising studies do not attempt to localize corruptions in an image, but treat them as a noise and filter it out using statistical approaches applied to the image globally. Even though this is a valid approach for image enhancement, an attempt to correct/enhance an image globally in case of only a localized corruption might be even detrimental since the uncorrupted parts are also affected by global operations. In addition, it is not usually possible to locally impute corrupted portions using denoising approaches. There exist several studies that aim localization as well. He *et al.* [1] and Dollar *et al.* [4] indicate that occlusion, as an example of corruption, is a common phenomenon and detrimental in pedestrian detection as well as face recognition applications. In this regard, detection of occluded, i.e., corrupted, visual objects had been previously investigated in a number of studies [35]–[38]. In these studies, occlusion detection is performed using domain specific knowledge (visual cues) or external information (object geometry), which, however, is not always available in general data imputation setting. From the machine learning perspective, descriptors are extracted from various parts of the occluded object in [39] and similarly, part-based descriptors are weighted with the occlusion measure in [40] to relieve the corresponding degrading effects. Since these approaches do not directly target handling occlusions, i.e., corruptions, they only provide partial or limited solutions. Several other studies propose solutions via extracting occlusion maps [41], [42]. In [41], histogram of gradients (HOG)-based classification errors and in [42], template based reconstruction errors are used to generate such an occlusion map. However, both studies assume rigid models and significantly rely on domain specific knowledge and, in general, fail to remain applicable if the data source belongs to another domain. In this paper, we assume that data is generic and no domain information is available, yet detection and imputation of corruption is necessary for improving the subsequent processing stages, such as classification.

B. Summary of Contributions

The summary of the contributions are as follows.

1) This is the first study that jointly handles localized data corruptions in a single, complete, and comprehensive statistical framework that is designed completely model free for the goal of separating a corruption and imputing the affected data attributes. We also provide a false alarm rate (in detecting corruptions) analysis of the framework via directed acyclic graphs.

2) A novel MAP estimator for data imputation and a novel distance measure for corruption localization purposes are proposed.

3) The proposed framework is computationally efficient in the sense that: a) it effectively utilizes a binary search for corruption separation and b) the computational load due to our MAP-based imputation is insignificant.

4) We propose a novel characterization for anomalies, e.g., rarities, incompatible combinations, and corruptions.

In Section II, we provide the problem description. We then present our algorithm in Section III and the associated computational complexity in Section IV. We report the corruption detection/localization performance of the proposed algorithm as well as the improvement in classification tasks achieved by the imputation in Section V. This paper concludes with a discussion in Section VI.

II. PROBLEM DESCRIPTION

We have a possibly corrupted test instance $\mathbf{x} \in \mathbb{R}^d$ along with a set of uncorrupted (clean) independent and identically distributed observations $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{N_s}\}$ as the nominal training (reference) data, where $\mathbf{s}_i = [s_{i1}, s_{i2}, \ldots, s_{id}] \in \mathbb{R}^d \sim f_0(\mathbf{s})$, d is data dimensionality, and f_0 is the unknown nominal density. The test instance $\mathbf{x}$ is considered to be corrupted with probability π by severe noise in multiple nonoverlapping intervals along its dimensions (attributes), which are completely unknown. Suppose that for such an interval, the corruption is localized and confined to the attributes $\mathbf{x}_c^{c+\beta-1} = \{x_c, x_{c+1}, \ldots, x_{c+\beta-1}\}$ for some c and β in $[1, d]$ with $c + \beta - 1 \leq d$. We assume that the corrupted attributes are uniformly and independently distributed, $z_i \in \mathbf{x}_c^{c+\beta-1} \sim U_Z(z)$, where U_Z is the uniform distribution defined in a finite support. Moreover, Z is also statistically independent with the true data and hence, the knowledge of $\mathbf{x}_c^{c+\beta-1}$ is irrelevant to the uncorrupted counterparts. Note that this corruption model implies a total erasure of data in several unknown portions due to an independent source overwriting the attributes in those portions, e.g., an occluder in computer vision applications [1], [4]. Typically, since no information is provided about the independent source in such applications, we consider that the uniformity assumption draws a worst case scenario and it is realistic. On the other hand, $\mathbf{x}$ is considered to be uncorrupted with probability $1 - \pi$. Therefore, whether a test instance $\mathbf{x}$ includes a corruption is unknown, and it is generally modeled to be drawn from the mixture $\mathbf{x} \sim (1 - \pi) f_0(\mathbf{x}) + \pi f_1(\mathbf{x})$ [8], where f_1 is the probability density of the corrupted instances.

The density f_1 can be derived from the unknown nominal density f_0 using the described corruption model if the distributions of c, β, and the number of corrupted intervals are further specified, which is unnecessary in the context of this paper. Hypothetically, if one can correct an instance $\mathbf{x}$

drawn from the density f_1 by replacing all the corrupted attributes, e.g., $\mathbf{x}_c^{c+\beta-1}$, with the underlying true attributes, e.g., $\bar{\mathbf{x}}_c^{c+\beta-1}$, and obtain $\hat{\mathbf{x}}$, then $\hat{\mathbf{x}}$ should follow the nominal density f_0. Similarly, if the corruptions in $\mathbf{x}$ can be localized, then the corresponding portions would follow the multivariate uniform density $U_Z(\mathbf{z})$ of the appropriate dimensionality. On the other hand, this corruption model potentially creates significant statistical deviations from the reference data since a corrupted observation $\mathbf{x} \sim f_1$ and f_1, in general, increasingly diverges from f_0 as the corruption strength increases. Here, the corruption strength can be considered as the number of corrupted attributes and/or the variance of the corruption $U_Z(z)$ that overwrites the true data. Furthermore, our modeling of corruptions poses a missing (incomplete) data problem since the unknown true attributes $\bar{\mathbf{x}}_c^{c+\beta-1}$ in a corrupted interval are statistically irrelevant to the corrupted attributes $\mathbf{x}_c^{c+\beta-1}$. In this paper, by exploiting the statistical deviations from the nominal distribution of observations, we aim to detect and localize the possible corruptions in a given instance $\mathbf{x}$ and impute the corrupted or missing attributes.

To this end, we formulate an anomaly detection approach to define this framework in Section III, where we draw the distinctions among several examples of anomalous observations and separate the event of corruption. Then, we propose our algorithm and analyze the associated false alarm probability in detecting corruptions as well as the computational complexity.

III. NOVEL FRAMEWORK FOR CORRUPTION DETECTION, LOCALIZATION, AND IMPUTATION

In this section, we develop a novel framework for a complete treatment of possible corruptions in the input data $\mathbf{x}$. For presentational clarity and without loss of generality, we assume that the input data $\mathbf{x}$ can be corrupted only in a single interval throughout this section. Note that the generalization to the case of corruptions spread onto several intervals is immediate and indeed, we present a corresponding detailed experiment in Section V. Since the corruptions are modeled as local statistical deviations within this framework, we give a brief description of the anomaly detection approach that we work with in Section III-A. Based on the characterization of corruptions through their distinctive properties in Section III-B, we present an algorithm named tree-based corruption separation (TCS). After we derive a novel MAP estimator for imputation in Section III-C, we derive the false alarm rate of the proposed framework in detecting the corruptions in Section III-D.

A. Detection of Statistical Deviations: Anomalies

A localized corruption is considered to affect an instance in a certain part(s) such that the affected attributes statistically deviate from the vast majority of the data. The proposed algorithm in this paper localizes the corrupted attributes by identifying the local anomalies through a series of statistical checks of the test instance with the reference data. In this section, we briefly describe the anomaly detection approach that we work with and present a novel distance measure for the corruption localization purpose.
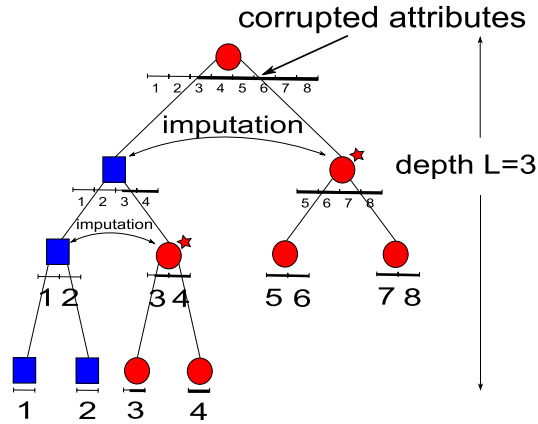
Fig. 1. Algorithm TCS with $\alpha = 0.5$.

The probability density of a possibly corrupted test instance $\mathbf{x}$ can be modeled as

$$\mathbf{x} \sim (1 - \pi) f_0(\mathbf{x}) + \pi f_1(\mathbf{x})$$

where $H_0 : \mathbf{x} \sim f_0(\mathbf{x})$ is the null hypothesis from which the nominal data are drawn, $H_1 : \mathbf{x} \sim f_1(\mathbf{x})$ is the hypothesis representing the corrupted observations, and $\pi \in [0, 1]$ is the corresponding mixing coefficient. Within the framework of anomaly detection approaches, the nominal distribution f_0 is usually assumed unknown or hard to estimate, and instead, a set of nominal observations is provided. Then for a given test instance $\mathbf{x}$, the task in [8] is to decide whether the null hypothesis H_0 was realized or the alternative H_1 such that the detection rate (of anomalies) is maximized with a constant false alarm rate τ. For this purpose, the score function [8]

$$\hat{p}_K(\mathbf{x}) = \frac{1}{N_s} \sum_{i=1}^{N_s} \mathbf{1}_{\{R_S(\mathbf{x};K) \le R_S(\mathbf{s}_i;K)\}} \qquad (1)$$

is proposed, where $\mathbf{1}_{\{.\}}$ is the indicator function and $R_S(\mathbf{x}; K)$ is the Euclidean distance from $\mathbf{x}$ to its nearest Kth neighbor in S, if $\mathbf{x} \notin S$, and to its nearest $(K + 1)$th neighbor in S otherwise. Based on this score function, the test instance $\mathbf{x}$ is declared as anomalous [8], if

$$\hat{p}_K(\mathbf{x}) \le \tau. \qquad (2)$$

When the mixing distribution f_1 is assumed uniform, it is shown in [8] that $\hat{p}_K(\mathbf{x})$ is an asymptotically consistent estimator of the density level of the test instance

$$p(\mathbf{x}) = \int_{\forall \mathbf{s}} \mathbf{1}_{\{f_0(\mathbf{x}) \ge f_0(\mathbf{s})\}} f_0(\mathbf{s}) d\mathbf{s} \qquad (3)$$

under certain smoothness conditions. Remarkably, $\{x : p(x) \ge \tau\}$ provides the minimum volume set at level τ, which is the most powerful decision region for testing H_0 versus H_1 with a constant false alarm rate τ [7]. We note that the precision of the test defined in (2) degrades faster with the dimensionality than it improves with the size of the training data. As a result, we here point out several practical issues about detecting the existence of a corruption with this approach.

