Comprehensible Counterfactual Explanation on Kolmogorov-Smirnov Test

Zicun Cong
Simon Fraser University
zcong@sfu.ca

Lingyang Chu
McMaster University
chul9@mcmaster.ca

Yu Yang
City University of Hong Kong
yuyang@cityu.edu.hk

Jian Pei
Simon Fraser University
jpei@cs.sfu.ca

ABSTRACT

The Kolmogorov-Smirnov (KS) test is popularly used in many applications, such as anomaly detection, astronomy, database security and AI systems. One challenge remained untouched is how we can obtain an explanation on why a test set fails the KS test. In this paper, we tackle the problem of producing counterfactual explanations for test data failing the KS test. Concept-wise, we propose the notion of most comprehensible counterfactual explanations, which accommodates both the KS test data and the user domain knowledge in producing explanations. Computation-wise, we develop an efficient algorithm MOCHE (for MOst CompreHensible Explanation) that avoids enumerating and checking an exponential number of subsets of the test set failing the KS test. MOCHE not only guarantees to produce the most comprehensible counterfactual explanations, but also is orders of magnitudes faster than the baselines. Experiment-wise, we present a systematic empirical study on a series of benchmark real datasets to verify the effectiveness, efficiency and scalability of most comprehensible counterfactual explanations and MOCHE.

PVLDB Reference Format:
Zicun Cong, Lingyang Chu, Yu Yang, and Jian Pei. Comprehensible Counterfactual Explanation on Kolmogorov-Smirnov Test. PVLDB, 14(9): 1583-1596, 2021.
doi:10.14778/3461535.3461546

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/research0610/MOCHE.

1 INTRODUCTION

The well-known Kolmogorov-Smirnov (KS) test [31] is a statistical hypothesis test that checks whether a test set is sampled from the same probability distribution as a reference set. If a reference set and a test set fail the KS test, it indicates that the two sets are unlikely from the same probability distribution. The KS test has been widely used to detect differences, changes and abnormalies in many areas, such as astronomy [43], database security [56] and AI systems [51]. Many important decisions are made based on the raised alarms about changes and abnormality, such as updating an AI model [66] and overhauling a manufacture line [30]. Understanding why a test set fails a KS test can build trust from users [54] and thus improve the decision quality. In some situations, the understanding may help save data labeling and model construction costs [66]. However, a failed KS test itself does not come with an explanation on which data points in the test set cause the failure.

Counterfactual explanations [41] have been widely adopted to interpret algorithmic decisions in many real world applications [16, 25, 42, 61], due to its beauty of being concise and easy to understand [41, 58]. A counterfactual explanation of a decision Y is the smallest set of relevant factors X such that changing X can alter the decision Y [41, 61]. For a *failed KS test*, where a reference set R and a test set T fail the KS test, a counterfactual explanation is a minimum subset S of the test set T such that removing the subset from T reverses the failed KS test into a passed one, that is, R and $T \setminus S$ pass the KS test. Using counterfactual explanations to interpret failed KS tests helps users gain more insights into changes and differences behind the failed KS tests.

Example 1 (Motivation). A public health officer may want to compare the distributions of COVID-19 cases reported in August and September 2020 in the province of British Columbia, Canada. She may use the cases in August as the reference set and those in September as the test set, where each COVID-19 reported case is associated with the age group of the patient. The cases are divided into 10 age groups, (0-10), (10-19), . . ., (80-89), and (90+). A failed KS test between the two sets suggests that the infected cases in those two months unlikely follow the same distribution on age groups. This information may be helpful to review the infection control policies. As a KS test may raise false alarms [50], the officer may want to have a counterfactual explanation on this failed KS test, which reveals the cases that are likely relevant to the change. Example 2 and our case study in Section 6.3 illustrate how such counterfactual explanations can help us to understand the changes.

Although interpreting failed KS tests is interesting and has many potential applications, it has not been touched in literature. Although there are many counterfactual explanation methods interpreting the decisions of machine learning models [9, 25, 36], unfortunately, the existing methods cannot be adopted to interpret failed KS tests. As reviewed in Section 2, to interpret a failed KS test, the existing methods have to solve an L_0-norm optimization problem, which is NP-hard [39]. Some methods [49, 51] try to select

Proceedings of the VLDB Endowment, Vol. 14, No. 9 ISSN 2150-8097.
doi:10.14778/3461535.3461546

the outliers in the test set as a hint to a failed KS test. However, because outlier detection methods and KS tests use different mechanisms to detect anomalies, there is no guarantee that the outliers are relevant to the failure of a KS test. Moreover, due to the Roshomon effect [41], multiple counterfactuals may co-exist for a failed KS test but not all of them are comprehensible to users [41]. Simply presenting all counterfactuals not only may overwhelm users but is also computationally expensive [40].

A desirable idea is to find a counterfactual explanation that is most consistent with a user's domain knowledge so that the explanation is best comprehensible to the user [17, 42]. However, none of the existing methods can find such most comprehensible explanations. Moreover, finding the most comprehensible explanation is far from trivial. A brute force method has to enumerate all subsets of a test set and, for each subset, conduct a KS test. Thus, the brute force method takes exponential time.

In this paper, we tackle the novel problem of producing counterfactual explanations for failed KS tests. We make several contributions. Concept-wise, we propose the notion of comprehensible counterfactual explanations. Given a failed KS test, we find a smallest subset of the test set such that removing the subset from the test set reverses the failed KS test into a passed one. To address user comprehensibility [17, 41], we take a user's domain knowledge represented as a preference order on the data points in the test set, and guarantee to find the counterfactual explanation that is most consistent with the preference. Computation-wise, we develop MOCHE (for MOst CompreHensible Explanation), a two-step fast method that guarantees to find the most comprehensible counterfactual explanation on a failed KS test. Specifically, MOCHE first identifies the number of data points in the explanation and then efficiently constructs the most comprehensible explanation. We establish an important insight that the size of removed data points is the smallest integer satisfying a group of inequalities. Leveraging this property, an efficient searching algorithm is designed to find the explanation size. Then, MOCHE efficiently constructs the most comprehensible explanation by one scan of the data points in the test set. Experiment-wise, we conduct a systematic empirical study on a series of benchmark real datasets to verify the effectiveness, efficiency and scalability of most comprehensible counterfactual explanations and MOCHE.

2 RELATED WORK

To the best of our knowledge, interpreting a failed KS test is a novel task that has not been systematically investigated in literature. Our study is broadly related to the Kolmogorov-Smirnov test [29, 30, 57], counterfactual explanations [9, 25, 36], adversarial attacks [14, 21, 47] and outlier detection [22, 27, 52].

The Kolmogorov-Smirnov (KS) test [31] is a well-known statistical hypothesis test that checks whether two samples are originated from the same probability distribution. With the advantages of being efficient, non-parametric, and distribution-free [33], the KS test has been widely used in many applications to detect differences, changes and abnormalities [22], such as identifying change points in time series [22, 30], maintaining machine learning models [23, 51, 57], ensuring quality of encrypted or anonymized data [8, 28], and protecting databases from intrusion attacks [56].

As illustrated in Section 1 and further elaborated later, understanding why a KS test is failed may be important in real world applications [49, 51]. However, a failed KS test itself does not provide any hints on which data points in the test set may be related to the failure. Therefore, finding explanations of failed KS tests is a natural next step.

Counterfactual explanations [41, 58, 61] have been widely adopted to interpret algorithmic decisions made in many real world applications [9, 25, 36]. Those methods [9, 25, 37, 41] interpret a prediction on a given instance by applying small and interpretable perturbations on the instance such that the prediction is changed [41]. For example, Fong *et al.* [25] interpret the prediction of an image by finding the smallest pixel-deletion mask that leads to the most significant drop of the prediction score. As an extension, Akula *et al.* [9] identify meaningful image patches that need to be added to or deleted from an input image. Van Looveren *et al.* [37] use class prototypes to generate counterfactuals that lie close to the classifier's training data distribution. Le *et al.* [36] use an entropy-based feature selection approach to limit the features to be perturbed.

Unfortunately, the existing counterfactual explanation methods cannot effectively and efficiently interpret a failed KS test by perturbing the data points in the test set. This is because, to minimize the number of perturbed data points, the existing methods need to minimize the L_0-norm of their perturbations [39]. However, such an optimization problem is NP-hard [39, 45]. The existing methods cannot guarantee to reach a global minimum for the optimization problem in an efficient manner.

One may think adversarial attack methods [14, 20, 21, 47, 48] may be extended to find counterfactual explanations on failed KS tests. To attack a target classifier, an adversarial attack method generates an imperceptible perturbation on an input so that the prediction on the input is changed. Brendel *et al.* [14] propose to generate adversarial perturbations by moving instances towards the estimated decision boundaries of a target model. Cheng *et al.* [20] formulate the black-box attack as an optimization problem, which can be solved by the zeroth order optimization approaches. Croce *et al.* [21] propose to attack image classifiers by applying randomly selected one-pixel modifications on images. One may generate counterfactual explanations on a failed KS test by attacking the KS test, that is, the perturbed data points can serve as a counterfactual explanation on the KS test. However, extending the existing adversarial attack methods to interpret failed KS tests also needs to minimize the L_0-norm of the perturbations and leads to the same computational challenge.

Outlier detection methods aim to detect samples that are different from the majority of the given data [7], such as distance-based approaches [10, 13, 27, 52], density-based approaches [15, 26, 46, 59] and ensemble-based approaches [22, 35]. In general, outliers are regarded as abnormal data points [7].

Even though both the KS test and outlier detection methods can detect anomalies in data, the detected outliers in the test set cannot be used as a counterfactual explanation on a failed KS test. This is because outlier detection methods and the KS test use different mechanisms to detect anomalies. Different from the KS test, the outlier detection methods do not compare the distributions of the reference set and the test set. Therefore, there is no guarantee that

outliers can explain a failed KS test. Just removing the outliers cannot guarantee to reverse a failed KS test to a passed one.

3 PROBLEM FORMULATION AND ANALYSIS

In this section, we first review the basics of the Kolmogorov-Smirnov (KS) test. Then, we investigate how to generate a counterfactual explanation on a KS test. Third, we discuss the comprehensibility of explanations, and formalize the problem of finding the most comprehensible explanation on a failed KS test. Next, we investigate the existence and uniqueness of most comprehensible counterfactual explanations. Last, we describe a brute force method.

3.1 The Kolmogorov-Smirnov Test

Denote by $R = \{r_1, \ldots, r_n\}$ a multi-set of real numbers from an unknown univariate probability distribution, and by $T = \{t_1, \ldots, t_m\}$ another multi-set of real numbers that are sampled from a distribution that may or may not be the same as R. We call R a *reference set* and T a *test set*. In this paper, by default multi-set is used. In the rest of the paper, we use the terms "set" and "multi-set" interchangeably unless specifically mentioned.

The Kolmogorov-Smirnov (KS) test checks whether T is sampled from the same probability distribution as R by comparing the empirical cumulative functions of R and T. In the KS test, the null hypothesis is that T is sampled from the same probability distribution as R.

Conducting the KS test consists of 3 steps as follows.

Step 1. We compute the *KS statistic* [23] by

$$D(R,T) = \max_{x \in R \cup T} |F_R(x) - F_T(x)|, \tag{1}$$

where $F_R(x)$ and $F_T(x)$ are the empirical cumulative functions of R and T, respectively. Here, a larger value of $D(R,T)$ indicates that the empirical cumulative functions of R and T are more different from each other.

Step 2. For a user-specified significance level α, we compute the corresponding target p-value [23] by $p = c_\alpha \sqrt{\frac{n+m}{n*m}}$, where $c_\alpha = \sqrt{-\frac{1}{2}\ln(\frac{\alpha}{2})}$ is the critical value at significance level α, $n = |R|$ is the number of data points in R, and $m = |T|$.

