# A Case-Based Reasoning Framework to Choose Trust Models for Different E-Marketplace Environments

**Athirai A. Irissappane** ATHIRAI001@E.NTU.EDU.SG
**Jie Zhang** ZHANGJ@NTU.EDU.SG
*School of Computer Engineering*
*Nanyang Technological University, Singapore*

## Abstract

The performance of trust models highly depend on the characteristics of the environments where they are applied. Thus, it becomes challenging to choose a suitable trust model for a given e-marketplace environment, especially when ground truth about the agent (buyer and seller) behavior is unknown (called *unknown environment*). We propose a case-based reasoning framework to choose suitable trust models for unknown environments, based on the intuition that if a trust model performs well in one environment, it will do so in another similar environment. Firstly, we build a case base with a number of simulated environments (with known ground truth) along with the trust models most suitable for each of them. Given an unknown environment, case-based retrieval algorithms retrieve the most similar case(s), and the trust model of the most similar case(s) is chosen as the most suitable model for the unknown environment. Evaluation results confirm the effectiveness of our framework in choosing suitable trust models for different e-marketplace environments.

## 1. Introduction

In multiagent e-marketplaces, self-interested selling agents may act maliciously by not delivering products with the same quality as promised. It is thus important for buying agents to reason about the trustworthiness (quality) of sellers in providing good quality products and determine which sellers to do business with. However, in such open and large environments, buyers often encounter sellers with which they have no previous experience. In this case, buyers often obtain advice (i.e., ratings) about the sellers from other buyers (called *advisors*). However, some advisors may also be dishonest and provide unfair ratings, to promote or demote some sellers (Irissappane et al., 2014).

Many trust models (Sabater & Sierra, 2005) have been proposed to assess seller trustworthiness, some of which, such as BLADE (Regan et al., 2006), also address the unfair rating problem. However, the performance (accuracy in predicting seller trustworthiness) of trust models is often highly affected by the environments where they are applied. Specifically, Fullam and Barber (2007) found out that the performance of trust models is influenced by environmental settings such as frequency of transactions, honesty of sellers and accuracy of advisors' ratings. A detailed comparison between BRS (Whitby et al., 2004), TRAVOS (Teacy et al., 2006) and Personalized (Zhang & Cohen, 2008) (see Sec. 2 for details) has been conducted by Zhang (2009) in a simulated dynamic e-marketplace environment. The results show that 1) BRS performs the best when buyers do not have much experience with sellers in the environment and the majority of advisors provide fair ratings about sellers; 2) TRAVOS has the advantage in the scenario where buyers have sufficient experience but advisors only lie about some specific sellers and 3) Personalized fares well when the majority of advisors are dishonest and sellers widely change their behavior over time.

In addition, almost all trust models rely on certain tuning parameters which may significantly affect their performance. For example, to identify a dishonest advisor, BRS uses the quantile parameter $(q)$ to determine whether the trustworthiness of a seller falls between $q$ quantile and $1 - q$ quantile of the distribution formed by the advisor's ratings to the seller. TRAVOS has the $bin$ parameter to divide $[0, 1]$ into $bin$ number of equal intervals, and Personalized uses the parameter of the minimum number of ratings required by buyers to have accurate modeling of seller trustworthiness.

Further, most trust models have only been evaluated in simulated e-marketplace environments, where ground truth i.e., the actual truth about agents' malicious behavior is known upfront, such as whether sellers deliver products with lower quality than what they promised and whether advisors provide unfair ratings. In simulated environments, the performance of trust models with specific parameter values can be evaluated, and the best models can then be easily chosen. However, for real e-marketplaces, it is difficult to obtain ground truth because it is expensive or time consuming to manually inspect every transaction. Even if we manage to find ground truth for a few real environments, we cannot guarantee that the best models in these environments will be the most suitable for all other environments. In addition, environments may keep changing, and a suitable model for an environment in one period may not be so in another period. Thus, choosing suitable trust models for real environments (where ground truth about agents' behavior is unknown, hence called *unknown environments*) is challenging and not well addressed, but important for practical applications.

In this paper, we propose a novel Case-Based Reasoning (CBR) framework to choose suitable trust models for unknown e-marketplace environments. CBR is a well-known artificial intelligence technique, which can be applied to complicated and unstructured problems relatively easily (Sormo et al., 2005). The fundamental concept in CBR is that similar problems will have similar solutions, with the advantage of learning continuously by just adding new cases to the case base. For the problem of choosing trust models, a similar intuition is that if a trust model performs well in one environment, it will do so in another similar environment. Thus, CBR becomes a suitable technique to address the problem by finding the trust models that are suitable for similar e-marketplace environments (i.e., similar problems). Specifically, in the proposed framework, we first find out the best trust models with their best parameter settings in a set of simulated environments, representing the case base. For a given unknown real environment, we find the most similar case(s) from the case base using case-based retrieval methods (Watson & Marir, 1994) such as $k$-nearest neighbors, K-dimension (K-d) trees, decision trees, etc. The trust model of the most similar case(s) is then chosen to be the most suitable trust model for the unknown environment.

The presented work is an extension to our previous work (Irissappane et al., 2013), which describes a simple framework to choose trust models using similarity based computation. In this paper, we make a number of additional contributions: 1) we formalize the framework to choose trust models using a case-based reasoning paradigm. Doing so, we have explored CBR techniques i.e., case representation and retrieval methodologies, to choose suitable trust models in an efficient manner; 2) we introduce additional case indexing and retrieval schemes, K-d trees and decision trees apart from $k$-nearest neighbors; 3) we introduce feature weights (in addition to feature selection), to improve the accuracy in determining the nearest neighbors in $k$-nearest neighbors and K-d tree retrieval techniques. While the above are contributions from the research perspective, we have also conducted more extensive and detailed experimentation to further demonstrate the effectiveness of the framework. Experimental results show that with a very high probability, our framework can choose the most suitable trust models to evaluate seller trustworthiness for different unknown environments. Evaluations also indicate that seller trustworthiness evaluated using trust models chosen

by our framework in a set of different e-market environments is more accurate than applying any specific trust model with its best parameter values in those environments. Specifically, the additional experiments: 1) justify the impact of using suitable trust models in e-marketplaces by demonstrating that suitable trust models produce more accurate estimate of seller trustworthiness and help buyers to make informed decisions, thereby resulting in greater utility for buyers than when using other (unsuitable) trust models; 2) consider an extended data set by increasing the number of cases in the case base from $972$ to $2268$ and show that the performance of the framework has improved using a larger case base; 3) compare the accuracy of $k$-nearest neighbors, K-d trees and decision trees in choosing suitable trust models and show that $k$-nearest neighbors and K-d trees outperform decision trees, while performing equally well; 4) compare the time complexity of the retrieval techniques, showing that decision trees require slightly lesser retrieval time than K-d trees, which in turn require lesser time than $k$-nearest neighbors; 5) show that adding weights to the features while determining the nearest neighbors in $k$-nearest neighbors and K-d trees improves the accuracy in choosing suitable trust models by a slight margin; 6) demonstrate that if the buyer chooses to aggregate the outcomes of all the trust models to determine seller trustworthiness instead of using a single most suitable trust model, it results in a high margin of error; 7) analyze the time complexity involved in extending the framework by adding new features to represent the environments in the case base and adding new defense models, both of which will improve the accuracy of the framework.

The rest of the paper is organized as follows. In Sec. 2, we provide an overview of the related research on choosing trust models. We clearly point out the shortcomings of the existing approaches, and explain how we cope with those shortcomings in our work. Sec. 3 describes the background on case-based reasoning. The detailed description of the framework is presented in Sec. 4. Here, we also describe how the framework can be extended to accommodate more trust models and different e-marketplace environments. In Sec. 5, we present the experimental results using seven trust models to demonstrate the accuracy of the framework (using $k$-nearest neighbors, K-d tree and decision tree retrieval) in correctly selecting the most suitable trust models for unknown environments. Finally, Sec. 6 concludes the current work and proposes future work.

## 2. Related Work

**Trust Models** Many trust models have been proposed in the literature. The Beta Reputation System (BRS) (Jøsang & Ismail, 2002) models seller trustworthiness as the expected value of the beta probability distribution of the (binary) ratings given by the advisors to the seller. To handle unfair ratings provided by advisors, Whitby et al. (2004) extend BRS to filter out those ratings that are not in the majority amongst other ones by using the Iterated Filtering approach. Specifically, if the cumulated trustworthiness score of a seller falls in the rejection area ($q$ quantile or $1-q$ quantile) of the beta distribution of an advisor's ratings to that seller, the advisor will be considered dishonest and filtered out. However, the Iterated Filtering approach is only effective when a significant majority of the ratings are fair, thereby leading to lower performance when the number of dishonest advisors is large. Teacy et al. (2006) propose TRAVOS to evaluate advisor trustworthiness, using it to discount their ratings before being aggregated to evaluate seller quality. TRAVOS divides the interval of $[0, 1]$ into $bin$ number of equal bins to determine the previous advice provided by the advisor that are similar to its current advice. Two pieces of advice are similar if they are within the same bin. The trustworthiness of the advisor is then calculated as the expected value of the beta probability density function representing the amount of the successful and unsuccessful interactions between

the buyer and the seller based on the previous advice. However, this model assumes that sellers behave consistently towards all the buyers in the e-marketplace, which might not be true in many cases. Yu and Singh (2003) use belief theory to represent trustworthiness scores. To determine seller quality, they rely on a referral network to find advisors, and thereby combine the beliefs of the advisors regarding the seller. The referral process begins with the buyer initially contacting a pre-defined *number of neighbors/advisors*, who may give an opinion about the seller or refer other advisors and continues until termination is reached. The referral process terminates in success when an opinion is received from an advisor and in failure when the *depth limit* of the referral network is reached or when it arrives at an advisor who neither gives an opinion nor a referral. Weights are also assigned to each advisor, in order to identify the deceptive ones.

The BLADE approach (Regan et al., 2006) applies Bayesian learning to reinterpret advisors' ratings instead of filtering the unfair ones. By establishing a correlation between seller properties and advisors' ratings, the buyer can infer advisors' subjective evaluation functions to derive certain properties of the seller. Though the reinterpretation helps to cope with advisors' subjectivity and deception simultaneously, a significant amount of evidence (ratings) is required to accurately determine the behavior of the advisors. Thereby, BLADE cannot perform effectively in sparse scenarios, where buyers do not have sufficient ratings to the sellers. In the personalized approach (Zhang & Cohen, 2008), the trustworthiness of a seller takes into account both the buyer's personal experience with the seller and the public knowledge about the seller. When the buyer has enough private information about (personal experience with) the seller (determined by the minimum number of transactions with the seller using the acceptable level of error $\epsilon$ and a confidence level $\gamma$), the buyer uses private knowledge alone, otherwise it uses an aggregation of both the private and public knowledge to compute the trustworthiness of the seller. A similar approach is used to compute advisor trustworthiness. Noorian et al. (2011) propose Prob-Cog, a two-layered cognitive approach to filter the ratings provided by advisors, based on the similarity between the ratings of the buyer and those of the advisor and the advisors' behavioral characteristics. In the first layer, advisors are filtered out if the average difference between the advisors' opinions and the buyer's personal ratings exceeds a threshold value $\mu$. In the second layer, the approach recognizes the behavioral characteristics of the advisors who have passed the first layer and subjectively evaluates their degree of trustworthiness. The approach has the advantage that it proposed the idea to differentiate advisors' behavior patterns. However, Prob-Cog assumes advisors' behavior to be consistent across all sellers, thereby making it inefficient when they dynamically change behavior by behaving honestly towards some sellers while being dishonest to others. The iCLUB approach (Liu et al., 2011) adopts a clustering technique DBSCAN, to filter out dishonest advisors based on local and global information. DBSCAN works by grouping points which are density-reachable i.e., not farther away than a given distance $\theta$ from each other. It also requires a pre-defined minimum number of points $minPts$ to form a dense region i.e., a cluster to be specified. In iCLUB, the DBSCAN clusters are formed using the ratings given by the buyer and advisors to the sellers. For a target seller, if advisors' ratings are not in the cluster containing the active buyer's ratings, the advisors are considered to be dishonest. When the buyer has no sufficient direct experience with the target seller (number of transactions is less than threshold $\tau$), the same process is applied on the non-target sellers.