Briefly, the conditions are described as follows.

1) A direct test of an instance $\mathbf{x}$ does not localize a possible corruption for imputation.

2) On the contrary, a truly corrupted instance, i.e., an instance of hypothesis H_1, does not necessarily test positive due to the limited training data, high dimensionality, as well as that the corruption might not be sufficiently strong.

3) Corruptions have further specific properties in addition to that they provide anomalies, which must be incorporated to achieve a better false alarm rate compared with τ.

1) Ranked Euclidean Distances: To address the first issue in this list, we propose a novel distance measure (not a metric in the mathematical sense), which is sensitive to only a certain α fraction of the attributes for a given pair of instances $\mathbf{x}$ and $\mathbf{y}$. For instance, a corruption of only a single attribute in a given test instance $\mathbf{x}$ might be significantly strong such that the whole instance turns anomalous with the test in (2) used with the standard Euclidean distance. In this case, any part of the instance $\mathbf{x}$ including the corrupted attribute would test positive, which creates an ambiguity in terms of the localization, i.e., separation, of the corrupted attribute, and in turn requires an exhaustive search over all possible subsets in the space of the attributes.

To overcome such ambiguities, we propose a distance measure so that the test in (2) results positive only when the corruption has a sufficiently large support, which disregards a prespecified fraction of the attributes that are most responsible for a possible corruption. We define this measure for an $\alpha \in [0, 1]$ as

$$h_\alpha(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{\lfloor d\alpha \rfloor} (x_{k(i)} - y_{k(i)})^2} \qquad (4)$$

where k is a permutation of the attributes with

$$|x_{k(1)} - y_{k(1)}| \le \cdots \le |x_{k(i)} - y_{k(i)}| \le \cdots \le |x_{k(d)} - y_{k(d)}|$$

and $\lfloor . \rfloor$ is the floor operator. Since this distance measure depends only on the α fraction of the least deviated attributes between $\mathbf{x}$ and $\mathbf{y}$, a corruption must have a support of at least $(d - \lfloor d\alpha \rfloor)$ length to make an instance anomalous with respect to the reference data. Here, $(1-\alpha)$ can be seen as the precision of the localization when an anomalous instance is checked with the test in (2) using the distance measure defined in (4). This precision obviously cannot be made arbitrarily large since as $1 - \alpha$ approaches 1, the distance h_α becomes more prone to noise and the correlation structure between the attributes is less exploited. We investigate this tradeoff further in our simulations. The distance measure h_α recovers the standard Euclidean distance when $\alpha = 1$ and will be named in the rest of this paper the ranked Euclidean distance. We note that for the cases $\alpha < 1$, h_α fails to be a metric in the mathematical sense, i.e., $h_\alpha(x, y) = 0 \Leftrightarrow x = y$ is not satisfied, which requires to specify a nominal density model on f_0 to derive the same asymptotic consistency in [8] for the score values $\hat{p}_K(x)$ in estimating the density levels $p(x)$ with h_α. However, we do not assume—in this paper—any density model for f_0 or do not take any stochastic assumptions regarding the data source.

In the following section, we characterize the corruptions by presenting their specific properties and propose an algorithm to localize and impute corruptions.
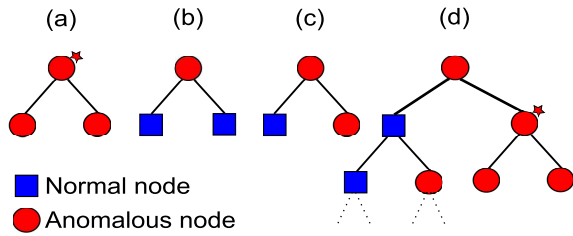
Fig. 2. Anomalous observation with several scenarios in its parts. Note that the starred nodes indicate localized corruptions. (a) Conclusive pattern: corruption is detected. (b) Conclusive pattern: corruption is rejected. (c) Inconclusive pattern. (d) Further exploration of the test instance.

B. Modeling of Localized Corruptions

If a test instance is subject to corruption in a small part only, the corruption might not be detectable when it is checked using an anomaly detection algorithm without a detailed analysis in its parts. On the other hand, an anomalous observation does not necessarily contain a corruption since it might be simply a false alarm, in fact an uncorrupted observation. To address these two issues, we propose a statistical analysis of a test instance through its parts using a binary partitioning tree in the space of data attributes on which we also provide a characterization to separate the event of corruption among possible anomaly scenarios.

Suppose that an instance $\mathbf{x} = [x_1, x_2, \ldots, x_d] \in \mathbb{R}^d$ corresponds to the root node R on a binary tree. Using halfway splits for presentational simplicity, let the set of attributes $V_{R_l} = \{x_1, x_2, \ldots, x_{\lfloor d/2 \rfloor}\}$ be assigned to the left child node R_l of the root and $V_{R_r} = \{x_{\lfloor d/2 \rfloor+1}, x_{\lfloor d/2 \rfloor+2}, \ldots, x_d\}$ assigned to the right child node R_r (Fig. 1). Note that $V_R = \{x_1, x_2, \ldots, x_d\}$ with $V_{R_l} \cap V_{R_r} = \emptyset$ and $V_R = V_{R_l} \cup V_{R_r}$. Based on this strategy for generating subparts of an instance, we propose Algorithm TCS to separate and impute corruptions, which recursively expands a depth-L binary tree to partition the space of attributes. For each node ν created in the course of this expansion, the corresponding attributes/part of the test instance, e.g., $\mathbf{x}_{V_{R_l}} := \mathbf{x}_1^{\lfloor d/2 \rfloor}$ with $\nu = R_l$, is checked whether it is consistent with the reference data restricted to those attributes, e.g., $S_{V_{R_l}} = \{s_1{}_1^{\lfloor d/2 \rfloor}, s_2{}_1^{\lfloor d/2 \rfloor}, \ldots, s_{N_s}{}_1^{\lfloor d/2 \rfloor}\}$ with $\nu = R_l$, using the test defined in (2). We here use the ranked Euclidean distance h_α in this testing with a prespecified α. Therefore, each node ν encountered in this expansion is assigned a binary label as anomalous/normal and a fully labeled (possibly unbalanced) tree is obtained for the test instance $\mathbf{x}$. We emphasize that Algorithm TCS does not completely construct this depth L-binary tree at the beginning, but instead expands it by creating the nodes and the edges as needed to achieve an efficient implementation, which continues until that each data attribute is decided to be corrupted or uncorrupted.

We consider several scenarios where the observation $\mathbf{x}_{V_\nu}$ at a node ν can be anomalous. In Fig. 2, the nodes are shown as circles if the corresponding part is found to be anomalous and squares otherwise. An anomaly can be wide spread onto the attributes and consist of anomalous subparts, as shown in Fig. 2(a). Since all of the subparts of a corrupted data part are also corrupted by definition, the pattern in Fig. 2(a) is regarded as a conclusive pattern. Hence, a corruption at the

starred node in Fig. 2(a) is declared, unless it is the root node. Note that a global corruption at the root is disregarded in this paper since it is not localized. In another case, an anomalous observation could be nonanomalous in its parts, as shown in Fig. 2(b), which simply happens due to an incompatible or rare combination of attributes in its subparts. This is a typical situation, where an anomalous observation is not corrupted. Hence, this case also provides a conclusive pattern in our consideration such that a corruption is rejected at the anomalous node. On the contrary, the case in Fig. 2(c) is an inconclusive pattern that suggests a corruption at the right child, however, whether the corruption is spread in the attributes of that child or localized is unknown. Hence, the attributes of the right child is further split and explored similarly. Then, if the conclusive pattern in Fig. 2(a) [or Fig. 2(b)] is realized, then the corruption is accepted and localized (or rejected) at the starred node in Fig. 2(d). Otherwise, the search continues. On the other hand, if a significantly small subset of the corrupted attributes are left at the left child node in Fig. 2(c), it might not be detectable and labeled as normal. Then the corresponding attributes should further be split, as shown in Fig. 2(d). This process recursively defines a corruption localization with an improved false alarm rate as several anomalies are rejected as they are false alarms, i.e., noncorrupted anomalies.

The introduced Algorithm TCS then searches the described binary tree in a breadth-first-search fashion for a corruption. When the conclusive (or terminating) pattern shown in Fig. 2(a) [Fig. 2(b)] is found in the course of this expansion, the search is stopped at the parent node of the found pattern, i.e., the tree is pruned on that branch, and corruption is declared (or no corruption is found and no action is necessary) for the corresponding attributes. This search of corruption at each branch starting from the root node continues to the corresponding leaf node unless a terminating pattern is found. Finally, if a conclusive pattern is not encountered at a branch from the root to an anomalous leaf, we opt to accept the corruption at the leaf to favor a better detection at a cost of an increased corruption false alarm rate. An illustration of the progress of the algorithm is given in Fig. 1, where the corrupted attributes are successfully located. Note that a small set of the attributes are mislabeled as corrupted, i.e., false alarms in the region 3, which can be corrected if the partitioning resolution is improved by increasing the depth L.

C. Maximum A Posteriori (MAP)-Based Imputation

We emphasize that in most of the detection and estimation applications, the posterior density, e.g., $f_0(\bar{\mathbf{x}}_{V_\nu}|\mathbf{x})$ in (5), of the target is too complicated to assume realistic parametric models so that the nonparametric approaches are often favored in such situations [43]. In accordance, we introduce an algorithm that works under a completely model free setting regarding both the localization of the corruptions and the imputation. Furthermore, we point out that when the posterior density is multimodal, MAP-based estimators are generally known to generate more plausible results compared with mmse-based estimators or simple (possibly weighted) averaging [44], which can even generate infeasible solutions [45]–[47]. This is often the case especially for the computer vision and

machine learning applications such as edge preserving image denoising [48]. For instance, the gradients in an occluded pedestrian image would get too smoothed in an MMSE-based imputation, which might cause the gradient-based feature extractors, e.g., HOG [49], to fail in the case of a pedestrian detection application [4], [43]. For these reasons, we propose a novel MAP-based imputation technique that always generates feasible and likely estimates and approximates the true MAP estimator as the size of the reference data increases.