Step 3. We compare p and $D(R,T)$. If $D(R,T) > p$, we reject the null hypothesis at significance level α. This means the empirical cumulative functions $F_R(x)$ and $F_T(x)$ are significantly different from each other, and thus it is unlikely T is sampled from the same distribution as R. If $D(R,T) \leq p$, we cannot reject the null hypothesis at significance level α. There is not enough evidence showing that T is not sampled from the same distribution as R.

If the null hypothesis is rejected by the KS test, we say R and T *fail* the KS test and it is a *failed KS test*. Otherwise, we say R and T *pass* the KS test.

To compute the KS statistic between R and T, we need to sort the elements in $R \cup T$ in ascending order. Therefore, it takes $O((n+m)\log(n+m))$ time to conduct the KS test.

3.2 Counterfactual Explanations on the KS Test

Why are we interested in failed KS tests? More often than not, a failed hypothesis test indicates something unusual or unexpected [18, 33, 51]. As many important decisions are made based on failed KS tests [30, 66], it is important to interpret a failed KS test so that we can make better responses to the change and abnormality alarms.

Counterfactual explanation is an explanation technique proposed by the community of explainable artificial intelligence. It has been well demonstrated to be more human-friendly than other types of explanations [41, 58]. The counterfactual explanation methods interpret a decision Y by finding a smallest set of relevant factors X, such that changing X can alter the decision Y [41, 61]. The set of factors X is called a counterfactual explanation on Y. Following the above principled idea, we have the following definition.

Definition 1. *For a reference set R and a test set T that fail the KS test at a significance level α, a **counterfactual explanation** on the failed KS test is a smallest subset $\mathcal{I}$ of the test set T, such that R and $T \setminus \mathcal{I}$ pass the KS test at the same significance level α.*

A counterfactual explanation is also called an *explanation* for short when the context is clear.

3.3 The Most Comprehensible Counterfactual Explanation on a KS Test

Like many previously proposed counterfactual explanations [25, 42, 61], the counterfactual explanations on a failed KS test suffer from the *Roshomon effect* [41], that is, the number of unique counterfactual explanations on a failed KS test can be as large as $\binom{|T|}{|\mathcal{I}|}$. Simply presenting all counterfactuals not only may overwhelm users but is also computationally expensive [40].

As discovered by many studies on counterfactual explanations [32, 42, 58], not all counterfactual explanations are equally comprehensible to a user. Due to the effect of confirmation bias [44], an explanation is more comprehensible if it is more consistent with the user's domain knowledge [38]. As a result, a typical way to overcome the Roshomon effect is to rank all explanations according to the user's preference based on the domain knowledge, and return the most preferred explanation to the user [11, 42].

Following the above idea, we model a user's preference as a total order on the data points in the test set T, that is, a *preference list* L on the test set T. Each data point has a unique rank in L. The data points having smaller ranks in L are more preferred by the user.

A typical task of recommendation system is to recommend a group of items to a user, such that the group best satisfies the user's preference [64]. The existing studies [12, 19, 60, 64] discover that a user's interest in a group is dominated by the user's top favorite items in the group. In the same vein, one can think of an explanation as a recommended group of data points. Given two explanations $\mathcal{I}_1$ and $\mathcal{I}_2$ on a failed KS test, if $\mathcal{I}_1$ includes better-ranked data points in L than $\mathcal{I}_2$ does, $\mathcal{I}_1$ is more preferred by the user than $\mathcal{I}_2$, and thus is more comprehensible.

Based on the above intuition, an explanation with a smaller lexicographical order[1] based on the preference list L is more preferred by the user. Specifically, for two explanations $\mathcal{I}_1$ and $\mathcal{I}_2$, $\mathcal{I}_1 \subseteq T$, $\mathcal{I}_2 \subseteq T$ and $|\mathcal{I}_1| = |\mathcal{I}_2|$, we sort the data points in $\mathcal{I}_1$ and $\mathcal{I}_2$ in the order of L. Denote by $\mathcal{I}[i]$ the i-th data point in $\mathcal{I}$ in the order

[1] Given a total order $\prec_L$ on items, the lexicographical order $\prec_{\text{lexicographical}}$ is $x_1 x_2 \cdots x_n \prec_{\text{lexicographical}} y_1 y_2 \cdots y_l$ if (1) $x_1 \prec_L y_1$; (2) there exists i_0 ($1 < i_0 \leq \min\{n, l\}$), $x_i = y_i$ for $1 \leq i < i_0$ and $x_{i_0} \prec_L y_{i_0}$; or (3) $m < l$ and for $1 \leq i \leq m$, $x_i = y_i$. Lexicographical order is also known as dictionary order.

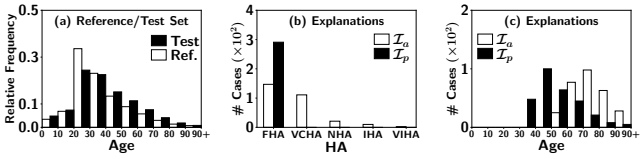

Figure 1: The histograms of (a) the reference set and the test set; (b) the distributions of the explanations $\mathcal{I}_a$ and $\mathcal{I}_p$ in Example 2 on HAs; and (c) those on age groups.

of L. Let $i_0 > 0$ be the smallest integer such that $\mathcal{I}_1[i_0] \neq \mathcal{I}_2[i_0]$. $\mathcal{I}_1$ precedes $\mathcal{I}_2$ in the lexicographical order, denoted by $\mathcal{I}_1 \prec \mathcal{I}_2$, if $\mathcal{I}_1[i_0]$ precedes $\mathcal{I}_2[i_0]$ in L. If $\mathcal{I}_1 \prec \mathcal{I}_2$, $\mathcal{I}_1$ includes more top-ranked data items in L than $\mathcal{I}_2$.

Definition 2. *Given a failed KS test and a preference list L, the **most comprehensible counterfactual explanation** is the explanation that has the smallest lexicographical order based on L.*

The notion of comprehensible explanation captures user preferences in domain knowledge. In order to capture different domain knowledge, we can employ different preference lists to sort the data points in the test set.

Example 2. Let us consider the KS test conducted on the COVID-19 cases discussed in Example 1. The reference set (August) and the test set (September) have 2,175 and 3,375 data points, respectively. The histograms of the two sets are shown in Figure 1a. Each bin on the X-axis represents an age group. Please refer to Section 6.1 for more details about the dataset. The two sets fail the KS test with significance level $\alpha = 0.05$.

Since COVID-19 may be more contagious in regions of larger population, a public health officer may sort the reported cases into a preference list L_p in population descending order of the reported health authority (HA for short). Please see Reference [3] for details about the HAs in British Columbia. The data points from HAs with large population are ranked higher, while the cases from the same HAs are sorted arbitrarily.

COVID-19 is also known to hit seniors harder. Alternatively, the officer may sort the reported cases into a preference list L_a in age group descending order. People in more senior age groups are ranked higher, while the cases from the same age group are sorted arbitrarily.

Given preference lists L_p and L_a, our method MOCHE produces the corresponding most comprehensible counterfactual explanations $\mathcal{I}_p$ and $\mathcal{I}_a$, respectively. Figures 1b and 1c show the distributions of the two explanations on HAs and age groups, respectively. The X-axis of Figure 1b shows the HA ids from left to right in population descending order. Both $\mathcal{I}_a$ and $\mathcal{I}_p$ include 291 data points.

As shown in Figure 1b, all data points in $\mathcal{I}_p$ are from FHA (Fraser HA), the HA with the largest population. Based on L_p, we have $\mathcal{I}_p \prec \mathcal{I}_a$ in lexicographical order, that is, $\mathcal{I}_p$ is more preferable than $\mathcal{I}_a$ when HAs of large population are concerned. As shown in Figure 1c, $\mathcal{I}_a$ contains more senior people. Based on L_a, we have $\mathcal{I}_a \prec \mathcal{I}_p$ in lexicographical order, that is, $\mathcal{I}_a$ is more preferable when senior people are more concerned.

$\mathcal{I}_a$ and $\mathcal{I}_p$ together suggest that some care homes in FHA (Fraser HA) may have COVID outbreaks, which matches the reality.

3.4 Existence and Uniqueness

For a failed KS test at significance level α, our task is to find the most comprehensible counterfactual explanation $\mathcal{I}$ on the KS test. Does such an explanation always exist? If so, is the most comprehensible counterfactual explanation unique?

Proposition 1. *When the significance level $\alpha \leq \frac{2}{e^2}$, there exists a unique most comprehensible explanation on a failed KS test.*

Proof. (Existence) Consider a subset $S \subset T$, where $|S| = |T| - 1$. $|T \setminus S| = 1$. The p-value of the KS test between R and $T \setminus S$ is $p = c_\alpha \sqrt{\frac{n+1}{n*1}}$, where n is the size of R and $c_\alpha = \sqrt{-\ln(\frac{\alpha}{2}) * \frac{1}{2}}$. Since $\alpha \leq \frac{2}{e^2}$, we have $c_\alpha \geq 1$ and $p \geq \sqrt{\frac{n+1}{n}} \geq 1$. Since $D(R, T \setminus S)$ is the absolute difference between the two empirical cumulative functions, $1 \geq D(R, T \setminus S)$. Therefore, $p \geq D(R, T \setminus S)$. That is, R and $T \setminus S$ pass the KS test. Since an explanation is a smallest subset that reverses a failed KS test to a passed one, there must exist an explanation on a failed KS test given R and $T \setminus S$ passing the test.

(Uniqueness) Since each data point has a unique rank in L, two distinct explanations cannot be equivalent in the lexicographical order. Thus, the most comprehensible explanation is unique. □

Statistical tests in practice typically use a significance level of 0.05 or lower. $\frac{2}{e^2} > 0.27$, which is far over the range of significance levels used in statistical tests. Therefore, our problem formulation is practical and guarantees a unique solution in practice.

3.5 A Brute Force Method

A naïve method to find the most comprehensible explanation is to enumerate all subsets of the test set T and check against Definition 2. This brute-force method checks an exponential number of subsets, which is prohibitive for large test sets.

Even in a brute force method, we can significantly reduce the number of subsets that need to be checked by early pruning a large number of subsets. According to Definitions 1 and 2, we can sort all subsets of T first by the size, from small to large, and then by the lexicographical order. This can be done by a breadth-first traversal of a set enumeration tree [55]. The first subset S in this order such that R and $T \setminus S$ pass the KS test is the most comprehensible explanation.

4 SEARCHING FOR EXPLANATION SIZE

In this section, we first describe the two-phase framework of MOCHE (for MOst CompreHensible Explanation), a fast method to find the most comprehensible counterfactual explanations. Then, we thoroughly explore how to compute the size of explanations fast.

4.1 MOCHE

According to Definition 1, all explanations have the same size. Once we find an explanation $\mathcal{I}$, we can safely ignore all subsets of T whose sizes are not equal to $|\mathcal{I}|$, no matter they can reverse the KS test or not. Based on this idea, the MOCHE method proceeds in two phases. In phase 1, MOCHE tries to find the size of explanations. In phase 2, MOCHE tries to identify the most comprehensible explanations, that is, the smallest one in lexicographical order.

A subset S of T such that $|S| = h$ is called an *h-subset*. An h-subset S is a *qualified h-subset* if R and $T \setminus S$ pass the KS test. The first bottleneck is to check, for a given $h > 0$, whether there exists a qualified h-subset S. A brute-force implementation has to conduct the KS test a large number of times on all h-subsets. The time complexity is $O(\binom{m}{h}(m+n-h)\log(n+m-h))$.

Our first major technical result in this section is that checking the existence of a qualified h-subset does not have to conduct the KS test on all h-subsets. With a carefully designed data structure named cumulative vector to represent an h-subset of T, we establish a fast verification method for qualified cumulative vectors. Checking the existence of a qualified h-cumulative vector and thus a qualified h-subset only takes $O(m+n)$ time.