As we can see, the performance of each trust model mentioned above varies depending on the environmental settings (especially buyer and seller behavior), where they are applied. Each trust model may not be the most suitable model for all the environments. Thus, for a given unknown

environment, it is necessary to choose from among a pool of trust models, in order to accurately assess seller trustworthiness and choose a good quality seller for a transaction.

**Existing Frameworks to Choose Trust Models** Only a few approaches have been proposed to choose trust models. For example, Hang et al. (2009) make use of explicitly indicated trust relationships by users in real-world systems (*e.g.,* FilmTrust) to evaluate trust models. For a weighted graph with vertices denoting agents and edges representing the direct relationship of trust from the agent at the source vertex to the agent at the target vertex, the weight (extent of trust between the agents at the vertices) of a particular edge can be determined from other relevant edges. For evaluation, an edge is temporarily removed and the weight on the edge is estimated. The accuracy in predicting the weight on the edge determines the effectiveness of the trust model. The major drawback with this method is that users may lie about their trust relationships, which in turn may affect the evaluation process. Some works (Wang et al., 2011; Irissappane & Zhang, 2014) use data from real-world e-markets (*e.g.,* eBay and Amazon) to evaluate the performance of trust models by their accuracy in predicting ratings of given transactions (i.e., for each seller, the ratings of the previous $i$ transactions are used to predict the $(i+1)^{th}$ rating to the seller). However, the ground truth about whether the ratings of those transactions are unfair may be unknown. One may argue that we can rely on buyers themselves to choose trust models because they know their true experience with sellers. But, it will be costly for buyers to evaluate each trust model with various parameters in the given environment.

Closely related to our work is the Personalized Trust Framework (PTF) (Huynh, 2009) that selects an appropriate trust model for a particular environment based on users' choice. Here, users can specify how to select a trust model based on the information about whose trustworthiness is to be evaluated and the configuration of trust models. In the framework, 1) a subject whose trustworthiness is to be evaluated is first sent to the trust manager. The trust manager stores many trust profiles which contain rules suggested by the end users, regarding which trust model to use for which subject; 2) the trust manager matches the subject's information with the trust profiles to find a suitable trust model and initializes the trust engine for the selected model; 3) the selected trust model then derives the trust value of the subject. PTF relies entirely on human intervention (users specify rules to select trust models). Though it is possible to identify certain rules to determine the most suitable trust model for some environments (e.g., BRS performs well when majority advisors are honest, BLADE performs well when advisors have subjective differences, etc.), it is impossible to know which models will perform the best in complex real world environments as they may have a variety of buyer and seller behavior. Also, the ground truth about the honesty and subjectivity of buyers and sellers is extremely challenging to determine, resulting in rules that will only be partial and thus insufficient to accurately choose suitable trust models when using PTF. On the other hand, in our case-based reasoning framework, we compare the properties of the unknown environment with existing cases in the case-base using an automated approach and choose suitable trust models, which are shown to be highly accurate through our experiments in Sec. 5.

## 3. Background

**Case-Based Reasoning (CBR)** is the process of solving new problems based on the solutions of similar past problems. Conceptually, CBR is commonly described by the CBR-cycle (Aamodt & Plaza, 1994). The CBR-cycle comprises of four activities: retrieve, reuse, revise and retain.

In the retrieve phase, one or more cases, similar to the new problem are selected from the case base. Many case-based retrieval algorithms exist in literature (Watson & Marir, 1994). Nearest

neighbor techniques (Duda & Hart, 1973) are perhaps the most widely used retrieval techniques in CBR. Distance measures such as Euclidean distance can be employed to identify the nearest neighbors (cases). Despite its simplicity, nearest neighbor retrieval has been successful in a large number of classification problems (Hastie et al., 2009). However, when the case base grows, the efficiency of retrieval decreases, because an increasing number of cases must be taken into account to find the most similar case. K-d trees (Wess et al., 1994), which organize the case base into a binary tree structure have been shown to reduce the complexity in the retrieval of the nearest neighbors. Alternatively, inductive retrieval algorithms (Soltani, 2013; Watson, 1999), determining which features do the best job in discriminating cases and generate a decision tree type structure to organize the cases in memory, can also be used to improve retrieval efficiency.

When one or more similar cases have been retrieved, the solution (or other problem solving information) contained in these cases is reused to solve the current problem. Reusing a retrieved solution can be quite simple, if the solution is returned unchanged as the proposed solution for the new problem. This is specifically the case for classification tasks with a limited number of solutions (classes) and a large number of cases. In such scenarios, every potential solution is contained in the case base and hence adaptation is usually not required. On the other hand, for synthetic tasks (such as configuration or planning) solution adaptation for the new problem is necessary.

In the revise phase, the solution determined so far is verified in the real world and possibly corrected or improved, e.g., by a domain expert. Finally, the retain phase takes the feedback from the revise phase and updates the knowledge, particularly the case base and the new problem solving experience becomes available for reuse in future problem solving episodes.

The major challenge in CBR resides in the retrieval of existing cases that are sufficiently similar to the new problem. Since e-marketplace environments with ground truth (existing cases) may not exist (or may be difficult to obtain), in our framework, we have to create them by simulations. In addition, in our framework, the features (characteristics of the e-marketplace environments) used to represent the cases in the case base are not known beforehand. We thus have to come up with an exhaustive list of potential features (to describe the e-marketplace) and carefully select the most relevant ones, in order to efficiently choose suitable trust models.

## 4. The Proposed Case-Based Reasoning Framework

Fig. 1 illustrates the detailed design of the framework. The most important component of the framework is the case base. To build the case base, we first simulate a large set of e-marketplace environments with known ground truth about the honesty of agents' behavior. Given a set of available trust models with specific values of their parameters (referred to as *candidate trust models*), we evaluate their performance in each simulated environment, where the best model is identified and forms a *best environment-model pair* (representing a *case* in the case base). In this process, we also choose the most relevant features to represent the cases, for efficient retrieval. Given an unknown real environment, the framework then extracts the set of carefully selected (most relevant) features and determines the most similar case(s) from the case base using case-based retrieval techniques. The trust model of the most similar case(s) is then reused as the solution for the unknown real environment. The given unknown environment along with the most suitable trust model is then retained in the framework for reuse in the future problem solving episodes. The major components of the framework and the detailed procedures are described in the following subsections.

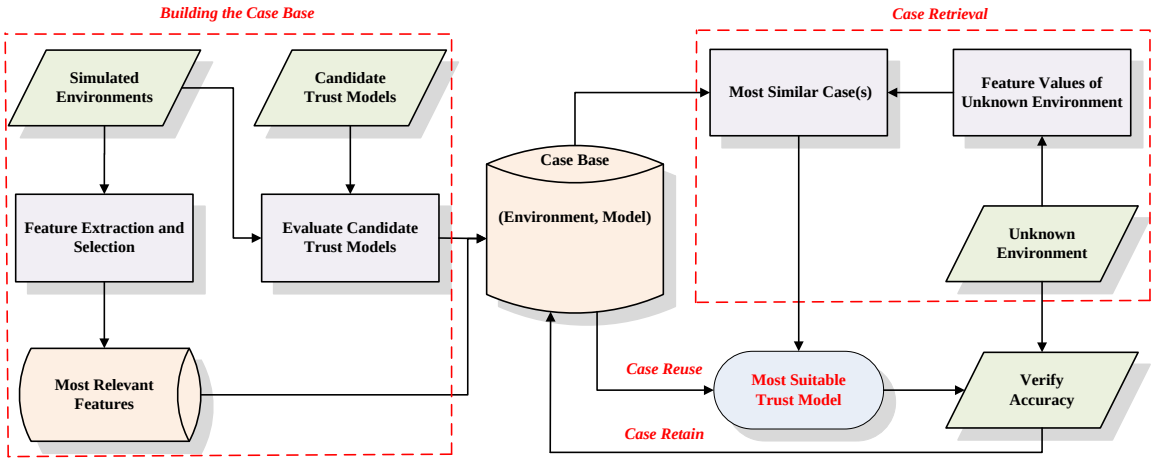



Figure 1: Design of the case-based reasoning framework

### 4.1 The Case Base

CBR is heavily dependant on the structure and content of the case base. In our framework, a case in the case base is described by an e-marketplace environment (represented by a set of carefully selected features) along with the trust model which performs the best in the environment. Unlike other domains, real e-marketplace environments with ground truth about the honesty of sellers and buyers are rare and may not exist, hence it becomes challenging to build the case base. We will mainly rely on simulations to create the existing cases in the case base.

**E-Marketplace Environments** An e-marketplace environment ($E$) consists of a set of sellers, a set of buyers, transactions (each of which is between a seller and a buyer with a certain monetary value) and ratings (each of which is given by a buyer to a seller at a specific time indicating whether the buyer is satisfied or not with the transaction). So, E is a tuple,

$$E = \langle S, B, \{T_{s,b} \,|s = 1...N_s, b = 1...N_b\}, \{R_{s,b} \,|s = 1...N_s, b = 1...N_b\}\rangle \tag{1}$$

where $S$ represents the set of all sellers, $B$ represents the set of all buyers, $N_s$ and $N_b$ are the numbers of sellers and buyers in $E$, respectively. $R_{s,b}$ denotes the set of ratings from buyer $b$ to seller $s$ for the transactions $T_{s,b}$. Each rating $r_{s,b} \in R_{s,b}$ for the transaction $t_{s,b} \in T_{s,b}$ is a tuple,

$$r_{s,b} = \langle id, s, b, h_s, h_b, t, val\rangle \tag{2}$$

where $id$, $s$, $b$ denote the rating index, index of the seller and that of the buyer, respectively. $h_s (\in [0, 1])$ and $h_b (\in \{honest, dishonest\})$ denote the ground truth i.e., the actual seller trustworthiness and honesty of the buyer for this transaction, respectively. A dishonest seller (with low trustworthiness) may advertise its products having high quality but actually deliver low quality ones or not deliver at all. Also, a dishonest buyer may lie about its satisfaction level of a transaction by providing an unfair rating. The $h_s$ and $h_b$ attributes help to distinguish such dishonest behaviors from the honest ones. The time (integer value denoting the day of simulation) when the rating is given is denoted by $t$. $val$ denotes the actual value of the rating, which can be binary (e.g., 0 or 1), multi-nominal (e.g., 1 - 5) or real (e.g., in the range $[0, 1]$).

There are two types of environments in our framework: 1) *known environments* ($E_{known}$), where the ground truth about seller and buyer honesty is known. The known environments along with their

most suitable trust models help in building the case base for our framework; 2) *unknown environments* ($E_{test}$) are those where ground truth is not known. They represent the test environments for which the most suitable trust models need to be determined.

To build the case base, we will simulate a large number of $E_{known}$ environments, to cover as many scenarios as possible and closely depict real-world environments. For example, we may simulate an environment with many sellers but fewer buyers (to represent a high provision e-marketplace) or with many buyers but fewer sellers (to illustrate a competitive e-marketplace). We may simulate a very sparse environment with few ratings provided by buyers, and a very dense environment where each seller is flooded with a large number of ratings. We may also simulate different scenarios where buyers are active or inactive in providing ratings. In these environments, we may also simulate sellers with different levels of honesty, and buyers launching different types of unfair rating attacks (Hoffman et al., 2009), including for example, unfair ratings to only reputable or disreputable sellers, a lot or few unfair ratings, unfair ratings given in a short or long time period, etc.