Once a corruption is localized for an instance $\mathbf{x}$ at a node ν, then our task is to estimate the original attributes $\bar{\mathbf{x}}_{V_\nu}$ using the training data set S as well as the instance $\mathbf{x}$ and impute accordingly, i.e., replace the corrupted attributes in $\mathbf{x}$ with the estimates. Since we assume the corrupted attributes $\mathbf{x}_{V_\nu}$ to be statistically independent with the underlying true data $\bar{\mathbf{x}}_{V_\nu}$, we treat the corrupted attributes as the missing data, which then should have no effect in the estimation of the true attributes. Hence, we condition this estimation of the data $\bar{\mathbf{x}}_{V_\nu}$ on the remaining attributes in $\mathbf{x}$. On the other hand, we note that in most of the applications such as the image compression [50], the data attributes being in sufficiently close proximity are usually modeled to manifest high correlation. In accordance, we propose to estimate the unknown data $\bar{\mathbf{x}}_{V_\nu}$ conditioned on the attributes $\mathbf{x}_{V_{\nu_s}}$ associated with its nearest neighbor (NN) on our tree, i.e., the sibling node ν_s of ν. Note that due to the localization of corruptions by Algorithm TCS, the attributes at the sibling node ν_s are certainly detected to be uncorrupted in the case of the standard Euclidean distance; and detected to be uncorrupted with significantly high probability in the case of the ranked Euclidean distance (Section III-D). In the following, we introduce a novel (MAP) estimator of the true data underlying the corrupted attributes based on the standard Euclidean distance (h_α with $\alpha = 1$) and then discuss the generalization over α for the ranked Euclidean distance measure. We also stress that the implementation of this estimator is only based on the outputs of our corruption localization algorithm, which are computed before the imputation phase in the course of Algorithm TCS. Therefore, computationally, the imputation phase that we develop is efficient such that it does not require further computations.

Since the only relevant part of the test instance $\mathbf{x}$ to the proposed MAP estimator is $\mathbf{x}_{V_{\nu_s}}$, we have

$$f_0(\bar{\mathbf{x}}_{V_\nu}|\mathbf{x}) = f_0(\bar{\mathbf{x}}_{V_\nu}|\mathbf{x}_{V_{\nu_s}}) \tag{5}$$

where $\bar{\mathbf{x}}_{V_\nu}$ represents a realization of the conditional probability density of the true data underlying the corrupted attributes V_ν. Then the MAP estimator of $\bar{\mathbf{x}}_{V_\nu}$ maximizes the posterior distribution as

$$\mathbf{x}_{V_\nu}^{\mathrm{MAP}} = \arg \sup_{\bar{\mathbf{x}}_{V_\nu} \in \mathbb{R}^{|V_\nu|}} f_0(\bar{\mathbf{x}}_{V_\nu}|\mathbf{x}_{V_{\nu_s}}).$$

For any $\epsilon > 0$ and under certain smoothness constraints on f_0 with $f_0(\bar{\mathbf{x}}_{V_\nu}) \neq 0$, let

$$B_\epsilon(\bar{\mathbf{x}}_{V_\nu}) \cap S_{V_\nu} \neq \emptyset$$

hold with some probability δ_{N_s}, where $B_\epsilon(\bar{\mathbf{x}}_{V_\nu})$ (with respect to the standard Euclidean distance) is the ϵ-ball around $\bar{\mathbf{x}}_{V_\nu}$ in $\mathbb{R}^{|V_\nu|}$ and $N_s = |S|$. Then we point out that

$$\lim_{N_s \to \infty} \delta_{N_s} = 1.$$

Algorithm 1 Algorithm TCS Tree-Based Corruption Separation

Input: $\alpha, K, \tau, L;\ S, \mathbf{x}$
1: Initialize $\mathcal{C} \leftarrow \emptyset$: set of corrupted attributes
2: Initialize $\mathbf{y} \leftarrow \mathbf{x}$: imputed test data
3: Create the root node $\nu \leftarrow R$ and label
4: **procedure** RECURSE(ν)
5:     Create nodes ν_l and ν_r; and label
6:     **if** the pattern in Fig. 2a **then**
7:         **if** ν is the root **then return**
8:         **else**
9:             Declare corruption at ν: $\mathcal{C} \leftarrow \mathcal{C} \cup V_\nu$
10:            Impute attributes V_ν in $\mathbf{y}$
11:            **return**
12:         **end if**
13:     **else if** the pattern in Fig. 2b **then return**
14:     **else if** ν is a parent of a leaf **then**
15:         **if** ν_j ($j = l$ or $j = r$) is anomalous **then**
16:            Declare corruption at ν_j: $\mathcal{C} \leftarrow \mathcal{C} \cup V_{\nu_j}$
17:            Impute attributes V_{ν_j} in $\mathbf{y}$
18:         **end if**
19:         **return**
20:     **else**
21:         RECURSE(ν_l) and RECURSE(ν_r)
22:     **end if**
23: **end procedure**
Return: $\mathcal{C}$ and $\mathbf{y}$

Hence, since ϵ can be made arbitrarily small, we obtain

$$\mathbf{x}_{V_\nu}^{\mathrm{MAP}} = \arg \lim_{N_s \to \infty} \sup_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}} f_0(\bar{\mathbf{x}}_{V_\nu}|\mathbf{x}_{V_{\nu_s}})$$

and by the Baye rule

$$\begin{aligned}
\mathbf{x}_{V_\nu}^{\mathrm{MAP}} &= \arg \lim_{N_s \to \infty} \sup_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}} \frac{f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{x}_{V_{\nu_s}})}{f_0(\mathbf{x}_{V_{\nu_s}})} \\
&= \arg \lim_{N_s \to \infty} \sup_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}} f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{x}_{V_{\nu_s}}) \tag{6}
\end{aligned}$$

with probability 1, where the denominator is dropped since it does not depend on the maximizer, i.e., $\bar{\mathbf{x}}_{V_\nu}$. To approximate the MAP estimator given in (6), we adapt the nonparametric k-nn (knn) based density estimation approach [51]. Let us define a small neighborhood around $\mathbf{x}_{V_{\nu_s}}$ in $\mathbb{R}^{|V_{\nu_s}|}$ as

$$\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}}) = \left\{ \mathbf{s} : R_S(x_{V_{\nu_s}}; \gamma\sqrt{N_s}) \geq h_{\alpha=1}(\mathbf{x}_{V_{\nu_s}}, \mathbf{s}) \right\} \tag{7}$$

where $h_{\alpha=1}(.,.)$ is the Euclidean distance and $R_S(x_{V_{\nu_s}}; \gamma\sqrt{N_s})$ is the $h_{\alpha=1}(.,.)$ distance from $x_{V_{\nu_s}}$ to its nearest $\gamma\sqrt{N_s}$th neighbor in $S_{V_{\nu_s}}$ for some $\gamma > 0$. Note that as $N_s \to \infty$, $\mathcal{L}(\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})) \to 0$, where $\mathcal{L}(.)$ is the Lebesgue measure. Then (6) yields

$$\mathbf{x}_{V_\nu}^{\mathrm{MAP}} = \arg \lim_{N_s \to \infty} \sup_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}} \frac{\int_{\mathbf{z} \in \mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})} f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})\mathbf{dz}}{\mathcal{L}(\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}}))} \tag{8}$$

with probability 1. When N_s is sufficiently large with $N_s \geq N_s^*$ for some N_s^* or $\mathcal{L}(\mathcal{N}_{N_s})$ is sufficiently small, we assume that $f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{x}_{V_{\nu_s}})$ is subject to negligible

variations only. Then we (with probability 1) obtain the approximation

$$
\begin{aligned}
\mathbf{x}_{V_\nu}^{\mathrm{MAP}} &= \arg \lim_{N_s \to \infty} \sup_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}} \frac{\int_{\mathbf{z} \in \mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})} f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z}) \mathbf{dz}}{\mathcal{L}(\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}}))} \\
&\simeq \arg \max_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}, \mathbf{z} \in \mathcal{N}_{N_s^*}(\mathbf{x}_{V_{\nu_s}})} f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})
\end{aligned} \tag{9}
$$

where to obtain the corresponding maximum in the reference set S, knowing the rank statistics in $f_0(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})$ is enough, i.e., explicitly estimating/computing the density is unnecessary. Therefore, using the density function defined in (3), we obtain

$$
\mathbf{x}_{V_\nu}^{\mathrm{MAP}} \simeq \arg \max_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}, \mathbf{z} \in \mathcal{N}_{N_s^*}(\mathbf{x}_{V_{\nu_s}})} p(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z}) \tag{10}
$$

For sufficiently large N_s, note that $\hat{p}_K(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})$ approximates $p(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})$ [8], i.e., $\forall(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})$

$$
|\hat{p}_K(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z}) - p(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z})| \simeq 0 \text{ almost surely.} \tag{11}
$$

Using the result in (10) in combination with (11), we propose to use MAP-based estimator of the true data underlying the corrupted attributes

$$
\mathbf{x}_{V_\nu}^{\mathrm{MAP}} \simeq \hat{\mathbf{x}}_{V_\nu} = \arg \max_{\bar{\mathbf{x}}_{V_\nu} \in S_{V_\nu}, \mathbf{z} \in \mathcal{N}_{N_s^*}(\mathbf{x}_{V_{\nu_s}})} \hat{p}_K(\bar{\mathbf{x}}_{V_\nu}, \mathbf{z}) \tag{12}
$$

based on which we replace, i.e., impute, the corrupted attributes $\mathbf{x}_{V_\nu}$ in the instance $\mathbf{x}$ with $\hat{\mathbf{x}}_{V_\nu}$ and obtain the imputed data as $\mathbf{y}$.

This estimator is implemented in Algorithm TCS at every node in the tree, where a corruption is detected. For example, the following have to be performed.

1) Obtain the K neighbors of the test instance in the reference data set S with respect to the attributes associated with the node ν_s.
2) For those neighbors in S, find the one, say s^*, attaining the largest score value defined in (1) using the attributes associated with the parent node ν_p.
3) Then impute the instance $\mathbf{x}$, which is detected to be corrupted at the node ν, using s^* for the attributes V_ν.

In the realistic case of high-dimensional and limited data, when the standard Euclidean distance is used as in our derivations, $\mathbf{x}_{V_{\nu_s}}$ might include corrupted attributes even though it is detected as normal, which clearly adversely affects the calculation of the neighborhood $\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})$ in (7). In addition, $\mathbf{x}_{V_\nu}$ might only include a small support of corruption, and then we would not like to impute $\mathbf{x}_{V_\nu}$ completely. To overcome these two issues, we propose to use the ranked Euclidean distance defined in (4). To this end, the neighborhood $\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})$ is defined using h_α with an appropriate $\alpha \neq 1$ in (7). This cancels the adverse effect, up to a certain degree, of a possible corruption in $\mathbf{x}_{V_{\nu_s}}$ as desired. Nevertheless, recalling that h_α only uses the α fraction of the attributes V_{ν_s} and set the others free, h_α is not a metric in the mathematical sense and then as $N_s \to \infty$, $\mathcal{L}(\mathcal{N}_{N_s}(\mathbf{x}_{V_{\nu_s}})) \to 0$ does not hold. As a result, the correlation structure given in (5) is less exploited in imputation as α decreases. Meanwhile, as α decreases, the support of the detected corruption in x_{V_ν} increases, i.e., localization improves. Therefore, we obviously have a tradeoff between the imputation quality and the localization, which is

sensitive to the choice of α and investigated in the experiments in greater detail. However, α should be set typically around 0.5–0.75 since we use half-way splits. Finally, note that the imputation brings almost no further computational complexity, since these steps do computationally depend only on the anomaly detection results (1) and (2) at the corrupted node, its sibling node, as well as its parent node, which are all generated prior to the imputation steps.