The second bottleneck is to find the size of explanations efficiently. A brute-force method has to search from 1 to $m-1$ one by one and, for each size h, check the h-subsets. The second major technical result in this section tackles this bottleneck by deriving a lower bound $\hat{k}$ on k, the size of all explanations. This lower bound reduces the search range of h from $[1,k]$ to $[\hat{k},k]$, which further reduces the time complexity of phase 1 to $O((m+n)\log(m)+(k-\hat{k})(m+n))$.

4.2 Cumulative Vectors

Essentially, the KS test compares the cumulative distribution functions of a reference set and a test set. Since there are only finite numbers of data points in a reference set and a test set, we can represent the cumulative distribution function of a reference set, a test set or a subset of the test set using a sequence of the values of the cumulative distribution function at the data points appearing at either the reference set or the test set. This observation motivates the design of the cumulative vectors.

We make a *base vector* $\mathbf{V} = \langle x_1, \ldots, x_q \rangle$ from sets R and T, such that $x_1, \ldots, x_q$ are the unique data points in $R \cup T$. No matter how many times x_i appears in $R \cup T$, it only appears once in $\mathbf{V}$. Thus, $q = \|R \cup T\|$, where R and T are treated as sets instead of multi-sets and q is the cardinality of the union, that is, duplicate items are not double counted. The elements in $\mathbf{V}$ are sorted in the value ascending order, that is $x_1 < x_2 < \cdots < x_q$.

Definition 3. *The* cumulative vector *of an h-subset $S \subseteq T$ is a $(q+1)$-dimensional vector $\mathbf{C}_S = \langle c_0, c_1, \ldots, c_q \rangle$, where $c_0 = 0$, and for $1 \le i \le q$, c_i is the number of data points in S that are smaller than or equal to x_i in $\mathbf{V}$, that is $c_i = |\{x \in S \mid x \le x_i\}|$. We also write c_i as $\mathbf{C}_S[i]$.*

Example 3. Consider a test set $T = \{13, 13, 12, 20\}$ and a reference set $R = \{14, 14, 14, 14, 20, 20, 20, 20\}$. The base vector $\mathbf{V} = \langle 12, 13, 14, 20 \rangle$. For a subset $S = \{13, 13\}$ of T, the cumulative vector is $\mathbf{C}_S = \langle 0, 0, 2, 2, 2 \rangle$.

According to Definition 3, a cumulative vector $\mathbf{C}_S$ contains all information to derive the cumulative distribution function $F_{T\setminus S}$ straightforwardly. For a cumulative vector $\mathbf{C}_S = \langle c_0, c_1, \ldots, c_q \rangle$ and any i $(1 \le i \le q)$, $c_i - c_{i-1}$ is the number of times that x_i appears in S. Thus, the value of the empirical cumulative distribution function of $T \setminus S$ at x_i can be computed by $F_{T\setminus S}(x_i) = \frac{\mathbf{C}_T[i]-c_i}{m-c_q}$, where $\mathbf{C}_T$ is the cumulative vector of T and $\mathbf{C}_T[i]$ is the i-th element of $\mathbf{C}_T$ and thus is the number of data points in T that are not larger than x_i.

Clearly, given a reference set R and a test set T, every unique subset $S \subseteq T$ corresponds to a unique cumulative vector $\mathbf{C}_S$ and a unique cumulative distribution function $F_{T\setminus S}$, and vice versa. Recall that, if a subset $T \setminus S$ and R pass the KS test, S is called a qualified h-subset, where $h = |S|$. Correspondingly, we call the cumulative vector $\mathbf{C}_S$ a qualified h-cumulative vector.

4.3 Existence of Qualified h-Cumulative Vectors

For a given h $(1 \le h \le |T|)$, can we quickly determine whether there exists a qualified h-cumulative vector and thus a qualified h-subset? Before we state the major result, we need the following.

Lemma 1. *Given a reference set R and a test set T, for $S \subset T$, $\mathbf{C}_S = \langle c_0, c_1, \ldots, c_q \rangle$ is a qualified cumulative vector if and only if, for each i $(1 \le i \le q)$, the following two inequalities hold.*

$$\max(\lceil \Gamma(i,h) - \Omega(h) \rceil, h - m + \mathbf{C}_T[i], \mathbf{C}_S[i-1]) \le \mathbf{C}_S[i] \quad (2a)$$

$$\mathbf{C}_S[i] \le \min(\lfloor \Gamma(i,h) + \Omega(h) \rfloor, \mathbf{C}_T[i] - \mathbf{C}_T[i-1] + \mathbf{C}_S[i-1], h) \quad (2b)$$

where $\Omega(h) = c_\alpha\sqrt{m - h + \frac{(m-h)^2}{n}}$, $\Gamma(i,h) = \mathbf{C}_T[i] - \frac{m-h}{n}\mathbf{C}_R[i]$, and $\mathbf{C}_R$ and $\mathbf{C}_T$ are the cumulative vectors of R and T, respectively.

Proof. (Necessity) According to the definition of KS statistic in Equation 1, an h-subset S is qualified if and only if $\forall i$ $(1 \le i \le q)$, $|F_R(x_i) - F_{T\setminus S}(x_i)| \le c_\alpha\sqrt{\frac{n+m-h}{n*(m-h)}}$. Since $F_R(x_i) = \frac{\mathbf{C}_R[i]}{n}$ and $F_{T\setminus S}(x_i) = \frac{\mathbf{C}_T[i]-\mathbf{C}_S[i]}{m-h}$, we have $\left|\frac{\mathbf{C}_R[i]}{n} - \frac{\mathbf{C}_T[i]-\mathbf{C}_S[i]}{m-h}\right| \le c_\alpha\sqrt{\frac{n+m-h}{n*(m-h)}}$. After simplification, we have $\Gamma(i,h)-\Omega(h) \le \mathbf{C}_S[i] \le \Gamma(i,h)+\Omega(h)$. Since $\mathbf{C}_S[i]$ is a non-negative integer, we immediately have

$$\lceil \Gamma(i,h) - \Omega(h) \rceil \le \mathbf{C}_S[i] \le \lfloor \Gamma(i,h) + \Omega(h) \rfloor. \quad (3)$$

Since $h - \mathbf{C}_S[i]$ and $m - \mathbf{C}_T[i]$ are the numbers of data points in S and T that are larger than x_i, respectively, and $S \subset T$, $h - \mathbf{C}_S[i] \le m - \mathbf{C}_T[i]$ holds, that is, $h - m + \mathbf{C}_T[i] \le \mathbf{C}_S[i]$. Since $\mathbf{C}_S[i-1] \le \mathbf{C}_S[i]$, Equation 2a holds.

Since $\mathbf{C}_S[i] - \mathbf{C}_S[i-1]$ and $\mathbf{C}_T[i] - \mathbf{C}_T[i-1]$ are the numbers of times x_i appears in S and T, respectively, and $S \subset T$, $\mathbf{C}_S[i] - \mathbf{C}_S[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$, that is, $\mathbf{C}_S[i] \le \mathbf{C}_S[i-1] + \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Using the righthand side of Equation 3 and $\mathbf{C}_S[i] \le h$ by definition, Equation 2b holds.

(Sufficiency) For any h-cumulative vector $\mathbf{C}_S$ that satisfies Equations 2a and 2b, we construct a set S such that for each i $(1 \le i \le q)$, data point x_i appears in S $(\mathbf{C}_S[i] - \mathbf{C}_S[i-1])$ times. Since $\mathbf{C}_S[i]$ and $\mathbf{C}_S[i-1]$ satisfy the inequality $\mathbf{C}_S[i] - \mathbf{C}_S[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$, the number of times x_i appearing in S is smaller than or equal to the number of times x_i appearing in T. Plugging $\mathbf{C}_T[q] = m$ into Equation 2a, we have $h \le \mathbf{C}_S[q]$. Since $\mathbf{C}_S[q]$ also satisfies Equation 2b, we have $h = \mathbf{C}_S[q]$. From the way that S is constructed, we know that S has $\mathbf{C}_S[q]$ elements. Therefore, S is an h-subset of T.

We show that S is a qualified h-subset. Since $\mathbf{C}_S$ satisfies Equation 2, for each i $(1 \le i \le q)$, $\mathbf{C}_S[i]$ satisfies Equation 3. Plugging Equation 3 into $F_{T\setminus S}$, we have $\forall i (1 \le i \le q)$, $|F_R(x_i) - F_{T\setminus S}(x_i)| \le c_\alpha\sqrt{\frac{n+m-h}{n*(m-h)}}$. This means R and $T \setminus S$ can pass the KS test, thus S is a qualified h-subset of T and $\mathbf{C}_S$ is a qualified h-cumulative vector. The sufficiency follows. □

Lemma 1 transforms conducting the KS test to checking Equations 2a and 2b. Given h ($1 \le h \le m-1$), Equations 2a and 2b recursively give a lower bound and an upper bound of each element $\mathbf{C}[i]$ ($1 \le i \le q$) of an h-cumulative vector $\mathbf{C}$, respectively. The lower bound and the upper bound of $\mathbf{C}[i]$ depend on the lower bound and the upper bound of $\mathbf{C}[i-1]$, respectively.

Denote by l_i^h and u_i^h the lower bound and the upper bound of $\mathbf{C}[i]$ in any qualified h-cumulative vector $\mathbf{C}$. We compute l_1^h and u_1^h by plugging $\mathbf{C}[0] = 0$ into Equations 2a and 2b. Then, we plug $\mathbf{C}[1] = l_1^h$ into Equation 2a and $\mathbf{C}[1] = u_1^h$ into Equation 2b to compute the lower bound l_2^h and the upper bound u_2^h of $\mathbf{C}[2]$, respectively. By iteratively plugging $\mathbf{C}[i-1] = l_{i-1}^h$ into Equation 2a and $\mathbf{C}[i-1] = u_{i-1}^h$ into Equation 2b, we can compute the lower bound and the upper bound of every $\mathbf{C}[i]$ of qualified h-cumulative vectors $\mathbf{C}$. The closed form formulae of l_i^h and u_i^h ($1 \le i \le q$) are

$$l_i^h = \max(\lceil M(i,h) - \Omega(h) \rceil, h - m + \mathbf{C}_T[i], 0) \tag{4a}$$

$$u_i^h = \min(\lfloor \Gamma(i,h) + \Omega(h) \rfloor, \mathbf{C}_T[i], h) \tag{4b}$$

where $M(i,h) = \max_{j=1}^{i}\{\Gamma(j,h)\}$. We define $l_0^h = u_0^h = 0$, as $\mathbf{C}[0] = 0$ is a constant.

Given the lower bounds l_i^h and the upper bounds u_i^h of the element in any qualified h-cumulative vectors, if for each i ($1 \le i \le q$), $l_i^h \le u_i^h$, we can construct an h-cumulative vector $\mathbf{C}$ by selecting each element $\mathbf{C}[i]$ from $[l_i^h, u_i^h]$. Based on this intuition, we use the lower bounds and the upper bounds of $\mathbf{C}[1], \mathbf{C}[2], \ldots, \mathbf{C}[q]$ to derive a sufficient and necessary condition for the existence of a qualified h-cumulative vector $\mathbf{C}$ as follows.

Theorem 1. *Given the KS test with a reference set R and a test set T, for h ($1 \le h \le m-1$), there exists a qualified h-cumulative vector if and only if for each i ($1 \le i \le q$), $l_i^h \le u_i^h$.*

Proof. (Necessity) Since l_i^h and u_i^h are the lower bound and the upper bound of $\mathbf{C}[i]$, respectively, the necessity is straightforward.

(Sufficiency) Assuming for each i ($1 \le i \le q$), $l_i^h \le u_i^h$, we construct a qualified h-cumulative vector $\mathbf{C}$ as follows. We start by setting $\mathbf{C}[q] = u_q$, and then for i iterating from q to 1, we choose an integer $\mathbf{C}[i-1]$ from $[l_{i-1}^h, u_{i-1}^h]$, such that $0 \le \mathbf{C}[i] - \mathbf{C}[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$.