**Candidate Trust Models** As exemplified in Sec. 2, many trust models have been proposed to evaluate seller trustworthiness in e-marketplaces. New trust models will also likely be proposed in the future. All these trust models can be considered as candidate trust models in our framework. In addition, most of them have some parameters to tune, which may result in different performance. Thus, a candidate trust model ($TM$) is defined as a trust model with a specific value for each of its parameters. For a parameter varying in a range, we divide its range into a number of equal intervals and randomly choose a value in each interval. Ideally, the larger number of intervals is better.

**Feature Extraction and Selection** To formally represent an environment in the case base, each environment can be described by a set of features, representing the characteristics of the environment (e.g., ratio of number of buyers versus sellers, variance of ratings per seller or per buyer, average number of transactions per time period, percentage of rated sellers, etc.). An exhaustive list of potential features is extracted from which the most relevant features can be identified and used to represent the environment, in order to reduce the computational cost and increase the efficiency of the framework. If $F = \{f_1, ..., f_n\}$ is the set of all features and $P(\hat{F})$ be the performance of the framework while using a subset $\hat{F} \subset F$ of features. The most relevant subset of features $\hat{F}^*$ is chosen such that the framework achieves the best performance, formalized as follows:

$$\hat{F}^* = \underset{\hat{F} \subset F}{arg\ max}\ P(\hat{F}) \tag{3}$$

Before constructing the case base, we simulate another set of e-marketplace environments and evaluate the performance of our framework in these environments using all the possible features. The features whose values significantly correlate to the performance of the framework are determined using five widely used correlation and regression analysis techniques, namely Pearson correlation, Kendall rank correlation, Spearman rank correlation, linear regression (backward) and linear regression (stepwise). The results of the correlation are also analyzed by the Paired-Samples T-test to check for statistical significance. Each correlation and regression analysis technique results in a subset of significantly relevant features recognized by that technique ($\hat{F} \subset F$). The most influential set of features ($\hat{F}^*$) is then determined[1] from the five subsets of features (each recognized by the above five techniques, respectively) using Eqn. 3, the details of which will be presented in Sec. 5.1. Thereby, only the features in $\hat{F}^*$ will be used to represent the environments in the case base.

1. This feature selection process will be used to determine the most influential features only while using $k$-nearest neighbors, K-d tree retrieval and not decision trees as it employs its own embedded feature selection methodology.

**Best Environment-Model Pairs** Given a set of known environments and a set of candidate trust models, we find out in each environment ($E_{known}$), which candidate model ($TM$) shows the best performance. Specifically, the performance $P(E_{known}, TM)$ is measured in terms of a performance metric, such as the Mean Absolute Error (MAE) in determining seller trustworthiness, given by Eqn. 4, where $T_s^{true}$ and $T_s^{predicted}$ represent the actual and predicted trustworthiness of seller $s$, respectively. The lower the MAE, the better is the performance of the trust model. The above evaluations result in a set of best environment-model pairs $(E_{known}, TM^*)$, which form the case base. If several models perform equally best in an environment, we keep them all in the case base.

$$\text{MAE} = \frac{1}{N_s} \sum_{s \in S} |T_s^{true} - T_s^{predicted}| \tag{4}$$

### 4.2 Case Retrieval

Given an unknown environment $E_{test}$, case-based retrieval algorithms will retrieve the most similar case(s), $(E_{known}, TM)$ pair(s), whose simulated environment $E_{known}$ is the most similar to $E_{test}$. Every retrieval algorithm is a combination of a procedure for searching the case base to find the most similar case and a similarity assessment procedure, which determines the similarity between the given unknown environment $E_{test}$ and a known environment $E_{known}$ in the case base.

Firstly, we will consider the structural manner in which cases are represented in the case base, which plays a major role in the efficient retrieval of cases. The choice of such case representation chiefly depends on the type of problems the CBR system is intended to solve, varying from relatively simple feature-value vectors, to complex data-structures. In the framework, we propose to represent the case base using two structural representations (Watson & Marir, 1994): 1) flat representation; 2) hierarchical representation, and analyze the performance of the framework in both scenarios.

**Flat Representation** The simplest format to represent the cases in the case base is to have simple feature-value vectors for the environments (Eqn. 5), obtained from the most influential features (more suitable for cases with numeric feature values). In this flat memory model, all cases are organized at the same level and no relationships between features or between cases are shown.

$$E =< f_i \mid \forall f_i \in \hat{F}^* > \tag{5}$$

Classical nearest neighbor (Duda & Hart, 1973) retrieval is a method of choice for the retrieval of the cases with flat representation, as shown in Fig. 2. Given an unknown environment $E_{test}$, it is compared with the cases in the case base and similar cases are found according to the similarity in the features between $E_{test}$ and $E_{known}$ environments, measured in terms of Euclidean distance,

$$dist(E_{test}, E_{known}) = \sqrt{\sum_{f_i \in \hat{F}^*} (E_{test}(f_i) - E_{known}(f_i))^2} \tag{6}$$

$$TM^* = \underset{TM \in \mathbb{TM}}{arg\ max}\ N(TM) \tag{7}$$

Additionally, we can also assign weights to the different features while calculating the distance in Eqn. 6. In $k$-nearest neighbors, $k$ cases, which are closest to $E_{test}$ based on similarity, are retrieved and the most similar case(s) are chosen by a majority vote, such that the suitable trust model of the most similar case(s) occurs the maximum number of times among the $k$ closest cases, as shown in Eqn. 7, where $N(TM)$ represents the number of times the trust model $TM$ appears in the $k$ closest

cases, $\mathbb{TM}$ represents the set of all candidate trust models in the framework and $TM^*$ represents the trust model of the most similar case(s), which is the most suitable trust model for $E_{test}$.

The retrieval time in this memory organization is very high ($O(|C|)$, where $|C|$ is the number of cases in the case base), since for each retrieval, all the cases in the case base must be compared to the target case $E_{test}$, making it unsuitable for large case bases. However, this approach has been verified to provide maximum accuracy and easy retention.

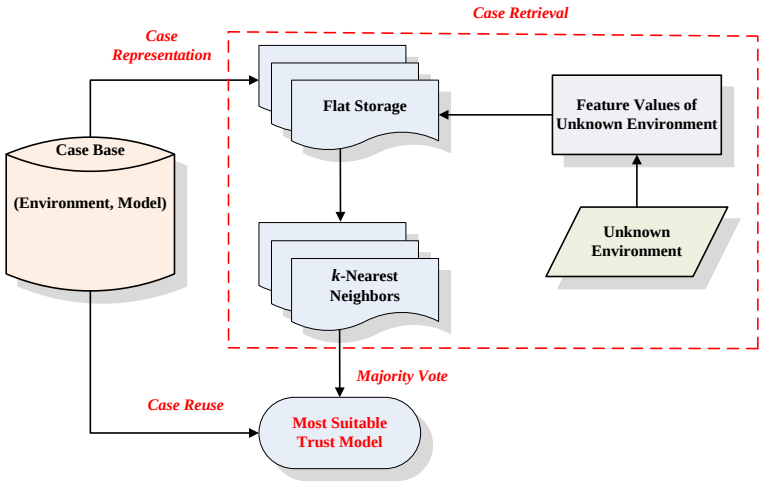



Figure 2: $k$-Nearest Neighbors retrieval

**Hierarchical Representation** For a more efficient and rapid retrieval, a more structured representation of the cases is necessary, because only a small subset of cases is needed to be considered during the retrieval from a large case base. Following a hierarchical representation helps to organize cases which share similar features under a more generalized structure. We demonstrate the use of two hierarchical tree structures which differ in their method of indexing (assigning indices to cases) but improve the efficiency of retrieval to a great extent.

**K-d Trees** Traditionally, K-dimensional (K-d) tree representation has been demonstrated to be very useful to reduce the retrieval time of the similar cases using nearest neighbors (Wess et al., 1994). K-d tree, where K represents the number of feature dimensions representing a case (i.e., K $= |\hat{F}^*|$), is a multi-dimensional binary search tree that splits the case base into groups of cases in such a way that each group contain cases that are similar to each other according to a given similarity measure. Each node in the K-d tree splits all its children along a specific feature, using a hyperplane that is perpendicular to the corresponding axis. At the root (which contains the entire case base), all children are split based on the first feature ($f_1 \in \hat{F}^*$), i.e., cases with $f_1$ less than (or equal to) the root will be in the left sub-tree and those greater than the root will be in the right sub-tree, as shown in Fig. 3. Each level down the tree divides the cases on the next feature $f_i \in \hat{F}^*$, returning to the first dimension $f_1$ once all other features have been exhausted. The leaves of the tree which contain a specific number of cases are called $buckets$. For partitioning, the median point of the feature ($f_1 = m_1$ as shown in Fig. 3) is selected for the root node and all cases with a smaller value (than $m_1$ for $f_1$) are placed to the left and larger to the right. A similar procedure is followed for the left and right sub-trees until the last trees to be partitioned are composed of few cases (not more than $bucketsize$).

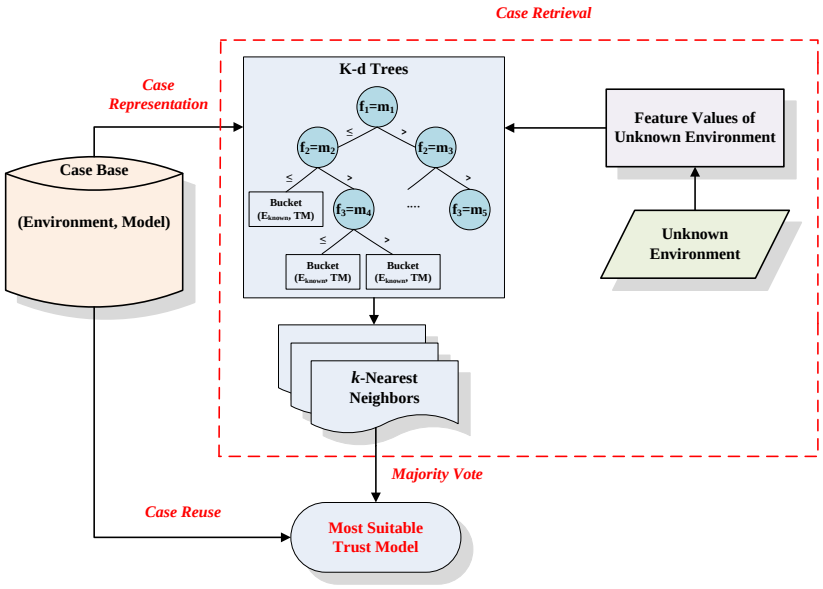



Figure 3: K-d Tree retrieval

For retrieval, a recursive search procedure is adopted. A queue containing $k$ most similar cases is maintained throughout the search. If the search examines a leaf node, the similarity of each case in the bucket with the given unknown environment $E_{test}$, is computed using Eqn. 6 as in $k$-nearest neighbors and the queue is updated. In case of a non-leaf node, the search is recursively called on the child node, where $E_{test}$ belongs (by comparing the features of $E_{test}$ with the partitioning value at each node). When this recursion terminates (at the non-leaf node), it is tested whether the other child of the node needs to be examined (if the geometric boundaries delimiting the cases under the node overlap the ball centered at $E_{test}$ with radius equal to the similarity of the $k^{th}$ nearest neighbor encountered so far, then the other child needs to be examined, and can be ignored otherwise). The procedure (unwinding the recursive search) is repeated until the root is reached. After determining the $k$ most similar cases (present in the queue), the most suitable trust model is determined using Eqn. 7. The average retrieval time for determining the $k$ most similar cases in K-d trees is found to be $O(k \times log|C|)$, where $|C|$ is the size of the case base.