In the following section, the proposed framework is shown to achieve a constant false alarm rate in terms of the corruption detection. Moreover, this false alarm rate is precisely calculated under a certain dependency structure among the anomalous/normal labels on the partitioning tree.

D. False Alarm Rate in Detecting Corruptions

Since the imputation is an overwriting operation, whether or not to impute a suspicious instance is certainly a critical decision. In case of a false decision, if the suspicious instance is in fact uncorrupted, i.e., a false alarm in detecting corruptions, the imputation would correspond to data loss. In this section, we study the rate of such occurrences and analyze the false alarm rate of the proposed algorithms in detecting corruptions.

The anomaly detection test applied at every node in Algorithm TCS operates with a constant false alarm rate τ, whereas the proposed approach is able to reject corruptions at anomalous nodes. For example, when the terminating pattern in Fig. 2(b) is encountered, all the anomalies that can be present in the tree rooted from the terminating pattern are rejected, i.e., they are not counted as corruptions. For this reason, the false alarm rate of the proposed approach must be defined in the sense of corruptions as opposed to anomalies. To analyze this false alarm rate in detecting corruptions, one also must account for the fact that the anomaly detection test at a node could be strongly correlated with the outputs of the previous tests in the course of Algorithm TCS, since the data attributes are in general correlated. In this section, we first model the labeling of the nodes, i.e., anomalous versus normal, on the partitioning tree (Fig. 1), as a directed acyclic graph [52] achieving a certain dependency structure and then derive the false alarm rate of Algorithm TCS. Under this modeling, we also show that the constant false alarm rate in detecting the local anomalies at each node also globally maps to a constant false alarm rate in detecting the corruptions.

Recall that Algorithm TCS expands the binary tree in Fig. 1 for a given uncorrupted test instance $\mathbf{s}$ and declares a corruption only if the conclusive pattern in Fig. 2(a) is encountered or a leaf node is found anomalous in the described breadth-first search. In addition to the corruption localization as well as the imputation capabilities of the proposed Algorithm TCS, let us denote the corruption detection in Algorithm TCS by $\mathcal{C}(\mathbf{s}) = 1$, if $\mathbf{s}$ is detected to be corrupted and $\mathcal{C}(\mathbf{s}) = 0$ otherwise. Then our task is to find the false alarm probability in detecting the corruptions, which is given by

$$
C_\tau = \int_{\forall \mathbf{s}} \mathcal{C}(\mathbf{s}) f_0(\mathbf{s}) \mathbf{ds} \tag{13}
$$

where τ is the constant false alarm rate of the detection at each node and f_0 is the nominal density. Next, we observe that

Algorithm TCS maps every data instance to a binary observation such that the nominal distribution f_0 is transformed into a multivariate Bernoulli distribution p_0

$$\mathbb{R}^d \rightarrow B^{2^{L+1}-1} \text{ via}$$

$$\mathbf{s} \rightarrow \mathcal{L}(\mathbf{s}) = \mathbf{u} = (u_R, u_{R_l}, u_{R_r}, u_{R_{ll}}, u_{R_{lr}}, u_{R_{rl}}, u_{R_{rr}}, \ldots)$$

where $B = \{-1, 1\}$, L is the depth, and u_R is the anomaly decision at the root node such that $u_R = 1$, if an anomaly detected and $u_R = -1$ otherwise. Similarly for the others such as u_{R_l} is the decision at the left hand child of the root and u_{R_r} is the decision at the right hand child. Note that the proposed algorithm does not completely construct the binary tree but expands, i.e., the nodes and the edges are created as needed. Therefore, we do not completely observe the binary vector $\mathbf{u}$ that an instance $\mathbf{s}$ maps to, however, we temporarily suppose that all the labels are available for ease of exposition. Once $\mathbf{s}$ is mapped to $\mathbf{u}$, since Algorithm TCS declares a corruption based on only the vector of binary labels $\mathbf{u}$, we equivalently have

$$C_\tau = P\left(\mathcal{C}(\mathbf{s}) = 1 \mid \mathbf{s}, \text{ in fact, is uncorrupted}\right)$$

$$= \sum_{\mathbf{u} \in \{-1,1\}^{2^{L+1}-1}} \mathcal{C}(\mathbf{u}) p_0(\mathbf{u})$$

$$= 1 - \sum_{\mathbf{u} \in \{-1,1\}^{2^{L+1}-1}} \mathcal{C}^c(\mathbf{u}) p_0(\mathbf{u}) \qquad (14)$$

where $\mathcal{C}(\mathbf{u})$ is the corruption decision (with abuse of notation), $\mathcal{C}^c(\mathbf{u})$ is the complement, i.e., $\mathcal{C}^c(\mathbf{u}) = 1 - \mathcal{C}(\mathbf{u})$, and p_0 is the corresponding nominal probability mass function such that

$$p_0(\mathbf{u}) = \int_{\forall \mathbf{s}: \mathcal{L}(\mathbf{s})=\mathbf{u}} f_0(\mathbf{s}) \, d\mathbf{s}.$$

To calculate the probability mass function p_0, we model the binary tree, where each node corresponds to a binary random variable, as a directed acyclic graph [52] such that the binary random variables at any two sibling nodes are independently conditioned on the knowledge of the label at the parent node. For any non leaf node v and its children v_l and v_r on the binary partitioning tree, we assume the following conditional independency for the associated random labels: $p_0(u_{v_l}, u_{v_r} \mid u_v) = p_0(u_{v_l} \mid u_v) p_0(u_{v_r} \mid u_v)$, from which we obtain (Fig. 3):

$$p_0(u_v, u_{v_l}, u_{v_r}) = p_0(u_{v_l}, u_{v_r} \mid u_v) p_0(u_v)$$

$$= p_0(u_{v_l} \mid u_v) p_0(u_{v_r} \mid u_v) p_0(u_v). \qquad (15)$$

Here, we emphasize that $\mathbf{s}$ (or $\mathbf{u}$) is assumed to be uncorrupted in the false alarm analysis to calculate the probability given in (13), i.e., it does not have any localized corruptions by definition. Then, without loss of generality, if $\mathbf{s}$ is declared as anomalous at the root node, then this anomaly is not due to a corruption but simply a rarity as the test in (2) is based on density levels. On the contrary to the case of corruption, since a rarity at a node is not a localized phenomenon, we expect that the children inherit the parent label independently. Therefore, we assumed the conditional independence in (15) as a generating dependency structure for the simplest graph presented in Fig. 3, which straightforwardly
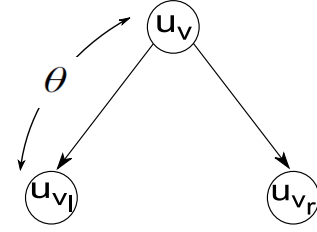

Fig. 3. Assuming the conditional independency: $p_0(u_v, u_{v_l}, u_{v_r}) = p_0(u_{v_l} \mid u_v) p_0(u_{v_r} \mid u_v) p_0(u_v)$. Moreover, $p_0(u_{v_l} \mid u_v) = (1 - \theta) p_0(u_{v_l}) + \theta 1_{\{u_{v_l}=u_v\}}$, where θ defines the dependency between the parent node and its siblings such that a positive covariance is embedded. Note that $\theta = 0$ implies independency.

generalizes to the binary tree of the anomalous versus normal labels from root to the leaves. Based on this, we obtain

$$p_0(\mathbf{u}) = p_0(\mathbf{u}_R \mid u_R)^* p_0(u_R)$$

$$= p_0(\mathbf{u}_{R_l} \mathbf{u}_{R_r} \mid u_R, u_{R_l}, u_{R_r}) p_0(u_{R_l}, u_{R_r} \mid u_R) p_0(u_R)$$

$$= p_0(\mathbf{u}_{R_l} \mid u_{R_l})^* p_0(\mathbf{u}_{R_r} \mid u_{R_r})^* p_0(u_{R_l} \mid u_R)$$

$$\times p_0(u_{R_r} \mid u_R) p_0(u_R) \qquad (16)$$

where $\mathbf{u}_R$ is the collection of the binary variables associated with the nodes in the tree rooted from node R that excludes u_R, and the last equation follows from (15) and the Bayes rule. We observe that the starred factors in the expression (16) are of similar forms such that the last equation can be expanded further using similar lines of derivations up until the leaves appear.

Thus, the calculation of $p_0(\mathbf{u})$ requires the calculation of the probabilities of the form $p_0(u_{v_l} \mid u_v)$ or $p_0(u_{v_r} \mid u_v)$, e.g., $p_0(u_{R_r} \mid u_R)$ in (16). Let us denote any child of the node v by v_s for generalization. Note that if u_v and u_{v_s} were independent, then we would have $p_0(u_{v_s} \mid u_v) = p_0(u_{v_s}) = \tau$ when $u_{v_s} = 1$. However, we anticipate a statistical dependency between u_v and u_{v_s} generating a positive covariance. That is, conditioned on the knowledge of u_v, we would like to impose that u_{v_s} is more likely to attain the value u_v compared with the prior conditions, i.e., v_s is likely to inherit the label of its parent. On the other hand, provided that u_v and u_{v_s} are identically dependent, we would have that $p_0(u_{v_s} \mid u_v) = 1_{\{u_v=u_{v_s}\}}$, where $1_{\{.\}}$ is the indicator function. To introduce this into the derivations, we parameterize the probability mass function $p_0(u_{v_s} \mid u_v)$ as the weighted average between $p_0(u_{v_s})$ and $1_{\{u_v=u_{v_s}\}}$ as

$$p_0(u_{v_s} \mid u_v) = (1 - \theta) p_0(u_{v_s}) + \theta 1_{\{u_v=u_{v_s}\}}$$

$$= (1 - \theta)(0.5 - u_{v_s}(0.5 - \tau)) + \theta \frac{1 + u_v u_{v_s}}{2} \qquad (17)$$

where $\theta \in [0, 1]$ is a parameter defining the degree of dependency, which generates an increasing covariance as θ increases in the interval $[0, 1]$ such that $\theta = 0$ implies the statistical independence of u_v and u_{v_s}; and $\theta = 1$ implies identical dependency. Then, the probability mass function $p_0(\mathbf{u})$ can be calculated using this parametrization based on the recursion in (16). Hence, exhaustively enumerating all possible $\mathbf{u}$s and running Algorithm TCS for each of them, one

can calculate the false alarm rate C_τ in (14), which is not a practical choice. Instead, through the conditional factorization in (15), we opt to simplify the expression (14) and obtain an efficient recursion. To this end, for a given node v with depth $1 \le i \le L - 2$, let us define the probability conditioned on u_v that Algorithm TCS does not declare a corruption in the tree rooted from v denoted by $\mathcal{F}(v; u_v)$ as

$$\mathcal{F}(v; u_v) = \sum_{\mathbf{u}_v \in B^{2^{L-i+1}-2}} \mathcal{C}^c((\mathbf{u}_v, u_v)) p_0(\mathbf{u}_v | u_v).$$