Now we show that such an integer $\mathbf{C}[i-1]$ always exists. Since l_i^h is derived by setting $\mathbf{C}[i-1] = l_{i-1}^h$ in Equation 2a and u_i^h is derived by setting $\mathbf{C}[i-1] = u_{i-1}^h$ in Equation 2b, we have $l_{i-1}^h \le l_i^h$ and $u_i^h \le u_{i-1}^h + \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Since $l_i^h \le \mathbf{C}[i] \le u_i^h$, we have $l_{i-1}^h \le \mathbf{C}[i] \le u_{i-1}^h + \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Since $l_{i-1}^h \le u_{i-1}^h$ and they are integers, there exists an integer $\mathbf{C}[i-1] \in [l_{i-1}^h, u_{i-1}^h]$, such that $0 \le \mathbf{C}[i] - \mathbf{C}[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Thus, an h-cumulative vector $\mathbf{C}$ can be constructed by iteratively applying the above operations to set up elements in $\mathbf{C}$.

Last, we prove that $\mathbf{C}$ is a qualified h-cumulative vector by showing that for each i ($1 \le i \le q$), $\mathbf{C}[i]$ satisfies Equations 2a and 2b. According to the definition of an h-cumulative vector, we have $\mathbf{C}[i-1] \le \mathbf{C}[i]$. By Equation 4a, we have $h - m + \mathbf{C}_T[i] \le l_i^h$ and $\lceil \Gamma(i,h) - \Omega(h) \rceil \le l_i^h$. Since $l_i^h \le \mathbf{C}[i]$, $\mathbf{C}[i]$ satisfies Equation 2a. By Equation 4b, we have $u_i^h \le h$ and $u_i^h \le \lfloor \Gamma(i,h) + \Omega(h) \rfloor$. According to how $\mathbf{C}[i-1]$ is selected, $\mathbf{C}[i]$ and $\mathbf{C}[i-1]$ satisfy $\mathbf{C}[i] - \mathbf{C}[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Since $\mathbf{C}[i] \le u_i^h$, $\mathbf{C}[i]$ satisfies Equation 2b. The sufficiency follows Lemma 1 immediately. □

According to Theorem 1, we can efficiently check the existence of a qualified h-cumulative vector by checking the q pairs of lower bounds and upper bounds, $(l_1^h, u_1^h), \ldots, (l_q^h, u_q^h)$. Each pair of bounds can be computed and checked in $O(1)$ time. Since $q \le n+m$, the time complexity of checking the existence of a qualified h-cumulative vector is $O(n+m)$.

Since the existence of a qualified h-cumulative vector is equivalent to the existence of a qualified h-subset, we can tackle the first efficiency bottleneck by checking the q pairs of lower bounds and upper bounds. This reduces the time complexity of checking the existence of a qualified h-subset from $O(\binom{m}{h}(m+n-h)\log(m+n-h))$ to $O(n+m)$.

To find the size of explanations, for each subset size h ($1 \le h \le m-1$), we need to apply Theorem 1 to check the existence of a qualified h-cumulative vector. Therefore, the overall time complexity of finding the size of explanations is $O(m(m+n))$. Next, we further reduce the time complexity to $O((m+n)\log(m) + (m+n)(k-\hat{k}))$, where $\hat{k}$ is a lower bound on the size of explanations k.

Example 4. One can verify that the reference set and the test set in Example 3 fail the KS test with significance level $\alpha = 0.3$. When $h = 1$, the lower bound $l_2^h = 2$ and the upper bound $u_2^h = 1$. As $l_2^h > u_2^h$, by Theorem 1, there does not exist a qualified 1-cumulative vector. When $h = 2$, $(l_1^h, u_1^h) = (0,1)$, $(l_2^h, u_2^h) = (1,2)$, $(l_3^h, u_3^h) = (1,2)$ and $(l_4^h, u_4^h) = (1,2)$. By Theorem 1, there exists a qualified 2-cumulative vector and thus a qualified 2-subset. Since the smallest size of a qualified subset is 2, the explanation size $k = 2$.

4.4 Finding a Lower Bound on Explanation Size by Binary Search

To tackle the second efficiency bottleneck, in this subsection, we develop a technique to find a lower bound on the size of explanations in $O((m+n)\log m)$ time. Using this technique, to find the size of explanations, we only need to check the subset sizes that are larger than or equal to the lower bound.

To reduce the number of subset sizes to be checked, we develop a necessary condition for the existence of a qualified h-cumulative vector with respect to h. The necessary condition is obtained by relaxing the sufficient and necessary condition stated in Theorem 1. The necessary condition has a nice monotonicity with respect to h. If an integer h ($1 \le h \le m-2$) satisfies the condition, all integers from $h+1$ to $m-1$ also satisfy the necessary condition. This is because the right hand side of each inequality in Equation 5 increases faster than its left hand side as h increases. Thus, we can leverage this property to find a lower bound $\hat{k}$ of the explanation size k by a binary search in $O((m+n)\log m)$ time. The lower bound reduces the search range of k from $[1,k]$ to $[\hat{k},k]$. This helps us further reduce the complexity of phase 1 in MOCHE from $O(m(n+m))$ to $O((n+m)\log(m) + (k-\hat{k})(n+m))$.

Theorem 2. *Given the KS test with a reference set R and a test set T, for h ($1 \le h \le m-1$), there exists a qualified h-cumulative vector*

only if for each i $(1 \le i \le q)$, the following holds.

$$0 \le \lfloor \Gamma(i,h) + \Omega(h) \rfloor \tag{5a}$$

$$\lceil M(i,h) - \Omega(h) \rceil \le h \tag{5b}$$

$$M(i,h) - \Omega(h) \le \Gamma(i,h) + \Omega(h) \tag{5c}$$

Moreover, if Equation 5 holds for $h > 0$, then it also holds for $h+1$.

Proof. We first prove the necessary condition. Since there exists a qualified h-cumulative vector, by Theorem 1, for each i $(1 \le i \le q)$, $l_i^h \le u_i^h$. l_i^h and u_i^h are the maximum and the minimum of the three terms in Equations 4a and 4b, respectively. Thus, every term in u_i^h is larger than or equal to every term in l_i^h. Therefore, we immediately have Equations 5a and 5b, as well as the following.

$$\lceil M(i,h) - \Omega(h) \rceil \le \lfloor \Gamma(i,h) + \Omega(h) \rfloor \tag{6}$$

Since $M(i,h) - \Omega(h) \le \lceil M(i,h) - \Omega(h) \rceil$ and $\lfloor \Gamma(i,h) + \Omega(h) \rfloor \le \Gamma(i,h) + \Omega(h)$, Equation 5c follows Equation 6 immediately.

Next, we prove the monotonicity of the necessary condition with respect to h. For each inequality in Equation 5, we show that for each i $(1 \le i \le q)$, if a size h $(1 \le h \le m-2)$ satisfies the inequality, the size $h+1$ also satisfies the inequality.

Equation 5a: Plugging the definitions of $\Gamma(i,h)$ and $\Omega(h)$ into Equation 5a, the inequality can be simplified to $\frac{C_T[i]}{m-h} - \frac{C_R[i]}{n} \ge -c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}}$. Since we have $-c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{m}} > -c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{m}}$ and $\frac{C_T[i]}{m-h-1} \ge \frac{C_T[i]}{m-h}$, we can get $\frac{C_T[i]}{m-h-1} - \frac{C_R[i]}{n} > -c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, which can be simplified to $0 \le \lfloor \Gamma(i,h+1) + \Omega(h+1) \rfloor$.

Equation 5b: Plugging the definition of $M(i,h)$ into Equation 5b, we have $\lceil \Gamma(j,h) - \Omega(h) \rceil \le h$, for each integer j $(1 \le j \le i)$. Plugging the definitions of $\Gamma(j,h)$ and $\Omega(h)$ into the inequality, the inequality can be simplified to $\frac{C_T[j]-h}{m-h} - \frac{C_R[j]}{n} \le c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}}$. Since $c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} < c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$ and $\frac{C_T[j]-h-1}{m-h-1} < \frac{C_T[j]-h}{m-h}$, we have $\frac{C_T[j]-h-1}{m-h-1} - \frac{C_R[j]}{n} \le c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, which can be simplified to $\Gamma(j,h+1) - \Omega(h+1) \le h$. Since h is an integer, we immediately have $\lceil \Gamma(j,h+1) - \Omega(h+1) \rceil \le h$. Applying the definition of $M(i,h)$, we have $\lceil M(i,h+1) - \Omega(h+1) \rceil \le h$.

Equation 5c: According to the definition of $M(i,h)$, from Equation 5c, we have $\Gamma(j,h) - \Omega(h) \le \Gamma(i,h) + \Omega(h)$, for each integer j $(1 \le j \le i)$. Plugging the definitions of $\Omega(h)$ and $\Gamma(j,h)$ into the inequality, the inequality can be simplified to $-2c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} \le \frac{C_T[i]-C_T[j]}{m-h} - \frac{1}{n}(C_R[i] - C_R[j])$. Since $\frac{C_T[i]-C_T[j]}{m-h-1} > \frac{C_T[i]-C_T[j]}{m-h}$ and $-2c_\alpha\sqrt{\frac{1}{m-h} + \frac{1}{n}} > -2c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}}$, we have the inequality $-2c_\alpha\sqrt{\frac{1}{m-h-1} + \frac{1}{n}} < \frac{C_T[i]-C_T[j]}{m-h-1} - \frac{1}{n}(C_R[i] - C_R[j])$, which can be simplified to $\Gamma(j,h+1) - \Omega(h+1) \le \Gamma(i,h+1) + \Omega(h+1)$. Applying the definition of $M(i,h)$, we have $M(i,h+1) - \Omega(h+1) \le \Gamma(i,h+1) + \Omega(h+1)$. □

The smallest integer $\hat{k}$ that satisfies the necessary condition in Theorem 2 is a lower bound on the size k of the explanations. We do not need to check any h-subset smaller than $\hat{k}$, as they are guaranteed not to contain a qualified h-cumulative vector. Based on the monotonicity of Equation 5 with respect to h, we can apply binary search to find the smallest integer $\hat{k}$ that satisfies Theorem 2. For $h \in [1, m-1]$, it takes $O(n+m)$ time to verify the q groups of inequalities in Theorem 2, because $q \le n+m$. Therefore, the overall time complexity of finding $\hat{k}$ is $O((m+n)\log m)$. Once $\hat{k}$ is found, we iteratively use Theorem 1 to find the exact size of explanations. The overall time complexity of finding the exact size of explanation is $O((m+n)\log m + (m+n)(k-\hat{k}))$, where k is the exact size. In the worst case, $k - \hat{k} = O(m)$, and the complexity is still $O(m(m+n))$. However, as verified by our experiments, $k - \hat{k}$ is often a very small number and our technique can significantly improve the efficiency of searching the size of explanations.

Example 5. Consider the failed KS test in Example 4. We apply binary search to find the lower bound $\hat{k} \in [1,3]$. We start with $h = \lfloor (1+3)/2 \rfloor = 2$ and find that $h = 2$ satisfies Theorem 2. Thus, $\hat{k} \le 2$. Then, we search the left half $[1,2]$ and set $h = \lfloor (1+2)/2 \rfloor = 1$. As $\lceil M(1,h) - \Omega(h) \rceil = 2$, Equation 5b does not hold and thus $h = 1$ does not satisfy Theorem 2. This concludes that $\hat{k} = 2$.

5 GENERATING MOST COMPREHENSIBLE EXPLANATIONS

Given the size of explanations k, the brute force method takes $O(\binom{m}{k}(m+n-k)\log(m+n-k))$ time to find the most comprehensible explanation by enumerating the k-subsets of T. In this section, we develop a method to directly construct the most comprehensible explanation in $O(m(n+m))$ time without enumerating the k-subsets.

An h-subset $S \subset T$ is called an *h-partial explanation* if there exists an explanation that is a superset of S. When it is clear from the context, we also call S a *partial explanation* for short.