**Decision Trees** Another hierarchical organization frequently used in CBR is Decision trees. Decision trees are induction-based models (Soltani, 2013) which learn general domain-specific knowledge from a set of training data and represent the knowledge in the form of trees. Decision trees (when compared to the other classes of learning methods), are quite fast, can be directly applied to the training data without much pre-processing and produce relatively interpretable models (Hastie et al., 2009). Unlike $k$-nearest neighbors and $K$-d trees, which use similarity based retrieval techniques, decision trees learn rules in order to determine the most suitable trust model. They also have an implicit feature selection process. Each node in the decision tree specifies a test of some feature attribute (e.g., $f_1$ in Fig. 4), and each branch descending from that node corresponds to possible values (e.g., $f_1 \leq v_1$ in Fig. 4 ) for this feature attribute. In making these trees, how much a feature can discriminate the cases is calculated (e.g., with information gain of cases) and the feature with highest discriminative power is located at the top of the tree. The calculation is again performed for the remaining features, thereby building the tree in a top-down fashion. The solution i.e., the most suitable trust model is located at the leaves of the tree.

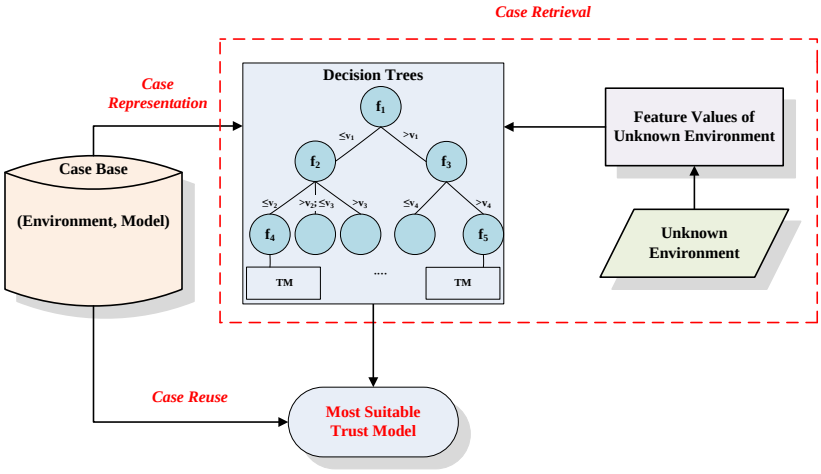



Figure 4: Decision Tree retrieval

Algorithms developed for decision trees are mostly the variations of a top-down, greedy search algorithm exemplified by ID3 (Quinlan, 1986) and its successor C4.5 (Quinlan, 1993). The algorithms construct the decision tree using the divide and conquer strategy i.e., they build the decision tree by recursively dividing the case base into subsets according to a splitting criterion called the information gain ratio. The intuition is to partition the case base in such a way that the information needed to classify a new case is reduced as much as possible. Eqn. 8 represents the information gain (discriminative power) for a feature $f_i \in F$, regarding a set of cases $C$ in the case base.

$$InformationGain(f_i, C) = Entropy(C) \; - \sum_{v \in V(f_i)} \frac{|C_v|}{|C|} Entropy(C_v) \tag{8}$$

$$\text{where,} \;\; Entropy(C) = \sum_{TM \in \mathbb{TM}} -p(TM)\, log_2\, p(TM) \tag{9}$$

$V(f_i)$ is the set of all possible values for feature $f_i$ and $C_v \in C$ is a set of cases with feature $f_i$ taking the value $v$. $p(TM)$ is the proportion of cases in the case base for which trust model $TM$ is the most suitable. Since decision trees have such a built-in feature selection methodology, with the most influential features selected and used in building the decision tree, we do not employ the feature selection process described in Sec. 4.1 before the case retrieval using decision trees, as it might affect the retrieval accuracy otherwise. For retrieval, features of the unknown environment ($E_{test}$) are compared with nodes in the tree, until it gets to one of the leaves that contains the most suitable trust model ($TM$ as shown in Fig. 4).

### 4.3 Case Reuse

After retrieving the most similar case(s) (using retrieval methods as discussed in the previous subsection) to the target case ($E_{test}$), the framework needs to reason according to the retrieved cases to find a reasonable and accurate solution (most suitable trust model) for $E_{test}$. The reuse of the solution can be done in two ways (Soltani, 2013): 1) reusing the solution of the retrieved case as the solution for the target case without any adaptation (applicable to classification problems); 2) adapting the retrieved solution to the target case, which is necessary for problem-solving tasks such as design, configuration, and planning. Since we deal with a classification problem, by identifying to which class (candidate trust model in our case) a given unknown environment belongs,

we do not perform any adaption and simply reuse the solution of the retrieved case(s)[2]. Thereby, our framework will choose the trust model of the retrieved case(s) as the most suitable model for $E_{test}$ (using $k$-nearest neighbors and K-d trees, while for decision tree retrieval the framework will directly choose the trust model suggested by the decision tree as no similar case(s) will be retrieved).

### 4.4 Case Retain

Case-based reasoning favors learning from experience. After choosing to reuse a solution from the retrieved case(s) for $E_{test}$, it may be found that the solution is, in fact, incorrect, thus providing an opportunity to learn from failure. Our framework offers a simple procedure, where the case solution is evaluated and if the solution is incorrect, it is revised and the best solution for $E_{test}$ is found. Then the new case along with the best trust model ($E_{test}, TM^*$) is retained in the case base (Fig. 1).

The proposed case-based reasoning framework is generic and can be further extended or concretized in the following aspects: 1) whenever a new trust model is proposed, it can be added into the framework. Our framework is capable of taking advantage of the trust model to improve the performance in evaluating seller trustworthiness; 2) whenever a new insightful feature is identified, it can be added into the framework to participate in the feature selection process and in fact may further improve the performance of the framework; 3) more promising feature selection methods such as incremental hill-climbers (Wettschereck & Aha, 1995), a wrapper model to measure the importance of features, can be adopted to enhance the performance of the framework and 4) more sophisticated memory representations can be used for a more efficient and fast retrieval of the cases.

## 5. Experimentation

We instantiate our framework and conduct a series of experiments to demonstrate its effectiveness in choosing suitable trust models. Firstly, we build the case base by generating a number of simulated environments and finding the most suitable trust models for them. In this process, we also determine the most influential features to represent the simulated environments in the case base. We then generate unknown (both simulated and real) environments for testing and verify the performance of the framework in choosing the best trust models for these unknown environments. We also compare the performance of $k$-nearest neighbors ($k$-NN), K-d tree (K-dT) and decision tree (DT) retrieval techniques, in finding the most suitable trust model for the given unknown environments.

### 5.1 The Case Base

**Simulated Environments** In the framework, 2268 e-marketplace environments ($E_{known}$) are simulated, consisting of different numbers of sellers (chosen from $\{10, 25, 50\}$) with different levels of trustworthiness $T_s^{true}$, uniformly distributed over $[0, 1]$. Sellers provide good quality products with a probability $T_s^{true}$ when interacting with each of the buyers. Honest buyers always provide correct opinions (similar to the actual seller trustworthiness $T_s^{true}$) about the sellers, while dishonest buyers[3] provide unfair ratings[4] i.e., incorrect opinions which are complimentary to the actual seller trustworthiness ($1 - T_s^{true}$). We simulate different distributions of fair ratings given by honest buy-

2. For more than one most similar cases, with different solutions, we randomly choose one of the solutions.
3. Buyers providing incorrect opinions due to subjective differences or ignorance are also considered dishonest.
4. Ratings in simulated environments are of the real type for being easily mapped to other types (binary, multi-nominal).

ers: 1) *sparse*, where a honest buyer rates a seller at most once; 2) *intensive*, where a honest buyer rates a seller more than once; 3) *mixed*, which is combination of sparse and intensive scenarios. We also simulate different unfair rating attack scenarios for dishonest buyers by adjusting $4$ parameters: 1) *individual attack frequency* denoting the average number of unfair ratings provided by each dishonest buyer which exhibit sparse, intensive or mixed behavior; 2) *attack period* referring to the period when unfair ratings are given, where $7$ and $100$ denote that dishonest buyers provide unfair ratings over one week (a concentrated attack) and $100$ days (a distributed attack), respectively. While dishonest buyers provide unfair ratings during the *attack period*, they behave honestly by providing fair ratings outside the *attack period*. This helps to simulate dynamic environments where buyers change their behaviors; (3) *attack target* taking a value of $0$ or $1$, indicating that attack targets are sellers with low trustworthiness ($T_s^{true} \leq 0.5$) or high trustworthiness ($T_s^{true} > 0.5$), respectively; 4) *overall attack rate* denoting the ratio of number of unfair ratings to fair ratings, chosen from $\{0.25, 1, 4\}$. Through the parameters of individual attack frequency and overall attack rate, the numbers of dishonest and honest buyers are determined. The marketplaces operate for $100$ days. The total number of ratings is chosen from $\{50, 75, 100, 150, 175, 200, 250\}$. We also limit the total number of ratings to $\{50\}$, $\{50, 100\}$, $\{50, 100, 200\}$ and $\{50, 100, 175, 250\}$ to simulate $324$, $648$, $972$ and $1296$ environments, respectively, in order to examine the influence of the number of simulated environments (size of the case base) on the performance of our framework.

**Candidate Trust Models** The framework includes $7$ representative trust models: BRS (Whitby et al., 2004), iCLUB (Liu et al., 2011), TRAVOS (Teacy et al., 2006), Personalized (Zhang & Cohen, 2008), Referral Networks (Yu & Singh, 2003), BLADE (Regan et al., 2006) and Prob-Cog (Noorian et al., 2011). The following parameters (as described in Sec. 2) are considered to design the candidate trust models: 1) for BRS, the quantile parameter $q \in \{0.05, 0.1, 0.3, 0.5\}$, which is used to filter dishonest buyers is considered; 2) for TRAVOS, the number of bins to determine the acceptable error level in buyers' ratings $bin \in \{2, 3, 5, 8, 10\}$ is considered; 3) for Referral Networks, *number of neighbors* $\in \{2, 4, 6\}$ and *depth limit* of referral networks $\in \{4, 6, 8\}$ are considered; 4) for Personalized, error level $\epsilon \in \{0.3, 0.5, 0.7\}$ and confidence level $\gamma \in \{0.3, 0.5, 0.7\}$ are considered; 5) for Prob-Cog, we consider the threshold to filter out dishonest buyers $\mu \in \{0.1, 0.2, \ldots, 0.9\}$; 6) for iCLUB, we consider the minimum number of ratings required to form a DBSCAN cluster $minPts \in [1, 6]$, maximum neighbor distance $\theta \in [0.3, 0.7]$ and threshold to choose the local or global component $\tau \in [3, 6]$. In the end, we obtain $45$ candidate trust models ($\mathbb{TM}$) in total.

**Feature Selection** We consider a set of $18$ potential features ($F$) to analyze the characteristics of the simulated environments, as listed in Table 1. We use some general statistical metrics to describe the features. For example, variance refers to the spread of values, skewness describes the asymmetry from the normal distribution, etc. A satisfactory seller refers to the one who receives more positive ratings than negative ones from buyers. An active buyer refers to the buyer, who provides at least one rating to any seller. The feature values for the simulated environments are extracted using the parameters to generate the simulated environments, as described in the above subsection. Since the features (in Table 1) do not depend on the ground truth (buyer and seller honesty), it is also easier to extract such feature values for unknown environments (with no ground truth).