Here, $\mathcal{F}(v; u_v)$ solely depends on the depth variable i due to the symmetric factorization by the conditional independency from parents to children. Therefore, the notation simplifies to $\mathcal{F}(i; 1)$ or $\mathcal{F}(i; -1)$. Using the four possible configurations for $(u_v = 1, u_{v_l}, u_{v_r})$, we can calculate $\mathcal{F}(i; 1)$ as a function of $\mathcal{F}(i + 1; .)$. Noting that two of those configurations are the conclusive patterns, termination and corruption patterns, we obtain

$$\mathcal{F}(i; 1) = q_1^2(-1) + 2q_1(-1)q_1(1)\mathcal{F}(i + 1; 1)\mathcal{F}(i + 1; -1)$$

where $q_i(j) = p_0(v_s = j | v = i)$ as a short-hand notation, the second term corresponds to the continuation of Algorithm TCS, and the first term corresponds to the terminating pattern. Unlike the second term, the first term does not have a multiplier since the search stops at such a node. Note that the corruption pattern is disregarded by definition. Similarly, we also have

$$\mathcal{F}(i; -1) = q_{-1}^2(1)\mathcal{F}^2(i + 1; 1) + q_{-1}^2(-1)\mathcal{F}^2(i + 1; -1) + 2q_{-1}(1)q_{-1}(-1)\mathcal{F}(i + 1; 1)\mathcal{F}(i + 1; -1).$$

Recalling that we declare corruptions at leaf nodes on the basis of local anomalies, we can further define

$$\mathcal{F}(L - 1; 1) = q_1^2(-1) \quad \text{and} \quad \mathcal{F}(L - 1; -1) = q_{-1}^2(-1)$$

and provide the initialization to the recursion $\mathcal{F}(i; 1)$ and $\mathcal{F}(i; -1)$. On the other hand, we never declare corruptions at the root since we are focused only on localized corruptions, which is an exception and can be straightforwardly incorporated in our recursions. In terms of the recursions regarding $\mathcal{F}(i; 1)$, the only change is that the corruption pattern should not be disregarded, which does not lead to a corruption detection and so does not stop the search. Then, we simply have

$$\mathcal{F}(0; 1) = q_1^2(-1) + q_1^2(1)\mathcal{F}^2(1; 1) \\ + 2q_1(-1)q_1(1)\mathcal{F}(i + 1; 1)\mathcal{F}(i + 1; -1)$$

and the recursion $\mathcal{F}(i; -1)$ stays valid for $\mathcal{F}(0; -1)$. Now that we have the recursion equations defined for all depth levels on the binary tree, we can efficiently calculate the false alarm rate of Algorithm TCS as follows. Letting R represent the root
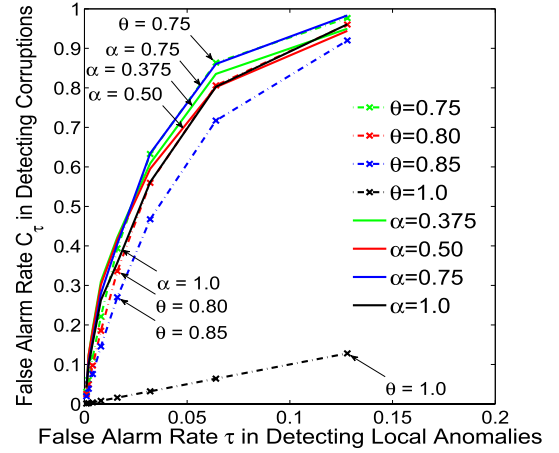

Fig. 4. Solid (dashed-dotted) curves correspond to the realizations (hypothetical results). The constant false alarm rate τ in detecting the local anomalies maps to a global constant false alarm rate C_τ in detecting the corruptions with algorithm TCS. We observe that setting $\theta \in [0.75, 0.8]$ well approximates the relation between τ and C_τ. In case of the identical dependency, i.e., $\theta = 1$ and $C_\tau = \tau$.

node, we obtain from (14)

$$1 - C_\tau = \sum_{\mathbf{u} \in B^{2^{L+1}-1}} \mathcal{C}^c(\mathbf{u}) p_0(\mathbf{u}) = p_0(u_R = 1)$$
$$\times \sum_{\mathbf{u}_R \in B^{2^{L+1}-2}} \mathcal{C}^c((u_R, \mathbf{u}_R)) p_0(\mathbf{u}_R | u_R = 1)$$
$$+ p_0(u_R = -1) \sum_{\mathbf{u}_R \in B^{2^{L+1}-2}} \mathcal{C}^c((u_R, \mathbf{u}_R))$$
$$\times p_0(\mathbf{u}_R | u_R = -1).$$

Then, recalling that $p_0(u_R = 1) = 1 - p_0(u_R = -1) = \tau$, the false alarm rate C_τ is given by

$$C_\tau = 1 - \tau\mathcal{F}(0; 1) - (1 - \tau)\mathcal{F}(0; -1) \qquad (18)$$

which is equivalent to first calculating the probability that Algorithm TCS never declares a corruption and then subtracting this probability from 1.

Since the false alarm rate C_τ of Algorithm TCS in detecting the corruptions as found in (18) is independent from the data, we conclude that the false alarm rate τ of the anomaly detection at each node maps to a constant false alarm probability of our corruption detection C_τ. Second, even though the dependency parameter θ does not appear, i.e., hidden, in (18), C_τ is clearly affected by θ. For example, if $\theta = 1$, i.e., if the binary label of a child node is identically dependent on the parent label and hence $p_0(u_{v_s} | u_v) = 1_{\{u_{v_s} = u_v\}}$, then it can be shown that $C_\tau = \tau$. If $\theta = 0$, i.e., if the binary label of a child node is independent with the parent label and hence $p_0(u_{v_s} | u_v) = p_0(u_{v_s})$, then obviously $C_\tau > \tau$. We experimentally discuss the quality of this hypothetical relation between τ and C_τ (Fig. 4) and the further details in Section V.

In the following section, we explain the important points of our implementation and discuss the corresponding computational complexity.

IV. COMPUTATIONAL COMPLEXITY

Computationally, the main building block in Algorithm TCS is the application of the anomaly test defined in (2), which computes the train-to-train distance matrix $D_S(i, j) = d(\mathbf{s}_i, \mathbf{s}_j)$ and the test-to-train distance vector $D_X(j) = d(\mathbf{x}, \mathbf{s}_j)$. Operating on these distances, the score function defined in (1) for the test instance must be computed, which then requires the computation and sorting of $R_S(\mathbf{s}_i; K)$. In addition, since we label each node as anomalous or not in our tree expansion, these distances must actually be computed at each node with respect to the corresponding attributes, e.g., $D_{S_{V_\nu}}(i, j)$ and $D_{X_{V_\nu}}(j)$ at a node ν. For this purpose, we adapt the integral image approach for the cases where the standard Euclidean distance is used. For example, let us define the volume $\mathcal{D}_S(i, j, k) = \sum_{h=1}^{k}(s_{ih} - s_{jh})^2, \forall i, j$ with $1 \leq k \leq d$; and $\mathcal{D}_S(i, j, 0) = 0, \forall i, j$ [similarly for $\mathcal{D}_X(j)$]. Then, we simply have $D_{S_{V_\nu}}(i, j) = (\mathcal{D}_S(i, j, k_2) - \mathcal{D}_S(i, j, k_1))^{1/2}$ at a node ν, where V_ν corresponds to the set of attributes in positions between $k_1 + 1$ and k_2. The volume $\mathcal{D}_S(i, j, k)$ and the sorting of $R_S(\mathbf{s}_i; K)$ can be computed offline once the training set is provided, which defines a training phase complexity $O(2^{L+1} N_s^2 \log_2 N_s)$, where sorting is the dominant contributor. For a given test instance, we compute $D_{X_{V_\nu}}(j)$ and sort at each node ν in the expansion of our tree, which defines the test phase complexity $O(2^{L+1} N_s \log_2 N_s)$ for our algorithm, where sorting is the dominant contributor. The computational load is multiplied by constant factors in the case of the ranked Euclidean distance.

V. EXPERIMENTS

In this section, we first discuss the efficacy of the false alarm rate estimation method explained in Section III-D and evaluate the performance of the critical steps in Algorithm TCS, which are the corruption detection, localization, and imputation. Then, we report the improvements achieved by the proposed framework in several classification tasks in comparison with a baseline of two state-of-the-art algorithms.

In the first set of experiments, we adapted a $0-1$ digit classification task consisting of a training set of 1500 samples and a test set of 750 samples based on the U.S. Postal Service (USPS) data [12]. Each of these samples is a 16×16 gray scale image of either a 0 image or a 1 image, where each pixel has a real intensity value in [0, 1]. We synthetically generate a corruption as described in Section II and apply to each instance in the test set with probability $\pi = (1/2)$. To be more precise, for a test instance chosen to be corrupted, we (uniformly) randomly specify a square region of size between $(10 - 50)\%$ of the total area, i.e., the number of pixels in the chosen region is not less than 25 and not more than 128, and overwrite each pixel in this region with a value randomly [using the uniform distribution $U_Z(z)$] drawn from the interval [0, 1]. Then, after the training and test instances are vectorized column wise such that $\mathbf{s}, \mathbf{x} \in \mathbb{R}^{256}$, the proposed Algorithm TCS is provided with the clean training data and run over the test set. We emphasize that by this vectorization scheme, the corrupted square region corresponds to multiple corrupted intervals in the vectorized observation. Hence, this example also illustrates that Algorithm TCS can handle multiple corruptions. Ideally, the neighborhood size
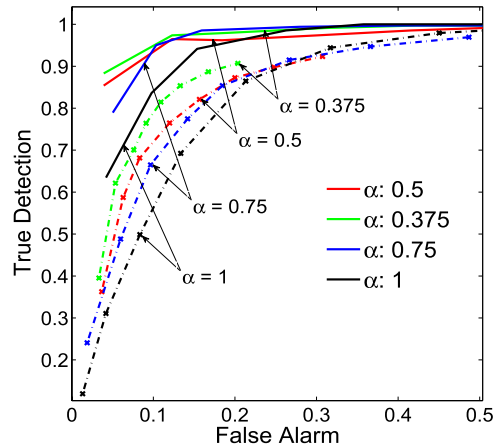

Fig. 5. Receiver operating characteristic curves for detection and localization of corruptions. Solid (dashed-dotted) curves correspond to detection (localization) performances.

parameter K for both imputation and corruption separation purposes should be optimized at every node of our binary tree since the data dimensionality from node to node varies. However, we opt not to optimize K for presentational clarity and set as $K = 8$ near the midpoint of [1, 16], which is empirically found appropriate. Using the $0 - 1$ digit USPS data, we investigate the response of the Algorithm TCS to the local anomaly detection false alarm rate $\tau \in \Gamma = \{0.001, 0.002, 0.004, 0.008, 0.016, 0.032, 0.064, 0.128\}$ and the ranked Euclidean distance parameter $\alpha \in \Delta = \{0.375, 0.5, 0.75, 1\}$. As for the depth parameter, we use the deepest possible tree with $L = 6$ such that the leaves are associated with four pixels and hence, one pixel at least is then used in the distance calculation with $\alpha = 0.375$.