According to Definition 1, the most comprehensible explanation is the explanation that has the smallest lexicographical order. This property facilitates the design of our construction algorithm. Our algorithm scans the data points in T in the order of L and selects the first data point x_{i_1} that is in an explanation, that is, x_{i_1} is a 1-partial explanation. Since x_{i_1} is the first such data point in L, the most comprehensible explanation must contain x_{i_1}, otherwise we have the contradiction that the explanation containing x_{i_1} precedes the most comprehensible explanation in the lexicographical order. Then, the algorithm continues to scan the points after x_{i_1} in L, still in the order of L, and finds the next data point x_{i_2} such that $\{x_{i_1}, x_{i_2}\}$ are part of an explanation, that is, $\{x_{i_1}, x_{i_2}\}$ is a 2-partial explanation. Clearly, $\{x_{i_1}, x_{i_2}\}$ is part of the most comprehensive explanation. The search continues until k points are obtained, which is the most comprehensible explanation. The construction method is summarized in Algorithm 1.

Now, the remaining question is how we can determine whether an h-subset S is a partial explanation. We first establish that a subset S is a partial explanation if and only if there exists a qualified k-cumulative vector, which satisfies a small group of inequalities derived from S. Then, we introduce a sufficient and necessary condition for the existence of such a k-cumulative vector, which can be efficiently checked in $O(n+m)$ time.

Lemma 2. *Given the KS test with a reference set R and a test set T, for a subset $S \subset T$, S is a partial explanation if and only if there exists a qualified k-cumulative vector $\mathbf{C}$, such that the following inequality*

Algorithm 1: Find the most comprehensible explanation

Input: a reference set R, a test set T, a significance level α, a preference list L, the size of explanations k
Output: $I :=$ the most comprehensible explanation

```
Initialize I ← ∅
T ← sort the data points in T in the order of L
for i ← 1; i ≤ |T|; i++ do
    if I ∪ {T[i]} is a partial explanation then I ← I ∪ {T[i]};
    if |I| = k then return I;
end
```

holds for $1 \le i \le q$,

$$\mathbf{C}[i] - \mathbf{C}[i-1] \ge \mathbf{C}_S[i] - \mathbf{C}_S[i-1] \tag{7}$$

Proof. (Necessity) Since S is a partial explanation, by definition, there exists an explanation $\mathcal{I}$ such that $S \subseteq \mathcal{I}$. Denote by $\mathbf{C}$ the qualified k-cumulative vector of $\mathcal{I}$. For each i $(1 \le i \le q)$, since $\mathbf{C}[i] - \mathbf{C}[i-1]$ and $\mathbf{C}_S[i] - \mathbf{C}_S[i-1]$ are the numbers of times x_i appearing in $\mathcal{I}$ and S, respectively, and $S \subseteq \mathcal{I}$, $\mathbf{C}[i] - \mathbf{C}[i-1] \ge \mathbf{C}_S[i] - \mathbf{C}_S[i-1]$ holds. The necessity follows.

(Sufficiency) Assume a qualified k-cumulative vector $\mathbf{C}$ that satisfies Equation 7. Let $\mathcal{I}$ be the explanation corresponding to $\mathbf{C}$. For each data point $x_i \in S$, x_i appears $\mathbf{C}_S[i] - \mathbf{C}_S[i-1]$ times in S. Due to Equation 7, x_i appears in $\mathcal{I}$ the same or more number of times. Thus, $S \subseteq \mathcal{I}$ and S is a partial explanation. □

Next, we derive a sufficient and necessary condition for the existence of such a k-cumulative vector $\mathbf{C}$ by investigating the lower bound and the upper bound of each element $\mathbf{C}[i]$. Since $\mathbf{C}$ is a qualified k-cumulative vector, by Theorem 1, $l_i^k \le \mathbf{C}[i] \le u_i^k$. Equation 7 can be rewritten as $\mathbf{C}[i-1] \le \mathbf{C}[i] - \mathbf{C}_S[i] + \mathbf{C}_S[i-1]$. That is, the upper bound of $\mathbf{C}[i-1]$ dependents on the upper bound of $\mathbf{C}[i]$. Denote by $\bar{l}_i = l_i^k$ a lower bound of $\mathbf{C}[i]$ and by $\bar{u}_i$ an upper bound of $\mathbf{C}[i]$. Since $\mathbf{C}[i] \le \bar{u}_i$ and $\mathbf{C}[i-1] \le u_{i-1}^k$, about the upper bounds we have, for i $(1 \le i \le q)$,

$$\bar{u}_{i-1} = \min(u_{i-1}^k, \bar{u}_i - \mathbf{C}_S[i] + \mathbf{C}_S[i-1]). \tag{8}$$

Given the size of explanations k, we first compute u_i^k for each i $(1 \le i \le q)$ by Equation 4b. Then, we iteratively compute $\bar{u}_i$ for each i $(0 \le i \le q)$. We define $\bar{u}_q = u_q^k$ and plug $\bar{u}_q$ into Equation 8 to compute $\bar{u}_{q-1}$, and iteratively compute the upper bound of each $\mathbf{C}[i]$ of a qualified k-cumulative vector $\mathbf{C}$ that satisfies Equation 7.

Since $\bar{u}_i$ depends on u_i^k, once the size of explanations k is determined using the techniques developed in Section 4, we can compute the value $\bar{u}_i$. Based on a similar intuition as Theorem 1, we can use the lower bound $\bar{l}_i$ and the upper bound $\bar{u}_i$ to derive a sufficient and necessary condition for the existence of a qualified k-cumulative vector $\mathbf{C}$ that satisfies Equation 7 as stated in the following result, and thus decide whether an h-subset S is a partial explanation.

Theorem 3. *Given the KS test with a reference set R and a test set T, for a subset $S \subset T$, there exists a qualified k-cumulative vector $\mathbf{C}$ that satisfies Equation 7 if and only if for each i $(0 \le i \le q)$, $\bar{l}_i \le \bar{u}_i$.*

Proof. (Sufficiency) Given S, assume for each i $(0 \le i \le q)$, $\bar{l}_i \le \bar{u}_i$. We construct a k-cumulative vector $\mathbf{C}$ such that for each i $(0 \le i \le q)$, $\mathbf{C}[i] = \bar{u}_i$. We show that $\mathbf{C}$ is a qualified k-cumulative vector and also satisfies Equation 7.

We first prove that $\mathbf{C}$ is a qualified k-cumulative vector by showing that $\mathbf{C}[0] = 0$, and each $\mathbf{C}[i]$ $(1 \le i \le q)$ satisfies Equations 2a and 2b. Since $l_0^k = \bar{l}_0 \le \bar{u}_0 \le u_0^k = 0$, we have $\bar{l}_0 = \bar{u}_0 = 0$ and thus $\mathbf{C}[0] = 0$. Plugging $\mathbf{C}[i-1] = \bar{u}_{i-1}$ and $\mathbf{C}[i] = \bar{u}_i$ into Equation 8, we have $\mathbf{C}[i-1] \le \mathbf{C}[i]$. Since $\bar{l}_i = l_i^k \le \mathbf{C}[i]$, from Equation 4a, we have $\lceil \Gamma(i,h) - \Omega(h) \rceil \le \mathbf{C}[i]$ and $h - m + \mathbf{C}_T[i] \le \mathbf{C}[i]$. Therefore, $\mathbf{C}[i]$ satisfies Equation 2a.

Plugging $\mathbf{C}[i-1] = \bar{u}_{i-1}$ into Equation 8, the value of $\mathbf{C}[i-1]$ falls into one of the following two cases.

Case 1: $\mathbf{C}[i-1] = u_{i-1}^k$. As u_i^k is derived by plugging $\mathbf{C}[i-1] = u_{i-1}^k$ into Equation 2b, we have $u_i^k \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1] + u_{i-1}^k$. Since $\mathbf{C}[i] \le u_i^k$ and $\mathbf{C}[i-1] = u_{i-1}^k$, we have $\mathbf{C}[i] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1] + \mathbf{C}[i-1]$.

Case 2: $\mathbf{C}[i-1] = \bar{u}_i - \mathbf{C}_S[i] + \mathbf{C}_S[i-1]$. Since $S \subset T$, $\mathbf{C}_S[i] - \mathbf{C}_S[i-1] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1]$. Since $\mathbf{C}[i] = \bar{u}_i$, we have $\mathbf{C}[i] \le \mathbf{C}_T[i] - \mathbf{C}_T[i-1] + \mathbf{C}[i-1]$.

Since $\mathbf{C}[i] = \bar{u}_i \le u_i^k$, from Equation 4b, we have $\mathbf{C}[i] \le k$ and $\mathbf{C}[i] \le \lfloor \Gamma(i,k) + \Omega(k) \rfloor$. Therefore, $\mathbf{C}[i]$ satisfies Equation 2b. By Lemma 1, $\mathbf{C}$ is a qualified k-cumulative vector.

Since for each i $(0 \le i \le q)$, $\mathbf{C}[i] = \bar{u}_i$, from Equation 8, we can derive that every $\mathbf{C}[i]$ satisfies Equation 7.

(Necessity) Given S, assume a qualified k-cumulative vector $\mathbf{C}$ that satisfies Equation 7. Since $\bar{l}_i$ and $\bar{u}_i$ are the lower and the upper bound of $\mathbf{C}[i]$, respectively, the necessity follows immediately. □

Example 6. Consider the failed KS test in Example 4, where the size of explanations k is 2. Suppose a user provides a preference list $L = [t_4, t_3, t_2, t_1]$. We initialize the constructed explanation $I = \emptyset$ and scan the data points in T in the order of L. For the first scanned data point $t_4 = 20$, we check if $S = I \cup \{t_4\}$ is a partial explanation. By Equation 8, the upper bound $\bar{u}_3 = 1$. As the lower bound $\bar{l}_3 = l_3^k = 2 > \bar{u}_3$, by Theorem 3, S is not a partial explanation and thus t_4 is not in any explanations.

We repeat the same step for the second scanned data point $t_3 = 12$. When $S = \{t_3\}$, $(\bar{l}_0, \bar{u}_0) = (0,0)$, $(\bar{l}_1, \bar{u}_1) = (0,1)$, and $(\bar{l}_2, \bar{u}_2) = (\bar{l}_3, \bar{u}_3) = (\bar{l}_4, \bar{u}_4) = (2,2)$. By Theorem 3, S is a partial explanation and thus we add t_3 to I. The third scanned data point $t_2 = 13$ is added to I for the same reason. As the size of $I = \{t_3, t_2\}$ is equal to k, I is the most comprehensible explanation on the failed KS test.

Given an explanation size k and a subset S, it takes $O(m+n)$ time to verify the $q+1$ groups of inequalities in Theorem 3, because $q \le n+m$. Since for each data point t_i in T, we need to check whether $I \cup \{t_i\}$ is a partial explanation, where I is the partial explanation found so far, the overall time complexity of constructing the most comprehensible explanation is $O(m(n+m))$.

As shown in Section 4, it takes $O((m+n)\log(m)) + O((n+m)(k-\hat{k})) = O((m+n)(\log m + k - \hat{k}))$ time to identify the explanation size. In total, our method takes $O(m(n+m))$ time to find the most comprehensible explanation for a failed KS test.

6 EXPERIMENTS

In this section, we evaluate the effectiveness of most comprehensible counterfactual explanations, and the efficiency and scalability of MOCHE. We describe the datasets and the experiment settings

Table 1: Some statistics of the datasets.

Dataset	# Time series	Length
AWS	17	1,243 ~ 4,700
AD	6	1,538 ~ 1,624
TRF	7	1,127 ~ 2,500
TWT	10	15,831 ~ 15,902
KC	7	1,882 ~ 22,695
ART	6	4032

in Section 6.1. Counterfactual explanations on a failed KS test have two fundamental requirements, being small and reversing the failed KS test. In Section 6.2, we evaluate the size of our explanations. In Section 6.2.1, we evaluate whether our explanations can reverse failed KS tests. In Section 6.3, we investigate the effectiveness of our method. Last, in Section 6.4, we verify the efficiency and scalability of our proposed method.