To select the most relevant features (for efficient retrieval using $k$-NN and K-dT), we adopt the five correlation and regression analysis techniques mentioned in Sec. 4.1. The results of the analysis of the $18$ features on how they are correlated to the performance (MAE) of the framework is shown in Table 1. Here, '*' denotes that the feature has a significant correlation (after Paired-Samples

Table 1: Selection of the most relevant features

| | Features | Pearson (C1) | Kendall (C2) | Spearman (C3) | Backward (C4) | Stepwise (C5) | All (C6) |
|---|---|---|---|---|---|---|---|
| 1 | Variance of the Percentage of Ratings for each Seller | | | | | * | * |
| 2 | Avg. Number of Ratings Provided by each Buyer for each Seller | * | * | * | * | * | * |
| 3 | Ratio of Number of Buyers versus Number of Sellers | * | * | * | | | * |
| 4 | Skewness of Rating Period | * | * | * | | | * |
| 5 | Variance of Percentage of Ratings Provided by each Buyer | * | * | * | * | * | * |
| 6 | Skewness of Number of Ratings Provided by each Buyer | | | | * | * | * |
| 7 | Percentage of Satisfactory Sellers | * | * | * | * | * | * |
| 8 | Number of Buyers | * | * | * | * | | * |
| 9 | Avg. Number of Ratings for each Seller | * | * | * | * | * | * |
| 10 | Variance of Number of Ratings provided by each Buyer | | | | * | * | * |
| 11 | Total Number of Ratings | * | * | * | * | * | * |
| 12 | Variance of Number of Ratings for each Seller | * | * | * | * | * | * |
| 13 | Skewness of Number of Ratings for each Seller | * | * | * | | * | * |
| 14 | Avg. Number of Transactions in each Day | | | | * | * | * |
| 15 | Total Percentage of Sellers Rated by Buyers | * | * | * | * | * | * |
| 16 | Time Period the Marketplace Operates | * | * | * | | | * |
| 17 | Maximum Percentage of Ratings for Sellers | * | * | * | * | | * |
| 18 | Total Percentage of Buyers who are Active in the Marketplace | * | | | * | * | * |

T-test) to the performance of the framework. In Table 1, columns C1, C2, C3, C4 and C5 represent the combination of the features flagged with '*'. C6 represents a combination of all the features. To verify the effectiveness of the 6 feature combinations, we randomly generate a large number of unknown environments and compare the results. We obtain an average MAE (using $k$-NN retrieval[5]) of $0.44$, $0.36$, $0.36$, $0.25$, $0.33$, $0.32$ for the combinations C1, C2, C3, C4, C5 and C6, respectively. C4 has the lowest MAE and is chosen as the set of most influential features ($\hat{F}^*$), by Eqn. 3. The features in C4 will be used for comparing the unknown and simulated environments in the rest of the experiments (using $k$-NN and K-dT retrieval to obtain the similar case(s) for $E_{test}$).

**Best Environment-Model Pairs** For each simulated environment, we find out the best candidate trust model based on the performance metric MAE. MAE is a suitable metric to assess the performance of the trust models because accurately determining the trustworthiness of sellers helps buyers to choose good transaction partners, thereby increasing their utility in the long run (as demonstrated by the experiments in Sec. 5.4). We first calculate the MAE of all candidate trust models in predicting seller trustworthiness for the simulated environments and select the one with the lowest MAE value. Here, we also compute the difference in MAE (for each seller in the e-marketplace environment) between pairs of trust models to assess if the MAE values of the most suitable trust model is significantly better than all the others (using Paired-Samples T-test). In the end, we obtain $3664$ best environment-model pairs[6] $(E_{known}, TM^*)$, which form the case base for the framework. Fig. 5(a) illustrates the number of simulated environments in the case base where each candidate trust model achieves the best performance, which are $733$, $306$, $448$, $979$, $190$, $253$ and $755$ for BRS, iCLUB, TRAVOS, Personalized, Referral, BLADE and Prob-Cog, respectively. The numbers indicate that the case base contains sufficient number of cases for each trust model. A sample case i.e., $(E_{known}, TM^*)$ is shown in Eqn. 10. Here $E_{known}$ is described by the $18$ features (we show all the features considered before the feature selection process for clarity) in the order as mentioned in Table 1 and $TM^*$ is the BLADE model for this environment.

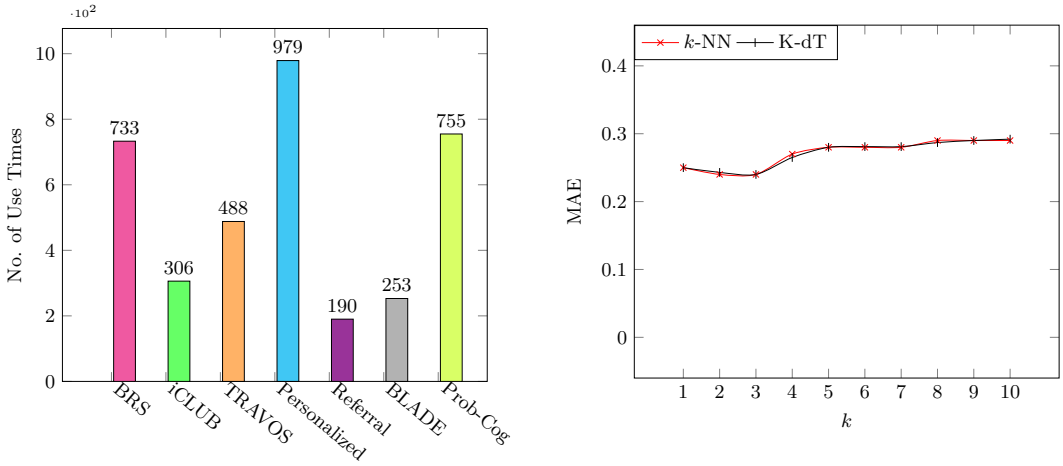
5. K-d tree retrieval obtains similar MAE values.

6. A simulated environment can have more than one most suitable trust model (which do not significantly outperform each other) with the lowest MAE.

$$(E_{known}, TM^*) = (< 0.30, 0.09, 18.2, 0.04, 0.4, 2.5, 0.6, 182, 19, 0.36, 200, 0.18, 4.3, 2, 1, 100, 0.62, 0.2 >, BLADE) \quad (10)$$

### 5.2 Case Retrieval Algorithms

We analyze the performance of $k$-NN, K-dT and DT retrieval techniques in identifying suitable trust models for unknown environments.




(a) (b)

Figure 5: (a) No. of times each trust model is selected as the most suitable model for simulated environments; (b) Influence of $k$ on the MAE in determining seller trustworthiness using $k$-NN and K-dT

We randomly generate 972 unknown environments (using different values for the parameters to generate the simulated environments) and evaluate the performance of $k$-NN and $k$-dT in choosing suitable trust models, for different values of $k$. Table 2 presents the influence of $k$ on the accuracy of choosing the most suitable trust models (with the most suitable parameters) for the 972 randomly generated unknown environments, using $k$-NN and $k$-dT retrieval techniques. *Correct Model* indicates that the trust model chosen is the same as the best model identified by evaluating all candidate trust models in the given unknown environment. *Correct Model and Paras* indicates that the correct trust model is chosen with the appropriate tuning parameters. Also, $\epsilon = 0.05$ is a tolerance value, indicating that the difference between the MAE of the chosen trust model and that of the truly most suitable model is within $\epsilon$. We find that the accuracy in choosing the correct trust models (with parameters) is the highest when $k = 3$ and acceptable when $k = 1, 2$ for both $k$-NN and K-dT. For $k$ values greater than 3, the performance decreases, signifying that the boundaries between the classes (candidate trust models) become less distinct. Fig. 5(b) shows the MAE in determining seller trustworthiness (corresponding to the accuracy in Table 2), when the value of $k$ is increased from 1 to 10. $k$-NN and K-dT obtain similar MAE values in determining seller trustworthiness for different values of $k$. Again, we find that when $k \in \{2, 3\}$, the lowest MAE (0.24) is achieved. Hence, we will use $k = 3$ for all the experiments (using $k$-NN and K-dT) in this paper.

For K-dT retrieval, we use the weka implementation (median based partitioning with a maximum of 20 instances in a leaf node). The K-d tree is built using the 3664 best environment-model pairs (described in the previous subsection). For the DT retrieval, we use the C4.5 algorithm (J48 weka implementation with pruning confidence 0.25 and minimum number of instances as 2, which

Table 2: Influence of $k$ on the accuracy of the framework

| $k$-NN | $k$=1 | $k$=2 | $k$=3 | $k$=4 | $k$=5 | $k$=6 | $k$=7 | $k$=8 | $k$=9 | $k$=10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Correct Model | 94.0% | 96.0% | 97.0% | 89.0% | 82.0% | 76.0% | 75.0% | 73.0% | 71.0% | 71.0% |
| Correct Model with $\epsilon$ | 97.0% | 97.0% | 99.0% | 91.0% | 84.0% | 78.0% | 77.0% | 75.0% | 73.0% | 73.0% |
| Correct Model and Paras | 92.0% | 94.0% | 97.0% | 85.0% | 77.0% | 71.0% | 69.0% | 67.0% | 65.0% | 65.0% |
| Correct Model and Paras with $\epsilon$ | 96.0% | 97.0% | 98.0% | 87.0% | 79.0% | 73.0% | 71.0% | 69.0% | 67.0% | 67.0% |
| K-dT | $k$=1 | $k$=2 | $k$=3 | $k$=4 | $k$=5 | $k$=6 | $k$=7 | $k$=8 | $k$=9 | $k$=10 |
| Correct Model | 94.0% | 95.0% | 97.0% | 89.0% | 83.0% | 76.0% | 75.0% | 74.0% | 72.0% | 71.0% |
| Correct Model with $\epsilon$ | 96.0% | 97.0% | 98.0% | 92.0% | 84.0% | 78.0% | 76.0% | 75.0% | 73.0% | 73.0% |
| Correct Model and Paras | 91.0% | 94.0% | 97.0% | 86.0% | 78.0% | 71.0% | 69.0% | 68.0% | 65.0% | 65.0% |
| Correct Model and Paras with $\epsilon$ | 95.0% | 96.0% | 98.0% | 87.0% | 79.0% | 73.0% | 71.0% | 69.0% | 67.0% | 66.0% |

are the default values). The decision tree is also built using the $3664$ best environment-model pairs, which will then be used to find the most suitable model for the unknown environments.

### 5.3 Unknown Environments for Testing

The framework is evaluated using $6$ categories of unknown environments $E_{test}$ (where ground truth about seller and buyer honesty is in fact known) in both normal and extreme scenarios.

**Unknown Random Environments** are generated using parameter values different from simulated environments such as: 1) number of sellers $\in \{33, 66, 99\}$; 2) total number of ratings $\in \{333, 666, 999\}$; 3) ratio of number of unfair ratings versus fair ratings $\in \{0.1, 1, 10\}$; 4) time period of attacks $\in \{50, 100\}$, from which $100$ environments are randomly chosen for testing.

**Unknown Real Environments** Real data is obtained from $IMDB.com$, where users rate movies directed by different directors. We remove outlying ratings and select only directors whose movies are very highly rated (resulting in $40$ different directors, with $1142$ movies rated by $188$ users). We then simulate $3$ types of unfair rating attacks, namely RepBad, RepSelf and RepTrap (Yang et al., 2008), to bad-mouth targeted directors (sellers in our case). We also employ a combination of these attacks. Finally, we generate $48$ real environments with simulated attacks.

**Large Environments** $160$ environments where number of sellers is larger than $50$, number of ratings is larger than $100$ and number of buyers is larger than $80$ are generated.