In Fig. 4, we compare the hypothetical false alarm rate C_τ we derive in Section III-D with the corresponding experimental realizations with respect to varying local anomaly detection false alarm $\tau \in \Gamma$. The hypothetical map from τ to C_τ is generated with several choices for the dependency parameter $\theta \in \{0.75, 0.8, 0.85, 1\}$, whereas the realizations correspond to several choices for the distance parameter $\alpha \in \Delta$. Our experiments indicate that when the statistical dependency θ in (17) from a parent node to one of its children nodes (Fig. 3) is chosen around 0.75, the relationship between the local anomaly false alarm rate τ and the corruption detection false alarm rate C_τ is accurately modeled. This experimentally shows that the labeling of local anomalies over a binary partitioning tree shown in Fig. 1 can be considered as a directed acyclic graph. We also observe that in the case of Euclidean distance, i.e., h_α with $\alpha = 1$, while $\theta \sim 0.75$ is more accurate for small τs, θ tends to approach 0.8 as τ increases for a better modeling. This small deviation mainly happens since the conditional independency assumption explained in Fig. 3 does not hold in case of Euclidean distance for a certain pattern. Although the labeling for a parent and its children nodes as $-1, 1, 1$ (a normal parent node with anomalous children nodes) is not possible with the standard Euclidean distance due to the test defined in (2), the directed acyclic graph modeling assigns it a positive probability, which then overestimates the corruption detection false alarm rate. Nevertheless, this positive probability is the smallest among the ones assigned
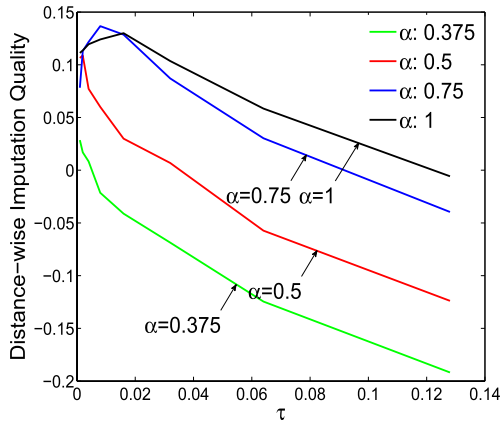
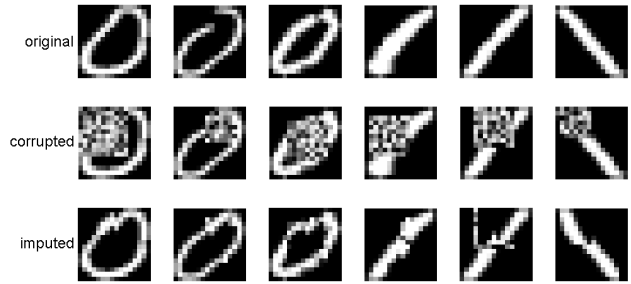Fig. 6.   Distance-wise imputation quality with $\tau \in \Gamma$.

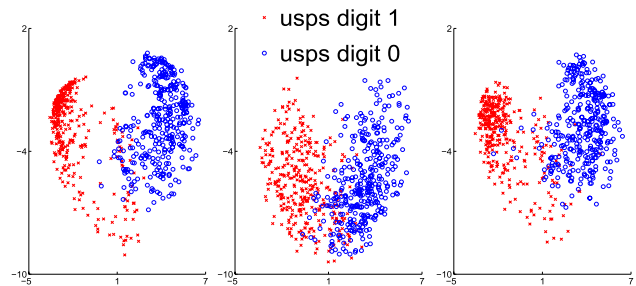Fig. 7.   Several visual examples on USPS data set.

Fig. 8.   Left: (uncorrupted) true data scatter; mean separation between two classes: 5.95 and linear SVM accuracy: 99.71%. Middle: corrupted data scatter; mean separation: 4.19, and classification accuracy: 90.57%. Right: imputed data scatter; mean separation: 5.37, and classification accuracy: 96.85%, which corresponds to ~68.0% improvement both in terms of the mean separation and classification.

to the all possible patterns of three nodes as desired and hence, the ordering of the patterns in terms of their probabilities is still reasonable even in the case of the Euclidean distance. On the contrary, since this pattern is also possible in the case of the ranked Euclidean distance, the accuracy of our hypothetical results improves as α decreases.

Next, we study the corruption detection and localization performance of our algorithm on the $0-1$ digit USPS data. In Fig. 5, we plot the empirical false alarm rates versus the empirical true detection rates in terms of both corruption detection and corruption localization with respect to $\tau \in \Gamma$. Here, the true detection rate is the empirical probability, i.e., relative frequency, of a truly corrupted data instance (data attribute in case of localization) to be declared corrupted, and the false alarm rate is the empirical probability of a truly uncorrupted data instance (data attribute in case of localization) to be declared corrupted. As we discuss in Section III, the ranked Euclidean distance is experimentally shown to produce a better detection as well as localization performance on the USPS data as α decreases. Recall that for a small α around 0.5, we enforce a corruption to be widely spread for Algorithm TCS to detect it at a node, which then clearly improves the localization. Similarly, the corruption detection performance also improves as α decreases. Since the ranked Euclidean distance disregards a certain fraction of the largest attribute-wise deviations, Algorithm TCS behaves conservatively in declaring corruptions. This reduces the false alarms in terms of the local anomalies and in turn, reduces the false encounters of the terminating pattern shown in Fig. 2(b). Hence, the corruption search is not stopped mistakenly, and Algorithm TCS does not miss certain corruptions, which leads to a better detection rate with the ranked Euclidean distance using a small α around 0.5. We emphasize that the local anomaly detection false alarm rate τ can be set independently for detection and localization to precisely determine the operating point on the Receiver Operating Characteristic curves in Fig. 5. However, in this paper, we use one single τ in all phases of Algorithm TCS. Note that when the false alarm rate is set around 0.1–0.2, our algorithm is able to provide a detection rate around 0.9 and a localization rate around 0.8.

On the other hand, the ranked Euclidean distance parameter α cannot be made arbitrarily small. Observe that with small α, only a small fraction of attributes is used in determination of $\mathcal{N}_{N_s}(\mathbf{x}_{V_{v_s}})$ in (7) despite that the rest of the

attributes might be informative through the local correlations and, hence, the imputation quality degrades. We illustrate this effect in Fig. 6, where we use the improvements in the distance wise deviations after imputation to measure the imputation quality. For this purpose, we define

$$\frac{1}{N_c} \sum_{i=1}^{N_c} \frac{h_{\alpha=1}(\bar{\mathbf{x}}_i, \mathbf{x}_i) - h_{\alpha=1}(\bar{\mathbf{x}}_i, \hat{\mathbf{x}}_i)}{h_{\alpha=1}(\bar{\mathbf{x}}_i, \mathbf{x}_i)} \qquad (19)$$

as the distance wise imputation quality, where N_c is the number of the corrupted test instance (which is approximately 750π), $\mathbf{x}_i$ is a corrupted test instance, $\bar{\mathbf{x}}_i$ is the uncorrupted original instance, and $\hat{\mathbf{x}}_i$ is the corresponding instance after imputation. Note that this quality metric measures on average how much of the distortion by the corruption is recovered after imputation. The average imputation quality defined in (19) is plotted versus the local anomaly detection false alarm $\tau \in \Gamma$ in Fig. 6. We first observe that for large τ, since the false alarm rate is also large, the imputation even further disturbs the data. Second, for small τ around 0.01, the proposed imputation technique is able to correct a corrupted instance up to 12% in case of $\alpha \sim 0.75$. Moreover, our experiments also indicate that for α less than 0.5, the ranked Euclidean distance is not able to produce desirable results despite its superiority in terms of detection and localization, which reinforces our discussion that α cannot be made too small.

Unlike an mmse-based approach, our MAP-based imputation does not target at minimizing the reconstruction error but the most likely replacement for a corruption. Indeed, an mmse-based estimator for imputation would produce visually blurry results, for instance, on the USPS data. In this regard, we present several visual examples that the proposed framework generates on the USPS data with

TABLE I

PROPOSED ALGORITHM TCS-MAP IS COMPARED WITH TWO BASELINE ALGORITHMS TCS-NN AND M-NN (CONSTRUCTED USING THE METHODS IN [11] AND [19]) IN TERMS OF THE CLASSIFICATION TASKS ON SEVERAL BENCHMARK DATA SETS. AVERAGE IMPROVEMENTS IN CLASSIFICATION ACCURACIES AFTER IMPUTATION ARE PRESENTED FOR ALL METHODS IN THE CASES OF CLEAN DATA TRAINING AND %5 CORRUPTED TRAINING