6.1 Datasets and Experiment Settings

6.1.1 Dataset Construction. We conduct experiments using 6 univariate time series datasets in the Numenta Anomaly Benchmark (NAB) repository [34] and a COVID-19 dataset.

COVID-19 Data. The COVID-19 dataset [2] is described in Examples 1 and 2. The 10 age groups in the dataset are encoded from young to old by integers from 1 to 10. We use the cases reported in August and September 2020 to build the reference set and the test set, respectively. The KS test fails at significance level 0.05, which indicates that the infected cases in those two months unlikely follow the same distribution on age groups. In Section 6.3, as a case study we interpret the failed KS test to find the data points that may likely be relevant to the failure.

We use the population descending order of the HAs to generate the preference list L of data points in the test set. The data points from the same HAs are sorted arbitrarily. We obtain the populations of the HAs from the website of Statistics Canada [6].

Time Series Data. Each dataset in the NAB repository contains 6 to 10 time series and each time series contains 1,000 to 20,000 observations. For each time series, the ground truth labels of abnormal observations are available. The AWS server metrics (AWS) dataset contains the time series of the CPU Utilization, Network Bytes In, and Disk Read Bytes of an AWS server. The online advertisement clicks (AD) dataset contains the time series of online advertisement clicking rates and cost per thousand impressions. The freeway traffic (TRF) dataset contains the time series of occupancy, speed, and travel time of freeway traffics collected by specific sensors. The Tweets (TWT) dataset contains the time series of numbers of Twitter mentions of publicly-traded companies such as Google, IBM, and Apple. The miscellaneous known causes (KC) dataset contains the time series from multiple domains, including machine temperature, number of NYC taxi passengers, and CPU usage of an AWS server. The artificial (ART) dataset contains the artificially-generated time series with varying types of distribution drifts [30].

We run a sliding window W of size w to obtain the reference set, and use the window of the same size following W immediately without any overlap as the test set. The reference set and the test set are muti-sets consisting of the observation values in corresponding sliding windows. The KS test is conducted multiple times as the sliding windows run through a time series. A failed KS test indicates that the time series has a distribution drift [30]. We interpret the failed KS test to find the data points that are likely relevant to the failure.

The significance level of the KS test is always set to 0.05 following the convention in statistical testing. We use a variety of window sizes, including 100, 200, 300, 1,000, 1,500, and 2,000.

We apply a widely used time series outlier detection method, Spectral Residual [53] to automatically generate the preference lists L of data points in the test sets. This preference list L reflects a user's domain knowledge about data abnormality. Data points with larger outlying scores are ranked higher in L. The data points with the same outlying scores are sorted randomly. We use the published Python codes of Spectral Residual [1] with the default parameters.

6.1.2 Baselines. To the best of our knowledge, interpreting failed KS tests has not been studied in literature. To evaluate the performance of MOCHE (M for short in figures), we design six baselines.

Greedy (GRD for short) generates a counterfactual explanation $\mathcal{I}$ by greedily selecting the first l data points in L such that R and $T \setminus \mathcal{I}$ can pass the KS test. When the preference list is generated by an outlier detection method, Greedy can be regarded as an extension of the outlier detection method to interpret failed KS tests.

Extended-CornerSearch (CS for short) is extended from CornerSearch [21], a state-of-the-art L_0-norm adversarial attack method on image classifiers. CornerSearch generates adversarial images by randomly searching a small portion of the top-K important pixels of input images and masking them to 0 or 1. Although CornerSearch is not proposed to interpret failed KS tests, it may be extended to serve the purpose. The Extended-CornerSearch treats data points as pixels and perturbs the selected data points $\mathcal{I}$ by removing them from T. After applying each perturbations, it conducts the KS test on R and $T \setminus \mathcal{I}$ to check if $\mathcal{I}$ is an explanation, meaning passing the KS test so that the unchanged part is regarded as normal by classifiers.

Extended-GRACE (GRC for short) is a direct extension from GRACE [36], the state-of-the-art counterfactual explanation method on neural networks. To interpret a prediction on an input vector $\mathbf{x}$, GRACE perturbs the most important K features of $\mathbf{x}$, which are ranked by an external method, to change the prediction. GRACE only accepts vectors as inputs and generates explanations by minimizing a target classifier's prediction scores. Correspondingly, we extend GRACE to interpret failed KS tests by first accommodating the inconsistency between the inputs of GRACE and our problem through a mapping from an m-dimensional vector $\mathbf{x}$ to a subset $S \subseteq T$, where $m = |T|$. We project $\mathbf{x}$ to its nearest 0-1 vector and put the i-th data point t_i into S if the i-th element of the vector is 0. Next, we extend the objective function of GRACE to find explanations on failed KS tests by perturbing $\mathbf{x}$ to minimize $g(\mathbf{x}) = \sqrt{\frac{n*(m-|S|)}{n+(m-|S|)}} D(R, T \setminus S)$, where S is the set of data points picked by vector $\mathbf{x}$. Based on the definition of the KS test, S is an explanation on the failed KS test if $g(\mathbf{x})$ is smaller than the critical value c_α. Since $g(\mathbf{x})$ is not differentiable, we adopt the zeroth order optimization algorithm in [20] to solve the problem. We skip the entropy-based feature selection step used in GRACE, as it requires

access to training data of classifiers, which is not available in our problem setting.

Extended-D3 is extended from D3 [59], an outlier detection method on data streams. Given a set of historical data points X, a new coming data point x is detected as an outlier if x has a low probability density in X. As D3 is not designed for interpreting failed KS tests, we extend D3 to serve the purpose. The Extended-D3 selects the data points in T that have high probability densities in T and low probability densities in R. Specifically, denote by f_R and f_T the estimated probability density functions of R and T, respectively. Extended-D3 sorts the data points t_i in T in $\frac{f_T(t_i)}{f_R(t_i)}$ descending order. Then, it greedily selects the first l data points such that R and $T \setminus \mathcal{I}$ can pass the KS test. By default, f_R and f_T are learned using the same way as D3. For the COVID-19 dataset, as the data values are discrete, we use the empirical probability mass functions of R and T as f_R and f_T, respectively. As Extended-D3 cannot take user preferences as input, it cannot produce comprehensible explanations. When the context is clear, we call this baseline method **D3** for short.

Extended-STOMP (STMP for short) is extended directly from STOMP [65], a widely used anomalous subsequence detection algorithm on time series. Given a regular time series $\mathbf{N}$, a query time series $\mathbf{Q}$, and a subsequence length q, STOMP aims to detect anomalous subsequences of length q (each is called a q-subsequence) in $\mathbf{Q}$. STOMP applies z-normalization on each subsequence and detects subsequences with anomalous shapes [65].

We extend STOMP to interpret the failed KS tests conducted on the time series datasets. For a failed KS test, let $\mathbf{N}$ and $\mathbf{Q}$ be the corresponding time series segments of the reference set and the test set, respectively. Extended-STOMP sorts the q-subsequences of $\mathbf{Q}$ by their anomalous scores in decreasing order. Then, the algorithm greedily selects the data points from the first l subsequences such that R and $T \setminus \mathcal{I}$ can pass the KS test. Same as D3, Extended-STOMP cannot produce comprehensible explanations.

Extended-Series2Graph (S2G for short) is extended from Series2Graph [13], a state-of-the-art anomalous subsequence detection method on time series. Series2Graph takes the same input as STOMP. It detects q-subsequences of $\mathbf{Q}$ with anomalous shapes by learning a subsequence embedding model. We extend Series2Graph to interpret failed KS tests in the same way as Extended-STOMP. Same as D3, S2G cannot produce comprehensible explanations.

6.1.3 Parameter Settings. By default the significance level in all KS tests is fixed to 0.05.

We adopt the same parameter setting used in [21] for CS. For GRC, we set $K = 100$ to be consistent with mcCS and set the remaining parameters to the same as [36]. We use the same parameters as [20] for the zeroth optimization algorithm used in GRC. The parameters of D3 are set to the same as [59]. We test STMP and S2G with a variety of q values, including $5\%|T|$, $10\%|T|$, $20\%|T|$, and $40\%|T|$. Since $q = 5\%|T|$ outperforms the other settings on producing small explanations, we choose $q = 5\%|T|$ for STMP and S2G in all experiments. The remaining parameters of STMP and S2G are set to the same as [65] and [13], respectively.

We use the published Python codes of Series2Graph [5] and STOMP [4]. The remaining algorithms are implemented in Python. All experiments are conducted on a server with two Xeon(R) Silver 4114 CPUs (2.20GHz), four Tesla P40 GPUs, 400GB main memory, and a 1.6TB SSD running Centos 7 OS. Our source code is published on GitHub https://github.com/research0610/MOCHE.

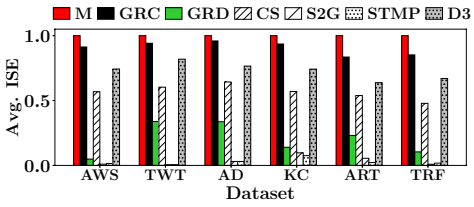

Figure 2: The average ISE, the larger the better.

Since CS and GRC cannot return explanations for all failed KS tests in 24 hours, for each combination of time series and window size, we uniformly sample 10 failed KS tests, where the test sets contain the corresponding ground truth of abnormal observations. We conduct all experiments on the sampled 2,690 failed KS tests.

6.2 Conciseness

Small explanations help users focus on predominant factors in a decision [62]. Therefore, being small is a key preference on counterfactual explanations [36, 41].

We design a binary variable *Is-Smallest-Explanation (ISE)* in the performance study of the compared methods in producing small counterfactual explanations. For the explanations produced by all methods on the same failed KS test, the ISE of the smallest explanation is 1, and 0 for the other explanations.

We evaluate MOCHE and the six baseline methods in ISE on the failed KS tests of the time series datasets. GRC and CS cannot find counterfactual explanations for some failed KS tests. To fairly compare the methods, among the 2,690 failed KS tests in those datasets, in this experiment we only consider the 847 ones (31.4%) where all methods can generate counterfactual explanations. Figure 2 shows the average ISE of all explanations.

STMP and S2G perform poorly. They choose some data points from the outlying subsequences as explanations on a failed KS test. Their outlying scores are computed on normalized subsequences, whose original distributions are changed [13]. Therefore, the data points from the outlying subsequences cannot explain why the KS test detects the distribution change between the reference set and the test set, and thus cannot find the smallest explanations on most of the failed KS tests.

D3 outperforms STMP and S2G. It interprets by comparing the estimated distributions of the reference set and the test set. However, limited by the approximation quality of its distribution estimator, D3 cannot always produce the smallest explanations.

GRD and CS do not perform well. Both methods generate explanations by taking the first several data points in the preference lists until the picked data points reverse the KS tests. However, since the preference lists are generated by a method independent from the KS test, some data points that are not highly relevant to the failure of the KS test may still be ranked high in the preference lists. As a result, those two methods may select many data points irrelevant to the failure of the KS test and lead to unnecessarily large explanations.

Table 2: The reverse factor, the larger the better.

Method	AWS	TWT	AD	KC	ART	TRF
CS	0.85	0.92	0.93	0.90	0.85	0.80
GRC	0.76	0.70	0.78	0.59	0.70	0.82

As a counterfactual explanation method, GRC generates explanations by solving an optimization problem, which allows it to re-rank the data points based on their effects on the KS tests. Therefore, as shown in Figure 2, GRC finds smaller explanations than the other baseline methods. However, GRC still cannot guarantee to find the smallest explanations all the time, because its objective function is non-differentiable and hard to minimize.

MOCHE guarantees to find the smallest explanation and thus has ISE value 1 in all cases.

6.2.1 Contrastivity. A counterfactual explanation on a failed KS test should reverse the failed KS test into a passed one. In this subsection, we quantitatively evaluate the performance of the methods in providing explanations that can reverse failed KS tests.