**Extremely Sparse Environments** are those where buyers do not provide sufficient ratings. Specifically, each buyer gives an average of $0.1$ ratings to sellers. We generate $36$ such environments where the number of sellers is $10$, total number of ratings is $100$, and overall attack rate $\in \{0.25, 1, 4\}$.

**Environments with Dynamic Buyer and Seller Behavior** $35$ environments (number of sellers is $10$ and total number of ratings is $50$) where sellers/buyers change their behavior dynamically are generated. Sellers change their behavior by providing complimentary quality products (than previously presented) after a random period of operation in the e-market. (Dishonest) buyers change their behavior by providing unfair ratings only during specific periods and behaving honestly, otherwise.

**Environments with Many Attacks** $24$ environments with intensive attacking scenarios, where attack rate is larger than $10$ are generated. We specifically use real data from $IMDB.com$ and simulate RepBad, RepSelf, RepTrap attacks and their combination.

### 5.4 Experimental Results

**Performance Comparison in Unknown Random and Real Environments** Table 3 presents the accuracy of our framework in choosing the most suitable trust models (with the most suitable parameters) in unknown random and real environments (using $k$-NN, K-dT and DT retrieval techniques). As mentioned in Sec. 5.2, a correct selection indicates that the trust model chosen is the same as the best model identified by evaluating all candidate trust models in a given unknown environment and $\epsilon$ is the tolerance value, indicating that the difference between the MAE of the chosen trust model and that of the truly most suitable model is within $\epsilon$.

Table 3: Accuracy of choosing most suitable trust models (with parameters) for unknown random and real environments

| Unknown Random Environments | | | | | |
|---|---|---|---|---|---|
| $k$-Nearest Neighbors ($k$-NN) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 81.0% | 84.0% | 92.0% | 96.0% | 97.0% |
| Correct Model with $\epsilon$ | 87.0% | 89.0% | 95.0% | 98.0% | 98.0% |
| Correct Models and Paras | 72.0% | 76.0% | 82.0% | 96.0% | 97.0% |
| Correct Model and Paras with $\epsilon$ | 85.0% | 86.0% | 94.0% | 97.0% | 98.0% |
| K-d Trees (K-dT) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 80.0% | 84.0% | 92.5% | 95.0% | 97.0% |
| Correct Model with $\epsilon$ | 87.0% | 90.0% | 95.0% | 98.0% | 98.0% |
| Correct Models and Paras | 71.0% | 76.0% | 82.5% | 95.0% | 97.0% |
| Correct Model and Paras with $\epsilon$ | 85.0% | 87.0% | 94.2% | 97.0% | 98.0% |
| Decision Trees (DT) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 52.0% | 54.0% | 63.0% | 72.0% | 80.0% |
| Correct Model with $\epsilon$ | 64.0% | 67.0% | 72.0% | 78.0% | 85.0% |
| Correct Model and Paras | 31.0% | 33.0% | 35.0% | 40.0% | 46.0% |
| Correct Model and Paras with $\epsilon$ | 49.0% | 53.0% | 58.0% | 63.0% | 67.0% |
| Unknown Real Environments | | | | | |
| $k$-Nearest Neighbors ($k$-NN) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 81.3% | 83.3% | 83.3% | 86.3% | 87.0% |
| Correct Model with $\epsilon$ | 89.6% | 95.8% | 95.8% | 97.3% | 97.3% |
| Correct Model and Paras | 72.9% | 75.0% | 77.1% | 79.3% | 81.3% |
| Correct Model and Paras with $\epsilon$ | 89.6% | 95.8% | 95.8% | 96.3% | 97.2% |
| K-d Trees (K-dT) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 80.1% | 82.3% | 83.3% | 86.7% | 87.5% |
| Correct Model with $\epsilon$ | 89.0% | 95.1% | 96.2% | 97.0% | 97.0% |
| Correct Model and Paras | 71.0% | 74.0% | 78.1% | 79.0% | 82.0% |
| Correct Model and Paras with $\epsilon$ | 88.0% | 95.0% | 95.2% | 96.0% | 97.5% |
| Decision Trees (DT) | 324 SE | 648 SE | 972 SE | 1296 SE | 2268 SE |
| Correct Model | 08.3% | 08.3% | 14.6% | 40.3% | 45.0 % |
| Correct Model with $\epsilon$ | 55.1% | 58.3% | 68.8% | 80.3% | 83.3% |
| Correct Model and Paras | 00.0% | 00.0% | 00.0% | 02.0% | 02.1% |
| Correct Model and Paras with $\epsilon$ | 54.3% | 59.2% | 68.8% | 80.3% | 83.3% |

From Table 3 (under unknown random environments), we can see that the accuracy of our framework increases as the number of simulated environments (SE) in the case base increases (the trend is the same for $k$-NN, K-dT and DT), and is the best when there are $2268$ simulated environments (SE) in the case base. This is because with a larger number of cases in the case base, it is easier to find a closely similar environment to the given unknown environment.

We also find that $k$-NN and K-dT show similar performance, while outperforming the DT retrieval technique. K-dT is mainly used to improve the retrieval time in $k$-NN through appropriate organization of cases in the form of trees. However, for retrieval, K-dT uses the same similarity

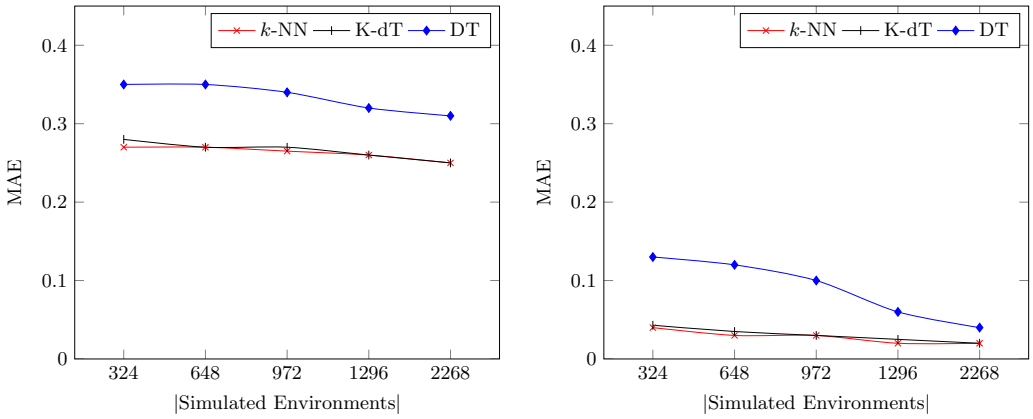



Figure 6: Influence of the number of simulated environments on the performance of $k$-NN, K-dT and DT retrieval techniques: (a) random environments; (b) real environments

measure and the same number of nearest neighbors as $k$-NN. This is the reason for the similar performance of $k$-NN and K-dT. On the other hand, DT retrieval shows a lower performance as it requires more training instances (cases in the case base) to learn the entire problem space (and build a complete decision tree) and is known to show a surge in performance when dealing with continuous feature values (Quinlan, 1996) (in our framework the feature values are continuous). With smaller number of cases (say 324 SE), DT obtains only an accuracy of $52.0\%$ in choosing the best trust model for the unknown random environments. Even with 2268 SE the accuracy is only $80.0\%$, which is less than the accuracy for $k$-NN with 324 SE. $k$-NN and K-dT obtain an accuracy of $97.0\%$ in selecting the most suitable models and a $98.0\%$ accuracy with a tolerance $\epsilon = 0.05$, for 2268 simulated environments. Thus, it shows that our framework, using the $k$-NN, K-dT retrieval techniques can choose candidate models whose performance is very close to the ideal case. Even with only 324 simulated environments, the performance of $k$-NN and K-dT is still acceptable, selecting the most suitable models with an accuracy of $81.0\%$ and $80.0\%$, respectively.

Fig. 6 shows the influence of the number of simulated environments (size of the case base) on the MAE obtained, while determining seller trustworthiness using the candidate trust model suggested by the $k$-NN, K-dT and DT retrieval techniques. The more accurate selection of the best trust model results in a lower MAE value in determining seller trustworthiness. Fig. 6(a) shows that $k$-NN and K-dT obtain a lower MAE than DT in all cases. When the number of simulated environments is 2268, $k$-NN and K-dT, both obtain an MAE of $0.25$, while DT obtains an MAE of $0.31$. However, when the number of simulated environments is increased, we find that the rate at which the MAE of DT decreases is greater than $k$-NN and K-dT, because with more training instances, DT can produce more accurate results. Eventually, when the number of simulated environments is further increased (greater than 2268), DT may show the same (even better) performance as $k$-NN and K-dT. However, we do not further increase the number of simulated environments in our experiments due to the complexity involved in building the case base, by evaluating all the $45$ candidate trust models in each simulated environment and selecting the most suitable model for each of them.

Table 3 (under unknown real environments) again shows that $k$-NN and K-dT perform equally well (with accuracy of $87.0\%$ and $87.5\%$ in selecting the most suitable models with 2268 simulated environments, respectively), outperforming DT retrieval (with accuracy of $45.0\%$) in real environments. We also notice that the accuracy of all the techniques is lower than that in the ran-

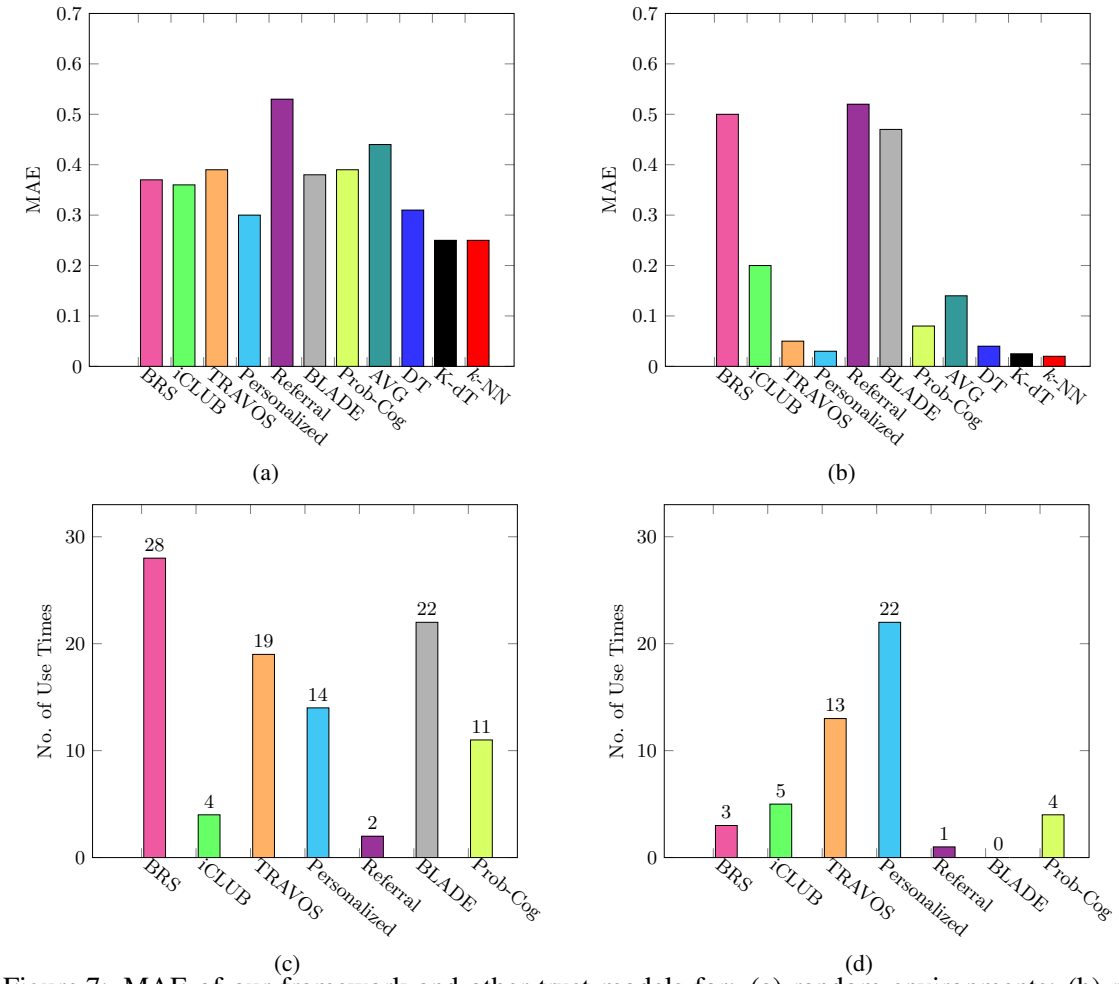



Figure 7: MAE of our framework and other trust models for: (a) random environments; (b) real environments; No. of times each trust model is selected as the most suitable for: (c) random environments; (d) real environments