Data Sets (dimensionality) Avg. $\mathbf{acc}(\bar{\mathbf{X}})$ / $\mathbf{acc}(\mathbf{X})$	Uncorrupted Training Avg. % Improvements with its Std. below TCS-MAP / TCS-NN / 4-NN (16-NN)	%5 Corrupted Training Avg. % Improvements with its Std. below TCS-MAP / TCS-NN / 4-NN (16-NN)
Image (18) 84.62 / 57.16	55.60 / **59.14** / 42.80 (10.34) 1.81 / 2.58 / 3.20 (1.47)	**40.96** / 40.95 / 27.84 (9.34) 1.32 / 1.92 / 3.32 (1.79)
Ringnorm (21) 76.90 / 64.40	**46.85** / 46.15 / 31.99 (19.42) 4.23 / 3.27 / 5.40 (4.67)	**43.36** / 39.60 / 26.36 (18.35) 4.25 / 4.89 / 4.21 (5.62)
Twonorm (21) 96.46 / 85.84	31.16 / **31.36** / 14.75 (16.46) 3.51 / 1.91 / 2.09 (2.71)	**22.60** / 20.97 / 11.64 (12.51) 3.49 / 1.96 / 1.70 (3.55)
Svmguide3 (21) 81.08 / 68.74	62.33 / **68.86** / 60.57 (6.86) 3.03 / 3.29 / 2.09 (5.38)	56.11 / **59.80** / 43.03 (12.04) 2.79 / 3.71 / 3.62 (4.15)
Breast Cancer (30) 97.30 / 85.66	72.19 / **72.24** / 19.07 (46.08) 3.65 / 3.47 / 2.99 (6.07)	**57.74** / 55.07 / 18.80 (42.33) 2.93 / 3.11 / 5.34 (5.68)
Spam Data (57) 88.66 / 62.56	**77.38** / 77.38 / 53.86 (64.85) 1.69 / 1.71 / 1.43 (1.58)	**69.46** / 68.18 / 33.78 (35.43) 2.01 / 1.88 / 2.07 (2.31)
Sonar (60) 75.51 / 70.43	40.25 / **48.00** / -25.83 (-38.20) 28.42 / 26.68 / 16.87 (36.62)	40.33 / **44.96** / -47.86 (-27.45) 24.64 / 25.79 / 25.39 (36.74)
BCI (117) 79.55 / 60.08	27.75 / **34.12** / 31.10 (8.19) 7.75 / 6.60 / 5.79 (9.33)	15.12 / **15.47** / 11.76 (8.79) 5.30 / 6.26 / 3.20 (5.16)
Digit1 (241) 95.96 / 87.80	76.03 / **81.96** / 39.29 (40.06) 4.61 / 3.30 / 2.44 (4.16)	25.29 / **27.36** / 19.49 (17.70) 3.65 / 2.92 / 3.57 (3.28)
G241c (241) 87.42 / 76.00	**13.04** / 12.72 / -13.41 (32.45) 7.65 / 8.13 / 5.79 (5.77)	**9.54** / 9.46 / 3.86 (10.95) 3.34 / 2.79 / 1.61 (3.14)
G241n (241) 85.60 / 77.54	**18.34** / 16.55 / -18.93 (34.64) 7.07 / 7.54 / 8.16 (5.40)	**4.93** / 2.85 / 1.47 (6.89) 3.25 / 3.59 / 3.16 (4.49)

$\tau = 0.016$, $\alpha = 0.75$, $K = 8$ in Fig. 7. Note that the presented visual examples tend to generate image gradients that are naturally aligned with the image statistics. We also observe some cases, where the corruption along a border between the cells of our partitioning tree remains after the imputation (see the second last column in Fig. 7). The residual corruptions in such cases can be handled by increasing the depth of the tree or using m-ary trees refer to the trees, where one has m many splits at each nodes. For example, if $m = 2$, the tree is a binary tree. In addition to the visual comparisons, we also evaluate the performance of the introduced framework in terms of the classification purposes. On the described $0-1$ digit USPS data, we report the data scatter plots of the test instances in Fig. 8, where we project the original, corrupted, and imputed test data onto the two eigen vectors of the training set with the largest eigenvalues for visualization. We clearly observe a better class separation between two classes after the imputation, when compared with the class separation in the corrupted data.

We emphasize that since the proposed tree-based corruption separation framework is a comprehensive one such that it operationally covers the partial solutions in the corresponding literature, it is not possible to provide a perfectly fair comparative analysis. Nevertheless, we compare the proposed framework with a baseline of algorithms constructed using the methods [11], [19] in terms of the classification tasks over the several well-known machine learning data sets [12], [13]. One of these methods, tree-based corruption separation nearest neighbor (TCS-NN), consists of the same TCS procedure that we propose but utilizes—instead of our MAP imputation—but utilizes the Nearest Neighbor (NN) imputation technique [19], which finds the NN of a corrupted instance with respect to the sibling attributes of the corrupted node and imputes.

The other method, M-split with Nearest Neighbor Imputation (M-NN), also utilizes the NN imputation, but does not have a fine/detailed corruption separation step. Instead, it splits an instance into M different segments [11], applies anomaly detection to each segment, and imputes an anomalous segment by its NN that is found with respect to the neighboring segment.

In these experiments, we use a depth-4 tree for our Algorithm TCS leading to 16 leaves/segments in the finest level with $K = 8$. For each data set, after scaling each data attribute into the interval [0, 1], we randomly choose 11 splits of the scaled data such that 2/3 of each split is reserved as the training (reference) data set (at most 1000 instances), and the rest is reserved as the test data set (at most 500 instances) in each split. Moreover, every instance in the test set of each split is randomly corrupted/overwritten from the uniform distribution with the support [0, 1] in a random interval of attributes, which includes at least 10% of the attributes (dimensionality) and at most 50% of them. Since $30 = (50 + 10)/2\%$ of each test instance is corrupted on average, choosing $M = 4$ is appropriate for the method M-NN. We also present results for the case $M = 16$. The first split is used for parameter selection purposes,[1] e.g., C parameter of a linear SVM α

---

[1]The same values for the parameters α, τ is used for both TCS-MAP, and TCS-NN to fairly and clearly observe the effect of using NN imputation instead of MAP imputation since using the same rate τ for both methods leads to the same Constant False Alarm Rate (CFAR) in corruption detection. In another separate experiment (Table II), we directly and explicitly compare the two imputation methods with the standard Euclidean distance only. The same τ is also used for M-NN, which definitely favors M-NN since it corresponds to a lower CFAR for M-NN. Euclidean distance, $\alpha = 1$, is always used for M-NN. Depth is always 4 with $K = 8$. C is always common to all methods.

and the remaining 10 splits are used for performance analysis. Then, in each of the remaining 10 splits, we train a linear SVM classifier on the training set and compute the classification accuracy on the uncorrupted, corrupted, and the imputed test data. In Table I, we report the average of improvements and the corresponding standard deviation of the average improvement (standard deviation (std) of the improvements divided by $\sqrt{10}$ is reported, i.e., std of the mean value estimator) in terms of the classification accuracy on the imputed test data over 10 splits. The improvement in classification accuracy is calculated as

$$\% \text{ Improvement } = 100 \times \frac{\mathbf{acc}(\hat{\mathbf{X}}) - \mathbf{acc}(\mathbf{X})}{\mathbf{acc}(\bar{\mathbf{X}}) - \mathbf{acc}(\mathbf{X})} \quad (20)$$

where $\mathbf{acc}(\bar{\mathbf{X}})$ [$\mathbf{acc}(\mathbf{X}), \mathbf{acc}(\hat{\mathbf{X}})$] is the classification accuracy—after clean data training—on the original uncorrupted test data $\bar{\mathbf{X}}$ (on the corrupted test data $\mathbf{X}$, on the imputed data $\hat{\mathbf{X}}$). Furthermore, to evaluate the robustness to corruptions in the training sets and to address the cases where clean data is not available, we randomly choose the 5% of each training data of each split, corrupt with the same corruption model, and then repeat[2] the same experiments. The results are summarized in Table I.

We observe that the proposed framework is successful at undoing the adverse effects of corruption and always provides (positive) improvements up to ~80% after imputation in terms of the classification performance. In addition, when the drop in accuracy due to the corruptions is relatively low, e.g., 5 units of drop from 80% to 75%, even a 1 unit gain after imputation corresponds to 20% = (1/5) improvements, which naturally results in relatively high standard deviations (see Sonar data set in Table I). Moreover, the proposed framework is shown to be robust to corruptions in the training data by these experiments.

Our algorithms, TCS-MAP, perform significantly better than the method M-NN and comparably with the method TCS-NN. We point out that the method TCS-NN strongly relies on the proposed TCS framework, which is the main reason underlying the success of TCS-NN. (Note that both of the methods M-NN and TCS-NN uses the NN imputation but M-NN does not have the proposed TCS step and the method M-NN is outperformed by TCS-NN.) The method M-NN occasionally even further corrupts the data; see the negative improvements for Sonar or G241c or G241n. Moreover, it is difficult to choose between the methods 4-NN and 16-NN since there is no clear superiority. For example, there is definitely a scaling issue for the method M-NN. If the corruption in an instance covers a small portion, e.g., 10%, then choosing M too large would leave several undetectable corruptions in addition to imputing large chunks of clean data due to the false alarms (negative improvements). Similarly, if the corruption in an instance covers a large portion, e.g., 50%, then choosing M too small would not only again harden the anomaly detection (due to the use of insufficient data in detecting corruptions) but also complicate the imputation since on what to condition

the imputation becomes ambiguous. For a good imputation in this case, one would need to identify (with significant imperfections due the use of insufficient data because of a small M) all the corrupted and uncorrupted segments and then condition at which the imputation is on uncorrupted ones, which leads to a nonhomogenous different evaluation for every instance that requires computationally a very high load. Therefore, choosing an appropriate M is in general hard since it must depend on the amount of the corruption, which might be unknown and random. The proposed framework resolves this scaling issue in a computationally efficient way via the binary searches and fast imputations. Moreover, our algorithm TCS-MAP is experimentally shown to be also robust to corruptions in training data in the sense that it strongly preserves its corruption separation/imputation capabilities even after including 5% corruptions in training.

In our experiments with uncorrupted (clean) data training, the MAP imputation and the NN imputation (only when combined with the proposed tree based corruption separation) performs comparably, which is due to the sparsity of the data compared with the dimensionality. However, we first note that the MAP estimator yields the NN estimator, if the neighborhood size is set $K = 1$ in the imputation phase, which can easily be achieved via cross validation for K. Clearly, with such a cross validation, the MAP imputation can only perform better than the NN imputation and we opt not to optimize K for presentational clarity. In addition, the NN imputation is definitely sensitive to corruptions in the training data since the attributes of the NN that are used for imputation can also be corrupted with a certain probability, e.g., with probability 0.05 in our experiments. On the other hand, that possibly (with probability 0.05) corrupted NN would achieve a lower score value $\hat{p}_K(x)$ if it was truly corrupted and it would not be picked by our MAP estimator for imputation. Thus, the proposed MAP estimator can handle such situations and is robust to corruptions, where the NN imputation performance potentially degrades more. In our experiments, the MAP imputation either enhances its superiority or becomes superior, or approaches NN imputation for most of the data sets after including corruptions in training, e.g., Image or Ringnorm data sets in Table I.