To measure the capability of a method, we use the *reverse factor (RF)*, which is the ratio $RF = \frac{\text{Number of reversed failed KS tests}}{\text{Total number of failed KS tests}}$. The larger the RF value, the stronger capability a method reversing failed KS tests.

Since GRC and CS cannot produce all results within 24 hours on some data sets, we constrain the two methods to only generate explanations using the top-100 ranked data points in the preference lists L. In other words, GRC and CS abort if a failed KS test does not have a counterfactual explanation that is a subset of the top-100 data points. To compare the methods in a fair manner, in this experiment, we only count the 1,293 (48.1%) among the 2,690 failed KS tests where GRC and CS do not abort. Table 2 shows the RF of CS and GRC. The RF values of the other methods are always 1 on all datasets.

CS and GRC cannot find counterfactual explanations for a large number of failed KS tests. The non-differential objective function of GRC is hard to optimize. CS likely samples the top-ranked data points in a preference list [21]. If the top-ranked data points are not relevant to the failure of a KS test, CS cannot reverse the failed KS test within its optimization steps. One may improve the RF of GRC and CS by more optimization steps. However, as to be shown in Section 6.4, these two methods are very slow, and more optimization steps make them even slower. The other baselines have a good RF. However, as shown in Figure 2, those methods tend to find large subsets of the test set as explanations, which are not informative [36, 41]

The RF of MOCHE is 1 on all datasets. MOCHE guarantees to produce the most comprehensible counterfactual explanations.

6.3 Effectiveness and Case Study

A counterfactual explanation on a failed KS test is effective if removing the explanation from the test set could make the distributions of the reference set and the test set similar. In this subsection, we first quantitatively evaluate the effectiveness of the explanations generated by all methods. Then, we conduct a case study to illustrate the effectiveness of the most comprehensible explanations.

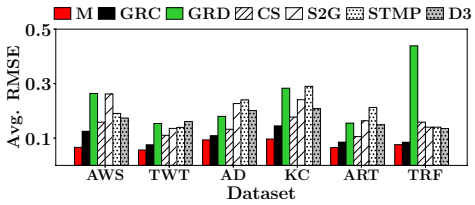

Figure 3: The average RMSE, the smaller the better.

We evaluate the effectiveness of an explanation $\mathcal{I}$ using the *root mean square error (RMSE)* between the empirical cumulative functions of R and $T' = T \setminus \mathcal{I}$. The RMSE is defined as $\text{RMSE} = \sqrt{\frac{\sum_{x \in R \cup T'} (F_R(x) - F_{T'}(x))^2}{|R \cup T'|}}$, where F_R and $F_{T'}$ are the empirical cumulative functions of R and T', respectively. A small RMSE value indicates the distributions of R and T' are similar and the explanation $\mathcal{I}$ is good.

We evaluate MOCHE and all baseline methods in RMSE on the failed KS tests of the time series datasets. Figure 3 shows the average RMSE on each data set for each method.

GRC performs best among all baselines, as it generates explanations on failed KS tests by minimizing the largest absolute difference between F_R and $F_{T'}$. However, as its non-differential objective function is hard to minimize, it cannot find a good solution to its optimization problem. As discussed in Section 6.2, the explanations generated by the other baselines include many data points that are irrelevant to the failure of the KS tests. Therefore, they tend to have large RMSE. MOCHE outperforms all baselines. It guarantees to produce the smallest explanations that can reverse the failed KS tests, and thus can guarantee the similarity of the distributions.

Let us examine the explanations on the failed KS test conducted on the COVID-19 dataset. The two sets are shown as histograms in Figure 1a. Figures 4a, 4b, and 4c show the histograms of the explanations produced by MOCHE, GRD and D3, respectively. In this case, among all baselines GRD and D3 produce the smallest explanations that can reverse the failed KS test. The empirical cumulative functions of the reference set, and the test set after removing each explanation are shown in Figure 4d.

Figure 4a shows that MOCHE selects some data points in the middle and senior age groups. MOCHE mainly selects the data points from age groups that have larger relative frequencies in the test set than in the reference set. As shown in Figure 1b, MOCHE only selects some data points from FHA (Fraser HA), the HA with the largest population. In September, the number of infected middle-aged and senior people in the HA increased dramatically, according to the news reports and analysis in media. As shown in Figure 4d, after removing the explanation, the distribution of the test set is most similar to that of the reference set. The results here match the real situation well.

In terms of explanation size, MOCHE, GRD, and D3 select 291 (8.6%$|T|$), 3,115 (92.3%$|T|$), and 3,370 (99.9%$|T|$) points in their explanations, respectively. GRD and D3 select almost all data points in the test set. Such explanations are not informative at all. Please note that STMP and S2G cannot interpret the failed KS test, as they can only work on time series.

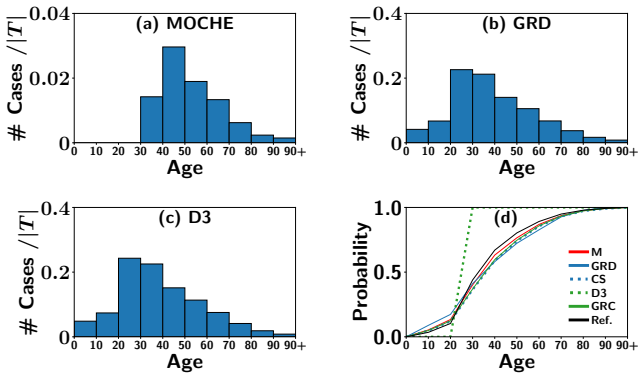

Figure 4: The explanations on the failed KS test conducted on the COVID-19 dataset. (d) shows the empirical cumulative functions of the reference set, and the test set after removing the explanations produced by different methods (best viewed in color).

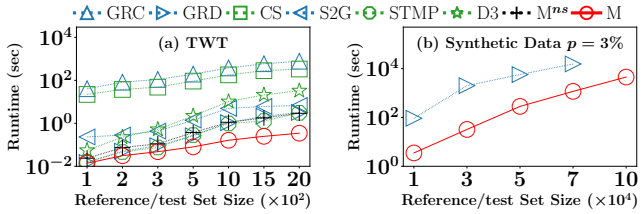

Figure 5: The runtime (plotted in logarithmic scale) on data set TWT and the synthetic dataset (best viewed in color).

6.4 Efficiency and Scalability

In this subsection, we report the runtime of all methods. In addition, to evaluate the effectiveness of our pruning techniques, we implement a lower-bound ablation MOCHE^{ns} by disabling the pruning using the lower bound of the explanation size (Section 4.4).

We vary the size of reference sets and test sets. As explained in Section 6.1, for a given reference/test set size, there are multiple failed KS tests. MOCHE constantly outperforms all baseline methods on all datasets. Limited by space, we only report the the average runtime of each method on the largest dataset TWT in Figure 5a. The runtime of all methods increases when the test sets become larger. MOCHE is 3 orders of magnitudes faster than GRC and CS.

The poor performance of all baseline methods is due to the cost of conducting huge numbers of KS tests. GRC needs to conduct $l \cdot m$ KS tests to find an explanation, where l is the number of optimization steps. According to the parameter settings in [36], in the worst case, GRC has to perform $l = 10,000$ steps.

CS has to generate a large number of samples to find an explanation, which takes a long time to verify. In the worst case, according to the parameter settings in [21], CS has to generate 150,000 random samples. GRD and D3 need to conduct the KS test after removing each data point. Since our estimated lower bound on the size k of explanations effectively reduces the search range of k, MOCHE interprets failed KS tests faster than MOCHE^{ns}.

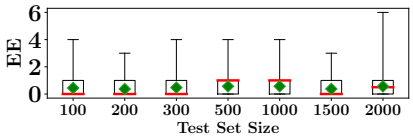

Figure 6: The estimation errors (EE) of the explanation size.

To comprehensively evaluate the efficiency, we construct large synthetic datasets to further compare the performance of MOCHE and GRD, the most efficient baseline method that can produce comprehensible explanations. Following the idea in [30], we first generate the reference set R and the test set T with the same size w from the normal distribution. Then, we replace a p fraction of T by data points sampled from a uniform distribution between $[-7, 7]$, such that R and T fail the KS test with significance level $\alpha = 0.05$. We use a variety of w and p values. We interpret the failed KS tests with randomly generated preference lists L. Our method constantly outperforms GRD on all these experiments. Limited by space, we only report the runtime on the synthetic dataset with $p = 3\%$ in Figure 5b. When $w = 10^5$, GRD cannot stop within 2 hours. MOCHE is at least 10 times faster than the most efficient baseline method.

To investigate the tightness of the lower bound $\hat{k}$ on the explanation size k, we also report the *estimation error (EE)* defined by $k - \hat{k}$. A small value of EE indicates that our estimated lower bound is tight. Figure 6 shows the results with respect to different sizes of test sets by box plot [63]. Each bar in the figure shows EE on the KS tests with a specific size of test sets. The upper and lower edges of a box show the first and third quartiles of the estimation errors, respectively. The upper and lower ends of an error bar show the maximum and minimum EE, respectively. The red line segment in a box and the green diamond marker show the median and the mean of the estimation errors, respectively.

For more than 25% of the failed KS tests, our estimated lower bound $\hat{k}$ is equal to the true value of k. For more than 75% of the failed KS tests, the estimation errors are up to 1. In the worst case (a KS test with 2,000 data points in the test set), our estimation error is only 6, much smaller than the test set size. Besides, we observe that when the test sets become larger, the average value of estimation errors is always smaller than 1. The results seem to suggest that estimation errors may be treated as a constant in practice. This result is consistent with our observation in Figure 5 that MOCHE is more efficient than MOCHE^{ns}.

7 CONCLUSIONS

In this paper, we tackle the novel problem of producing counterfactual explanations on failed KS tests. We propose the notion of most comprehensible counterfactual explanation, and develop a two-phase algorithm, MOCHE, which guarantees to find the most comprehensible explanation fast. We report extensive experiments demonstrating the superior capability of MOCHE in efficiently interpreting failed KS tests. As future work, we plan to extend MOCHE to interpret failed KS tests conducted on multidimensional data points [24, 51].

REFERENCES

[1] 2021. Alibi Detect. https://github.com/SeldonIO/alibi-detect. Accessed: 2021-05-11.

[2] 2021. BC COVID-19 Data. http://www.bccdc.ca/health-info/diseases-conditions/covid-19/data. Accessed: 2021-05-11.

[3] 2021. Health Authority Boundaries. https://catalogue.data.gov.bc.ca/dataset/health-authority-boundaries. Accessed: 2021-05-11.

[4] 2021. Matrix Profile. https://matrixprofile.org. Accessed: 2021-05-11.

[5] 2021. Series2Graph. http://helios.mi.parisdescartes.fr/~themisp/series2graph/. Accessed: 2021-05-11.

[6] 2021. Statistics Canada. https://www12.statcan.gc.ca/census-recensement/2016/dp-pd/prof/index.cfm. Accessed: 2021-05-11.