dom environments. This is because, characteristics of real environments may vary extensively from those of the simulated environments in the case base, making it difficult for the retrieval algorithms to identify similar cases whose simulated environment is similar to the real one. Nevertheless, the performance of $k$-NN and K-dT in unknown real environments is also sufficient (greater than $86\%$). Fig. 6(b) again shows that $k$-NN, K-dT perform better than decision trees. However, we can see that the MAE values in Fig. 6(b) are smaller than those in Fig. 6(a), though the accuracy for real environments (Table 3) is lower than unknown random environments. This is because, we assume that the sellers (in real environments) are either of high or low quality (while in unknown random environments seller quality is uniformly distributed in $[0, 1]$), thus easily identifiable by the candidate trust models. We also find that the average MAE in determining seller trustworthiness using the truly most suitable trust models for the unknown real environments is $0.01$, while for the unknown random environments it is $0.22$, a comparatively larger value. Thereby, for unknown real environments, the framework chooses a trust model whose MAE is similar to that of the truly most suitable trust model to obtain better accuracy, which in this case has a lower value than the MAE for the

unknown random environments. The greater rate at which MAE decreases for DT is very evident in Fig. 6(b), since in real environments we assume seller trustworthiness to be binary, thereby, even a small variation in the choice of trust models can impact the MAE values to a great extent.

Fig. 7(a-b) show the MAE of our framework in comparison with the other trust models in unknown random and unknown real environments. For the other trust models in the unknown environment, we use their best parameter values. We show the performance of $k$-NN, K-dT and DT using $2268$ simulated environments, to obtain the best performance. To demonstrate the scenario when buyers may choose to aggregate the outcomes of all trust models (instead of using a single most suitable trust model), while determining seller trustworthiness, we also show the MAE obtained by adopting such a heuristic denoted by AVG in Fig. 7(a-b). From Fig. 7(a), we find that $k$-NN and K-dT obtain the lowest MAE of $0.25$, showing that they are able to choose better trust models to evaluate seller trustworthiness than always applying a single model. DT obtains an MAE of $0.31$, a higher value than Personalized with MAE $0.30$. AVG obtains a higher MAE of $0.44$, showing that using the aggregated outcome of all trust models may not result in accurate values for seller trustworthiness. For the unknown real environments (Fig. 7(b)), again $k$-NN and K-dT obtain the lowest MAE ($0.022$ and $0.025$, respectively) when compared to other trust models.

Fig. 7(c-d) shows the numbers of unknown random environments and unknown real environments, respectively for which each trust model is chosen as the most suitable one, using the $k$-NN retrieval technique (K-dT also obtains similar values). The numbers are $28, 4, 19, 14, 2, 22$ and $11$ for BRS, iCLUB, TRAVOS, Personalized, Referral, BLADE and Prob-Cog, respectively for the $100$ unknown random environments, and $3, 5, 13, 22, 1, 0$ and $4$ for these models in the $48$ unknown real environments. The numbers signify that our framework is able to choose different models from a candidate set for various unknown environments. The difference in the use times of the trust models between random and real environments also indicate that trust models perform differently in different kinds of environments.

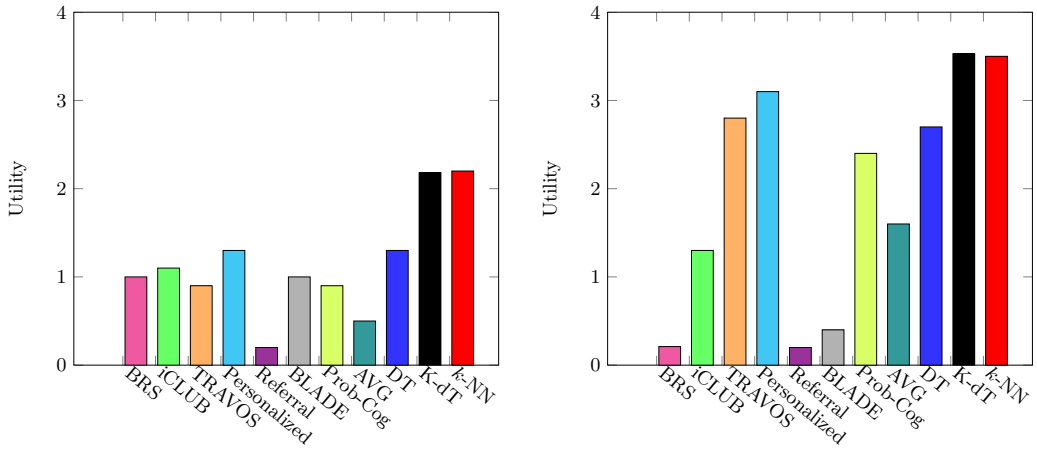



Figure 8: Average utility of buyers: (a) random environments; (b) real environments

Fig. 8(a-b) show the average utility of all the buyers in the e-marketplace corresponding to the MAE of the trust models in Fig. 7(a-b). Specifically, a buyer gains a reward of $+5$ when he chooses a high quality seller (by evaluating the trustworthiness of all the sellers in the market using the prescribed trust model), with $T_s^{true} > 0.5$ and a penalty of $-5$, on choosing a low quality seller with $T_s^{true} \leq 0.5$, for a transaction. Fig. 8(a) shows that $k$-NN and K-dT obtain the highest utility of

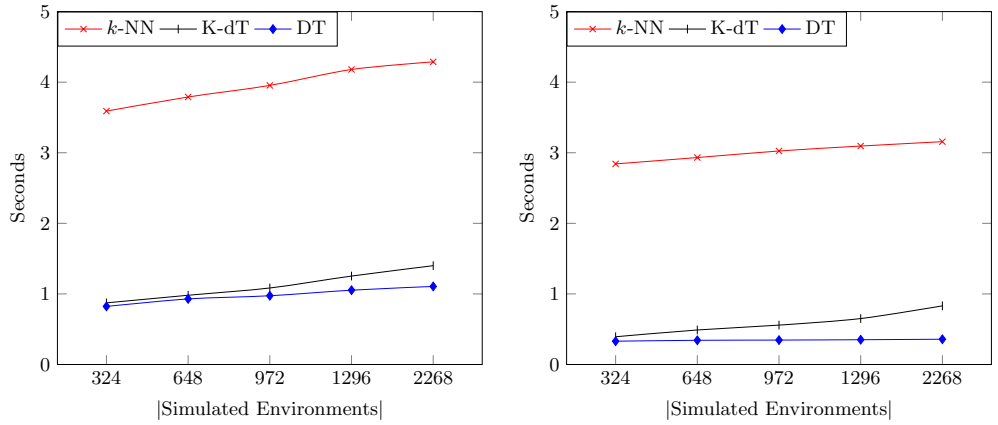



Figure 9: Time to choose the best trust models: (a) random environments; (b) real environments

2.20 and 2.18, respectively. The trend also shows that trust models with a higher MAE (in Fig. 7(a)) have a lower utility than those with a lower MAE, because when the buyers are able to accurately predict seller trustworthiness, they can correctly choose good quality sellers as transaction partners. Fig. 8(b) also shows that $k$-NN and K-dT obtain the highest utility of 3.50 and 3.53, respectively. The experiments in Fig. 8(a-b) also justify that MAE is a suitable metric to assess the performance of a trust model in a given environment, as it (indirectly) monitors the decisions of the buyers, by having a large impact on the utility/value addition gained from their transactions with the sellers.

Fig. 9(a-b) show the time taken by the framework to choose the best trust models for the unknown random and real environments, using $k$-NN, K-dT and DT retrieval techniques. Though K-dT obtains a similar accuracy as $k$-NN (Table 3 and Fig. 7), it greatly improves the time taken to find the most suitable trust model, as shown in Fig. 9(a-b). Specifically, Fig. 9(a) shows that the time taken to find the suitable trust models for the 100 unknown random environments by $k$-NN, K-dT and DT is $4.18s$, $1.40s$ and $1.10s$, using 2268 simulated environments in the case base, respectively. We find that both K-dT and DT approaches are faster than $k$-NN, which compares the features of the unknown environment with all the cases in the case base (Soltani, 2013). K-dT and DT use a tree structure to represent the cases in the case base (as described in Sec. 4.2) and retrieve the most suitable trust model by traversing the tree. However, decision tree retrieval is slightly faster than K-dT. This is because in K-dT, the dimensionality (number of features) of the cases and the number of similar cases ($k$ nearest neighbors) that are needed to be retrieved, affect the retrieval time (requiring more number of leaves to be visited through backtracking). Literature (Ahmed, 2004; Vempala, 2012) also shows that with high-dimensional data (greater than 20), most of the leaves in the K-d tree are visited and the efficiency is no better than exhaustive $k$-NN search, which can be a concern when the feature space in the framework is further increased. From Fig. 9(b), we can again see that for unknown real environments K-dT and DT require lesser retrieval time than $k$-NN. The time taken by $k$-NN, K-dT and DT is $3.15s$, $0.83s$ and $0.35s$, using 2268 simulated environments in the case base, respectively. However, the values are lower than those in Fig. 9(a), since we consider the time taken to choose the best trust models for all the 100 unknown random environments, while the number of real environments considered is only 48. Though the time taken by K-dT is slightly higher than DT, it is still comparable and shows a much better performance in terms of retrieval accuracy (Table 3 and Fig. 7).