To further demonstrate the power of the proposed estimator, we devise a separate experiment, where we use a data set consisting of two Gaussian components with unitary covariances. We use the means $[1, -1]$ and $[-1, 1]$ for positive and negative classes, respectively. We generate 1000 samples as the training data, (500 for each class) and next suppose that the second attribute of each sample is corrupted/missing and, therefore, imputed by our MAP estimator with varying neighborhood size K and the NN estimator. Note that in this part, we use standard Euclidean distance only. After repeating this 100 times, the resulting imputed data is compared with the original training data in the MSE sense and in terms of the classification accuracy. We summarize our findings in Table II, where the MAP imputation with $K = 1$ coincides with the NN imputation. The MAP imputation is consistently better than the NN imputation as expected. For instance, when $K = 12$, we obtain 26% improvements in the mse sense and 10% improvements in terms of the classification (original classification accuracy is around 92%).

---

[2]Exactly the same splits with the same parameters are used. The only difference is that 5% of each training set is corrupted. Note that in these experiments, we first train a linear SVM using clean training data and then, test the trained model on uncorrupted (clean) test data, corrupted test data, and corrupted test data imputed by the clean training data and corrupted test data imputed by %5 corrupted training data.

TABLE II

PERFORMANCE (MSE AND CLASSIFICATION) OF THE PROPOSED MAP
IMPUTATION ON THE DATA SAMPLED FROM A 2 COMPONENT
GAUSSIAN MIXTURE MODEL. THE MAP IMPUTATION IS
CONSISTENTLY BETTER THAN THE NN ($K = 1$)
IMPUTATION FOR ALL Ks

K	1	4	8	12	16
MSE	2.90	2.31	2.19	2.14	2.12
% Classification Accuracy	79.49	80.13	80.45	80.74	80.86

VI. CONCLUSION

In this paper, we proposed a comprehensive framework for handling localized and severe data corruptions. The novel contributions of the proposed framework includes: 1) a first algorithm to jointly detect and localize such corruptions by identifying the local anomalies; 2) a maximum *a posteriori*-based estimator for imputation and a distance measure for corruption separation purposes; 3) computational efficiency via the binary searches and the fast imputations; and 4) a characterization for anomalous observations, e.g., rarities, incompatible combinations, and corruptions. We point out that our algorithm does not assume prior information or a model for the input data and, instead, works in a completely data driven way. Furthermore, we conducted a false alarm rate analysis and showed that the desired false alarm rate in detecting corruptions can be set independently with the input data. Our algorithm is tested against the corruptions in several well-known machine learning data sets and experimentally shown to provide significant improvements in terms of classification purposes with strong corruption separation. The proposed algorithms outperform the typical approaches and are robust to varying training phase conditions.

REFERENCES

[1] R. He, W.-S. Zheng, B.-G. Hu, and X.-W. Kong, "Two-stage nonnegative sparse representation for large-scale face recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 35–46, Jan. 2013.

[2] S. Zafeiriou, G. Tzimiropoulos, M. Petrou, and T. Stathaki, "Regularized kernel discriminant analysis with a robust kernel for face recognition and verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 526–534, Mar. 2012.

[3] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Netw.*, vol. 10, no. 2, pp. 439–443, Mar. 1999.

[4] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[5] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[6] G. Li, C. Wen, Z. G. Li, A. Zhang, F. Yang, and K. Mao, "Model-based online learning with kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 356–369, Mar. 2013.

[7] C. D. Scott and R. D. Nowak, "Learning minimum volume sets," *J. Mach. Learn. Res.*, vol. 7, pp. 665–704, Jan. 2006.

[8] M. Zhao and V. Saligrama, "Anomaly detection with score functions based on nearest neighbor graphs," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran & Associates Inc., 2009.

[9] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009, Art. ID 15.

[10] R. Perdisci, G. Gu, and W. Lee, "Using an ensemble of one-class SVM classifiers to harden payload-based anomaly detection systems," in *Proc. 6th Int. Conf. Data Mining (ICDM)*, Dec. 2006, pp. 488–498.

[11] V. Saligrama and M. Zhao, "Local anomaly detection," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 969–983.

[12] K. Bache and M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[13] J. Kim and C. D. Scott, "L_2 kernel classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1822–1831, Oct. 2010.

[14] J. Sun and S. Keates, "Canonical correlation analysis on data with censoring and error information," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 12, pp. 1909–1919, Dec. 2013.

[15] Y. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 72–87, Jan. 2011.

[16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the *EM* algorithm," *J. Roy. Statist. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

[17] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys* (Wiley Series in Probability and Mathematical Statistics). New York, NY, USA: Wiley, 1987.

[18] R. J. A. Little and D. B. Rubin, *Statistical Analysis With Missing Data*, vol. 539. New York, NY, USA: Wiley, 1987.

[19] O. Troyanskaya *et al.*, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[20] P. Smolensky, *Information Processing in Dynamical Systems: Foundations of Harmony Theory*. Cambridge, MA, USA: MIT Press, 1986.

[21] G. E. A. P. A. Batista and M. C. Monard, "An analysis of four missing data treatment methods for supervised learning," *Appl. Artif. Intell., Int. J.*, vol. 17, nos. 5–6, pp. 519–533, 2003.

[22] J. Besag, "Statistical analysis of non-lattice data," *J. Roy. Statist. Soc., Ser. D (The Statistician)*, vol. 24, no. 3, pp. 179–195, 1975.

[23] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie, "Dependency networks for inference, collaborative filtering, and data visualization," *J. Mach. Learn. Res.*, vol. 1, pp. 49–75, Jan. 2001.

[24] B. M. Marlin, "Missing data problems in machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2008.

[25] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 1994.

[26] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, Jan. 2011.

[27] Q. Wang and J. N. K. Rao, "Empirical likelihood-based inference under imputation for missing response data," *Ann. Statist.*, vol. 30, no. 3, pp. 896–924, 2002.

[28] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 849–862, Apr. 2013.

[29] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 60–65.

[30] Y. Weiss and W. T. Freeman, "What makes a good model of natural images?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[31] S. Lyu and E. P. Simoncelli, "Statistical modeling of images with fields of Gaussian scale mixtures," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2006.

[32] P. Galllnarl, Y. Le Cun, S. Thlrla, and F. Fogelman Soullé, "Mémoires associatives distribuees: Une comparaison," in *Proc. Cognitiva*, vol. 87. 1987, p. 93.

[33] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran & Associates Inc., 2008.

[34] M. Ranzato, V. Mnih, J. M. Susskind, and G. E. Hinton, "Modeling natural images using gated MRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2206–2222, Sep. 2013.

[35] P.-S. Toh and A. K. Forrest, "Occlusion detection in early vision," in *Proc. 3rd Int. Conf. Comput. Vis.*, Dec. 1990, pp. 126–132.

[36] N. H. C. Yung and A. H. S. Lai, "Detection of vehicle occlusion using a generalized deformable model," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4. May/Jun. 1998, pp. 154–157.

[37] C. C. C. Pang, W. W. L. Lam, and N. H. C. Yung, "A novel method for resolving vehicle occlusion in a monocular traffic-image sequence," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 3, pp. 129–141, Sep. 2004.

[38] W. Zhang, J. Q. Wu, X. Yang, and X. Fang, "Multilevel framework to detect and handle vehicle occlusion," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 161–174, Mar. 2008.

[39] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, Apr. 2001.

[40] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, "Multi-cue pedestrian classification with partial occlusion handling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 990–997.

[41] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 32–39.

[42] X. Mei, H. Ling, Y. Wu, E. Blasch, and L. Bai, "Minimum error bounded efficient ℓ_1 tracker with occlusion detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1257–1264.

[43] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 1. New York, NY, USA: Springer-Verlag, 2006.

[44] M. A. Donmez and S. S. Kozat, "Steady state and transient MSE analysis of convexly constrained mixture methods," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 3314–3321, Jun. 2012.

[45] S. Godsill, A. Doucet, and M. West, "Maximum *a posteriori* sequence estimation using Monte Carlo particle filters," *Ann. Inst. Statist. Math.*, vol. 53, no. 1, pp. 82–96, 2001.

[46] Y. Bar-Shalom and X.-R. Li, *Multitarget-Multisensor Tracking: Principles and Techniques*. New York, NY, USA: Academic, 1995.

[47] J. M. Mendel and C. S. Burrus, *Maximum-Likelihood Deconvolution: A Journey into Model-Based Signal Processing* (Signal Processing and Digital Filtering). New York, NY, USA: Springer-Verlag, 1990.

[48] G. K. Chantas, N. P. Galatsanos, and A. C. Likas, "Bayesian restoration using a new nonstationary edge-preserving image prior," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2987–2997, Oct. 2006.

[49] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.

[50] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[51] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, vol. 26. Boca Raton, FL, USA: CRC Press, 1986.

[52] K. Thulasiraman and M. N. S. Swamy, *Graphs: Theory and Algorithms*. New York, NY, USA: Wiley, 2011.

Huseyin Ozkan received the B.Sc. degrees in electrical engineering and mathematics from Bogazici University, Istanbul, Turkey, in 2007, and the M.Sc. degree in electrical engineering from Boston University, Boston, MA, USA, in 2010. He is currently pursuing the Ph.D. degree in electrical engineering with Bilkent University, Ankara, Turkey.

His current research interests include statistical learning, statistical signal processing, and computer vision. He is also with the MGEO Division, Aselsan Inc., Ankara, where he conducts research about large area surveillance.

Ozgun Soner Pelvan received the B.Sc. degree in electrical engineering from Middle East Technical University, Ankara, Turkey, in 2004, and the M.Sc. degree from Kaiserslautern Technical University, Kaiserslautern, Germany, in 2007. He is currently pursuing the Ph.D. degree in electrical engineering with Middle East Technical University. His current research interests include statistical signal processing and wireless networks.

Suleyman S. Kozat (SM'99) received the B.Sc. degree from Bilkent University, Ankara, Turkey, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2001 and 2004, respectively.

He joined the IBM Research Thomas J. Watson Research Center, Yorktown Heights, NY, USA, as a Research Staff Member with the Pervasive Speech Technologies Group, where he focused on problems related to statistical signal processing and machine learning. He was a Research Associate with the Cryptography and Anti-Piracy Group, Microsoft Research, Redmond, WA, USA. He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Bilkent University. His current research interests include intelligent systems, adaptive filtering for smart data analytics, online learning, and machine learning algorithms for signal processing.

Dr. Kozat was a recipient of the IBM Faculty Award by IBM Research in 2011, the Outstanding Faculty Award by Koc University, İstanbul, Turkey, in 2011, the Outstanding Young Researcher Award by the Turkish National Academy of Sciences in 2010, the ODTU Prof. Dr. Mustafa N. Parlar Research Encouragement Award in 2011, and the Career Award by the Scientific Research Council of Turkey in 2009.