[7] Charu C. Aggarwal. 2013. *Outlier Analysis*. Springer. https://doi.org/10.1007/978-1-4614-6396-2

[8] Rakesh Agrawal, Jerry Kiernan, Ramakrishnan Srikant, and Yirong Xu. 2004. Order-Preserving Encryption for Numeric Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, Paris, France, June 13-18, 2004*, Gerhard Weikum, Arnd Christian König, and Stefan Deßloch (Eds.). ACM, 563–574. https://doi.org/10.1145/1007568.1007632

[9] Arjun R. Akula, Shuai Wang, and Song-Chun Zhu. 2020. CoCoX: Generating Conceptual and Counterfactual Explanations via Fault-Lines. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2594–2601. https://aaai.org/ojs/index.php/AAAI/article/view/5643

[10] Fabrizio Angiulli and Clara Pizzuti. 2002. Fast Outlier Detection in High Dimensional Spaces. In *Principles of Data Mining and Knowledge Discovery, 6th European Conference, PKDD 2002, Helsinki, Finland, August 19-23, 2002, Proceedings (Lecture Notes in Computer Science)*, Tapio Elomaa, Heikki Mannila, and Hannu Toivonen (Eds.), Vol. 2431. Springer, 15–26. https://doi.org/10.1007/3-540-45681-3_2

[11] André Artelt and Barbara Hammer. 2020. Efficient computation of counterfactual explanations of LVQ models. In *28th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2020, Bruges, Belgium, October 2-4, 2020*. 19–24. https://www.esann.org/sites/default/files/proceedings/2020/ES2020-55.pdf

[12] Idir Benouaret, Sihem Amer-Yahia, and Senjuti Basu Roy. 2019. An Efficient Greedy Algorithm for Sequence Recommendation. In *Database and Expert Systems Applications - 30th International Conference, DEXA 2019, Linz, Austria, August 26-29, 2019, Proceedings, Part I (Lecture Notes in Computer Science)*, Sven Hartmann, Josef Küng, Sharma Chakravarthy, Gabriele Anderst-Kotsis, A Min Tjoa, and Ismail Khalil (Eds.), Vol. 11706. Springer, 314–326. https://doi.org/10.1007/978-3-030-27615-7_24

[13] Paul Boniol and Themis Palpanas. 2020. Series2graph: Graph-based subsequence anomaly detection for time series. *Proceedings of the VLDB Endowment* 13, 12 (2020), 1821–1834.

[14] Wieland Brendel, Jonas Rauber, and Matthias Bethge. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=SyZI0GWCZ

[15] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*, Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein (Eds.). ACM, 93–104. https://doi.org/10.1145/342009.335388

[16] Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian K. Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, Tegan Maharaj, Pang Wei Koh, Sara Hooker, Jade Leung, Andrew Trask, Emma Bluemke, Jonathan Lebensbold, Cullen O'Keefe, Mark Koren, Theo Ryffel, J. B. Rubinovitz, Tamay Besiroglu, Federica Carugati, Jack Clark, Peter Eckersley, Sarah de Haas, Maritza Johnson, Ben Laurie, Alex Ingerman, Igor Krawczuk, Amanda Askell, Rosario Cammarota, Andrew Lohn, David Krueger, Charlotte Stix, Peter Henderson, Logan Graham, Carina Prunkl, Bianca Martin, Elizabeth Seger, Noa Zilberman, Seán Ó hÉigeartaigh, Frens Kroeger, Girish Sastry, Rebecca Kagan, Adrian Weller, Brian Tse, Elizabeth Barnes, Allan Dafoe, Paul Scharre, Ariel Herbert-Voss, Martijn Rasser, Shagun Sodhani, Carrick Flynn, Thomas Krendl Gilbert, Lisa Dyer, Saif Khan, Yoshua Bengio, and Markus Anderljung. 2020. Toward Trustworthy AI Development: Mechanisms for Supporting Verifiable Claims. *CoRR* abs/2004.07213 (2020). arXiv:2004.07213 https://arxiv.org/abs/2004.07213

[17] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. 2019. Machine learning interpretability: A survey on methods and metrics. *Electronics* 8, 8 (2019), 832.

[18] Hongsong Chen, Caixia Meng, Zhiguang Shan, Zhongchuan Fu, and Bharat K Bhargava. 2019. A novel Low-rate Denial of Service attack detection approach in ZigBee wireless sensor network by combining Hilbert-Huang Transformation and Trust Evaluation. *IEEE Access* 7 (2019), 32853–32866.

[19] Xuefeng Chen, Yifeng Zeng, Gao Cong, Shengchao Qin, Yanping Xiang, and Yuanshun Dai. 2015. On Information Coverage for Location Category Based Point-of-Interest Recommendation. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, Blai Bonet and Sven Koenig (Eds.). AAAI Press, 37–43. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9703

[20] Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2019. Query-Efficient Hard-label Black-box Attack: An Optimization-based Approach. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. https://openreview.net/forum?id=rJlk6iRqKX

[21] Francesco Croce and Matthias Hein. 2019. Sparse and Imperceivable Adversarial Attacks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 4723–4731. https://doi.org/10.1109/ICCV.2019.00482

[22] Zhiguo Ding and Minrui Fei. 2013. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes* 46, 20 (2013), 12–17.

[23] Denis Moreira dos Reis, Peter A. Flach, Stan Matwin, and Gustavo E. A. P. A. Batista. 2016. Fast Unsupervised Online Drift Detection Using Incremental Kolmogorov-Smirnov Test. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 1545–1554. https://doi.org/10.1145/2939672.2939836

[24] G. Fasano and A. Franceschini. 1987. A multidimensional version of the Kolmogorov–Smirnov test. *Monthly Notices of the Royal Astronomical Society* 225, 1 (03 1987), 155–170. https://doi.org/10.1093/mnras/225.1.155 arXiv:https://academic.oup.com/mnras/article-pdf/225/1/155/18522274/mnras225-0155.pdf

[25] Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 3449–3457. https://doi.org/10.1109/ICCV.2017.371

[26] Markus Goldstein and Andreas Dengel. 2012. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. *KI-2012: Poster and Demo Track* (2012), 59–63.

[27] Xiaoyi Gu, Leman Akoglu, and Alessandro Rinaldo. 2019. Statistical Analysis of Nearest Neighbor Methods for Anomaly Detection. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 10921–10931. https://proceedings.neurips.cc/paper/2019/hash/805163a0f0f128e473726ccda5f91bac-Abstract.html

[28] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. 2008. Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment* 1, 1 (2008), 102–114.

[29] Fabian Keller, Emmanuel Müller, and Klemens Böhm. 2012. HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. In *IEEE 28th International Conference on Data Engineering (ICDE 2012), Washington, DC, USA (Arlington, Virginia), 1-5 April, 2012*, Anastasios Kementsietsidis and Marcos Antonio Vaz Salles (Eds.). IEEE Computer Society, 1037–1048. https://doi.org/10.1109/ICDE.2012.88

[30] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. 2004. Detecting change in data streams. In *VLDB*, Vol. 4. Toronto, Canada, 180–191.

[31] Jerome Klotz. 1967. Asymptotic efficiency of the two sample Kolmogorov-Smirnov test. *J. Amer. Statist. Assoc.* 62, 319 (1967), 932–938.

[32] Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. 2017. Counterfactual fairness. In *Advances in neural information processing systems*. 4066–4076.

[33] Ashwin Lall. 2015. Data streaming algorithms for the Kolmogorov-Smirnov test. In *2015 IEEE International Conference on Big Data, Big Data 2015, Santa Clara, CA, USA, October 29 - November 1, 2015*. IEEE Computer Society, 95–104. https://doi.org/10.1109/BigData.2015.7363746

[34] Alexander Lavin and Subutai Ahmad. 2015. Evaluating Real-Time Anomaly Detection Algorithms–The Numenta Anomaly Benchmark. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 38–44.

[35] Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 157–166.

[36] Thai Le, Suhang Wang, and Dongwon Lee. 2020. GRACE: Generating Concise and Informative Contrastive Sample to Explain Neural Network Model's Prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 238–248. https://doi.org/10.1145/3394486.3403066

[37] Arnaud Van Looveren and Janis Klaise. 2019. Interpretable Counterfactual Explanations Guided by Prototypes. *CoRR* abs/1907.02584 (2019). arXiv:1907.02584 http://arxiv.org/abs/1907.02584

[38] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.

[39] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. 2019. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9087–9096.

[40] Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

[41] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, Adrienne Raglin, and Huan Liu. 2020. Causal Interpretability for Machine Learning-Problems, Methods and Evaluation. *ACM SIGKDD Explorations Newsletter* 22, 1 (2020), 18–33.

[42] Ramaravind K Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*. 607–617.

[43] Sigurd Kirkevold Næss. 2012. Application of the Kolmogorov-Smirnov test to CMB data: Is the universe really weakly random? *Astronomy & Astrophysics* 538 (2012), A17.

[44] Raymond S Nickerson. 1998. Confirmation bias: A ubiquitous phenomenon in many guises. *Review of general psychology* 2, 2 (1998), 175–220.

[45] Mila Nikolova. 2013. Description of the Minimizers of Least Squares Regularized with \ell_0-norm. Uniqueness of the Global Minimizer. *SIAM Journal on Imaging Sciences* 6, 2 (2013), 904–937.

[46] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B Gibbons, and Christos Faloutsos. 2003. Loci: Fast outlier detection using the local correlation integral. In *Proceedings 19th international conference on data engineering (Cat. No. 03CH37405)*. IEEE, 315–326.

[47] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*. 506–519.

[48] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *CoRR* abs/1605.07277 (2016). arXiv:1605.07277 http://arxiv.org/abs/1605.07277

[49] Fábio Pinto, Marco O. P. Sampaio, and Pedro Bizarro. 2019. Automatic Model Monitoring for Data Streams. *CoRR* abs/1908.04240 (2019). arXiv:1908.04240 http://arxiv.org/abs/1908.04240

[50] Neoklis Polyzotis, Martin Zinkevich, Sudip Roy, Eric Breck, and Steven Whang. 2019. Data validation for machine learning. *Proceedings of Machine Learning and Systems* 1 (2019), 334–347.

[51] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. 2019. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*. 1396–1408.

[52] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 427–438.

[53] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-Series Anomaly Detection Service at Microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3009–3017.

[54] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[55] Ron Rymon. 1992. Search through Systematic Set Enumeration. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning* (Cambridge, MA) *(KR'92)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 539–550.

[56] Ricardo Jorge Santos, Jorge Bernardino, and Marco Vieira. 2014. Approaches and challenges in database intrusion detection. *ACM Sigmod Record* 43, 3 (2014), 36–47.

[57] Sebastian Schelter, Tammo Rukat, and Felix Biessmann. 2020. Learning to Validate the Predictions of Black Box Classifiers on Unseen Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. 1289–1299.

[58] Kacper Sokol and Peter A. Flach. 2019. Counterfactual Explanations of Machine Learning Predictions: Opportunities and Challenges for AI Safety. In *Workshop on Artificial Intelligence Safety 2019 co-located with the Thirty-Third AAAI Conference on Artificial Intelligence 2019 (AAAI-19), Honolulu, Hawaii, January 27, 2019 (CEUR Workshop Proceedings)*, Huáscar Espinoza, Seán Ó hÉigeartaigh, Xiaowei Huang, José Hernández-Orallo, and Mauricio Castillo-Effen (Eds.), Vol. 2301. CEUR-WS.org. http://ceur-ws.org/Vol-2301/paper_20.pdf

[59] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos. 2006. Online Outlier Detection in Sensor Data Using Non-Parametric Models. In *Proceedings of the 32nd International Conference on Very Large Data Bases* (Seoul, Korea) *(VLDB '06)*. VLDB Endowment, 187–198.

[60] Sebastian Tschiatschek, Adish Singla, and Andreas Krause. 2017. Selecting Sequences of Items via Submodular Maximization.. In *AAAI*. 2667–2673.

[61] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harv. JL & Tech.* 31 (2017), 841.

[62] Danding Wang, Qian Yang, Ashraf Abdul, and Brian Y Lim. 2019. Designing theory-driven user-centric explainable AI. In *Proceedings of the 2019 CHI conference on human factors in computing systems*. 1–15.

[63] David F Williamson, Robert A Parker, and Juliette S Kendrick. 1989. The box plot: a simple visual method to interpret data. *Annals of internal medicine* 110, 11 (1989), 916–921.

[64] Min Xie, Laks VS Lakshmanan, and Peter T Wood. 2010. Breaking out of the box of recommendations: from items to packages. In *Proceedings of the fourth ACM conference on Recommender systems*. 151–158.

[65] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*. Ieee, 1317–1322.

[66] Shujian Yu, Xiaoyang Wang, and José C. Príncipe. 2018. Request-and-Reverify: Hierarchical Hypothesis Testing for Concept Drift Detection with Expensive Labels. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, Jérôme Lang (Ed.). ijcai.org, 3033–3039. https://doi.org/10.24963/ijcai.2018/421