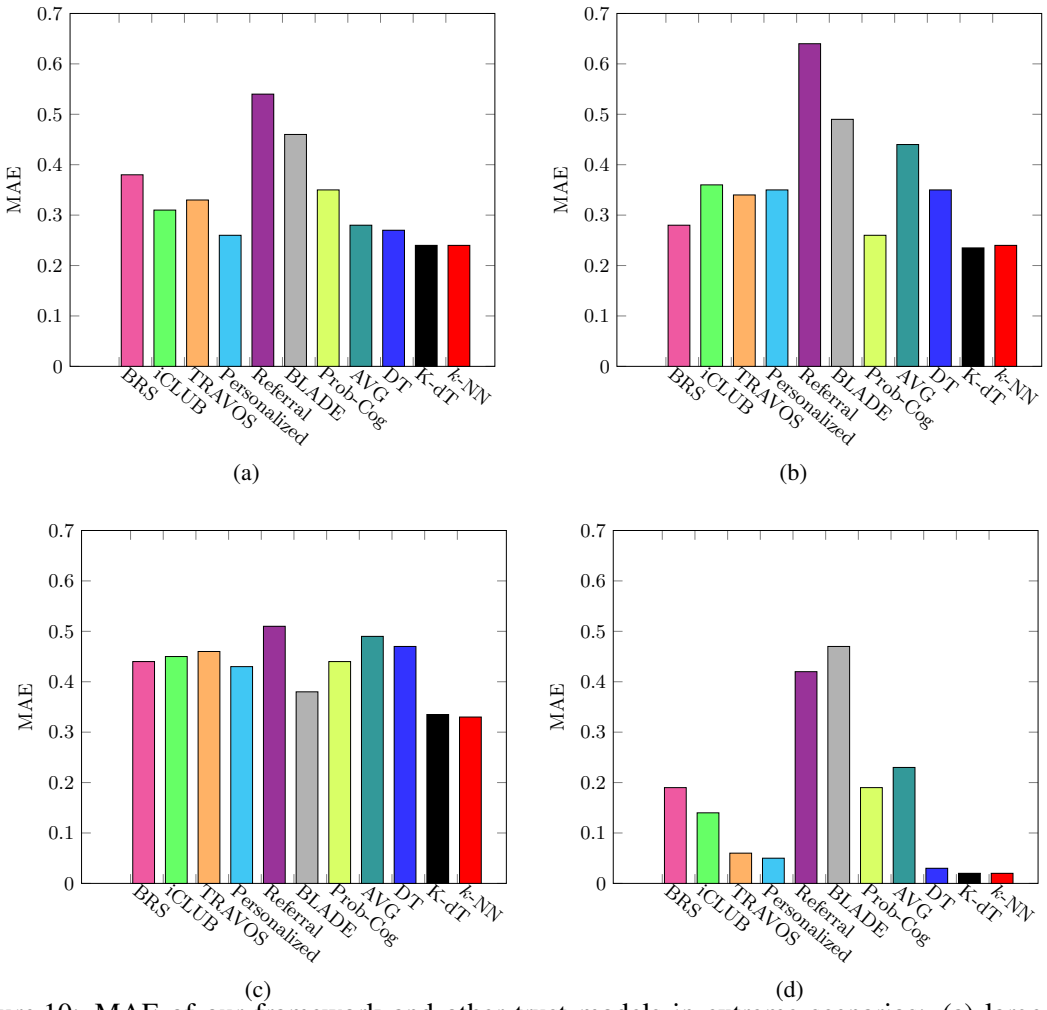



Figure 10: MAE of our framework and other trust models in extreme scenarios: (a) large environments; (b) extremely sparse environments; (c) environments with dynamic seller behavior and (d) environments with many attacks

**Performance Comparison in Extreme Scenarios** Fig. 10 shows the MAE of trust models in the 4 extreme scenarios (i.e., large environments, extremely sparse environments, environments with dynamic seller behavior and environments with many attacks). Table 4 presents the probability of choosing trust models in each of these 4 extreme cases, using $k$-NN retrieval technique (K-dT obtains almost the same probabilities as $k$-NN). Results show that $k$-NN and K-dT outperform DT, AVG and all the other trust models, while performing equally well in these environments.

More specifically, Fig. 10(a) shows that $k$-NN and K-dT obtain the lowest MAE of $0.24$ in large environments. We also find that iCLUB, TRAVOS and Personalized obtain smaller MAE than other trust models in these large environments. The reason is that these three trust models are able to distinguish dishonest and honest advisors when they get sufficient rating sources. From Table 4 under large environments, we can see that $k$-NN selects iCLUB, TRAVOS and Personalized with the highest probabilities, 26.9%, 23.8% and 38.2%, respectively. Fig. 10(a) also shows that DT and AVG obtain a larger MAE value ($0.27$ and $0.28$, respectively) than $k$-NN and K-dT.

From Fig. 10(b), in sparse environments, again $k$-NN and K-dT obtain the lowest MAE of $0.24$ and $0.23$ and DT obtains a MAE value of $0.35$. BRS and Prob-Cog perform better than other trust models, because BRS adopts the 'majority-rule' to consider the opinions from other advisors, and Prob-Cog extends the incompetence tolerance threshold to incorporate a larger number of advisors' ratings. Both models obtain a comparatively low MAE as they are less restrictive in accepting opinions from advisors than other trust models. In Table 4, under sparse environments, $k$-NN selects BRS and Prob-Cog with the highest probabilities, 25.7% and 52.7%, respectively.

Fig. 10(c) shows that $k$-NN and K-dT obtain the lowest MAE and Personalized and BLADE outperform other trust models in the environments where sellers change their behavior dynamically. To explain, Personalized considers advisors' latest ratings within a certain time window, which alleviates the influence of sellers' dynamic behavior. BLADE re-interprets advisors' ratings based on learning, thereby takes into account the changing behavior of buyers and sellers. In Table 4, $k$-NN selects Personalized and BLADE with the highest probabilities, 27.8% and 31.5%, respectively. Fig. 10(c) also signifies that our framework is able to deal with scenarios where buyers and sellers change their behavior, because the case base already contains environments representing such dynamic behavior (as described in Sec. 5.1), along with their most suitable trust models.

Fig. 10(d) shows that $k$-NN an K-dT outperform other trust models with the lowest MAE of $0.02$ and Personalized and TRAVOS perform well in the environments with many attacks. DT obtains an MAE of $0.03$, a comparable but greater value than $k$-NN and K-dT. The characteristics of attacks play a major role in judging the performance of the trust models. In these extreme environments, the attackers (dishonest advisors) first give honest ratings to non-target sellers to promote themselves, and then provide unfair ratings to bad-mouth target sellers. The performance of Personalized and TRAVOS is better because they both model advisor trustworthiness more accurately by comparing buyers' own opinions and advisors' ratings on commonly rated sellers. Also, in the environments, we select only buyers with sufficient personal experience (ratings) which Personalized and TRAVOS take advantage of. In Table 4, $k$-NN selects Personalized and TRAVOS with probabilities $58.7\%$ and $13.8\%$ for the environments with many attacks, respectively.

Table 4: Probability of choosing trust models in the four extreme e-market scenarios

| Trust Models | Large | Sparse | Dynamic | Many Attacks |
|---|---|---|---|---|
| BRS | 0.6% | 25.7% | 7.4% | 6.9% |
| iCLUB | 26.9% | 13.9% | 7.2% | 10.3% |
| TRAVOS | 23.8% | 2.1% | 4.6% | 13.8% |
| Personalized | 38.2% | 2.8% | 27.8% | 58.7% |
| Referral | 0.6% | 2.8% | 18.5% | 0.0% |
| BLADE | 9.3% | 0.0% | 31.5% | 0.0% |
| Prob-Cog | 0.6% | 52.7% | 3.0% | 10.3% |

In summary, from Fig. 10 and Table 4, the results indicate that our framework (using $k$-NN and K-dT retrieval technique) is able to select suitable trust models for extreme scenarios and obtain more accurate seller trustworthiness than AVG and any other individual trust model. Also, we find that the performance of $k$-NN and K-dT retrieval is better than DT retrieval in all the cases. Decision trees are induction models which learn rules (based on features) to determine suitable trust models. This comes close to the method that PTF (described in Sec. 2) works, the only difference being that in decision trees the rules are learned and organized in the form of trees, while in PTF the rules need to be manually specified by the user in a pre-defined format (Huynh, 2009). However, we can see that with the available number of simulated environments (2268) in the case base, decision trees

cannot learn the complete domain knowledge to construct trees which help to accurately determine suitable trust models. We can thereby infer that using a rule based system such as PTF will also result in such moderate performance as decision trees (with the given size of the case base).

**Analysis on the Possible Improvements to the Framework** It has been demonstrated in literature that feature weighting (assigning weights to individual features) after feature selection (selecting a subset of relevant features and ignoring others), can improve the performance of $k$-NN (Tahir et al., 2007). Thus, to further improve the performance of the framework (using $k$-NN and thereby, K-dT retrieval), we can assign weights to each (most influential) feature. We conduct experiments using the linear adaptive filters-least mean squares (Mitchell, 1997), with learning rate $0.2$, to determine the weights for the features, using $972$ randomly generated environments. The weights for the $13$ most influential features (in the same order as C4 in Table 1), determined using $k$-NN are, $0.16$, $0.02$, $0.01$, $0.02$, $0.03$, $0.2$, $0.09$, $0.05$, $0.04$, $0.1$, $0.17$, $0.01$ and $0.1$, respectively. We use the same weights for K-dT and analyze its performance. Table 5 shows the performance of $k$-NN and K-dT, using the above feature weights while calculating the similarity between the environments in order to determine the most suitable trust model. $k$-NN obtains an improvement of $1.0\%$ and $2.0\%$ in terms of accuracy in selecting the suitable trust model for unknown random and real environments (when compared to the values in Table 3, where $k$-NN and K-dT assign equal weights to all the $13$ influential features), respectively. For extreme scenarios, $k$-NN obtains an improvement of (at most) $2.0\%$. K-dT obtains a similar accuracy improvement of $0.5\%$ and $2.0\%$ for unknown random and real environments, while for the extreme scenarios, it obtains an improvement of (at most) $2.0\%$.

Table 5: Influence of using feature weights in $k$-NN and K-dT

| $k$-NN + feature weights | Random | Real | Large | Sparse | Dynamic | Attacks |
|---|---|---|---|---|---|---|
| Correct Model | 98.0% | 89.0% | 93.0% | 90.0% | 97.0% | 86.0% |
| Correct Model with $\epsilon$ | 98.5% | 98.0% | 95.0% | 98.0% | 97.0% | 92.0% |
| Correct Models and Paras | 97.0% | 82.0% | 95.0% | 96.0% | 97.0% | 80.1% |
| Correct Model and Paras with $\epsilon$ | 98.0% | 97.2% | 96.1% | 97.0% | 98.0% | 95.4% |
| MAE | 0.23 | 0.01 | 0.22 | 0.22 | 0.30 | 0.01 |
| Accuracy Improvement | 1.0% | 2.0% | 1.0% | 1.0% | 2.0% | 0.0% |
| K-dT + feature weights | Random | Real | Large | Sparse | Dynamic | Attacks |
| Correct Model | 97.5% | 89.0% | 94.0% | 90.2% | 97.1% | 85.0% |
| Correct Model with $\epsilon$ | 98.5% | 97.0% | 95.0% | 97.0% | 97.0% | 92.0% |
| Correct Models and Paras | 97.0% | 82.0% | 95.0% | 96.0% | 98.0% | 80.0% |
| Correct Model and Paras with $\epsilon$ | 98.0% | 97.2% | 96.1% | 97.0% | 98.0% | 96.0% |
| MAE | 0.23 | 0.02 | 0.21 | 0.22 | 0.30 | 0.01 |
| Accuracy Improvement | 0.5% | 2.0% | 1.0% | 2.0% | 2.0% | 0.0% |

Also, as mentioned in Sec. 4, the framework can be extended by adding new features and trust models. Specifically, to add a new feature we need to: 1) generate a new set of $E_{known}$ environments, including the new feature; 2) select the most influential features $\hat{F}^*$ (using the $5$ correlation and regression techniques as mentioned in Table 1), from the new extended feature set, by testing them in randomly generated $E_{test}$ environments, and 3) build the new case base. Thus, the time complexity for adding a new feature is $O((|E_{known}| + |E_{test}|) * \mathbb{TM} + 5 * t_{model})$, which includes the time taken to find the actual performance of all the defense models $\mathbb{TM}$ in the known $|E_{known}|$ and test $|E_{test}|$ environments (represented by $(|E_{known}| + |E_{test}|) * \mathbb{TM}$) and the time taken by $t_{model}$ ($model \in \{k\text{-NN, K-dT}\}$) to find the most suitable trust models using the $5$ different feature combinations (represented by $5 * t_{model}$). If $E_{known} = 2268$, $E_{test} = 972$, $\mathbb{TM} = 45$, $k = 3$ and model=K-dT, then the total time taken to build the new case base on adding a new feature is nearly 3 hours. Adding a new trust model simply takes 3.6 mins ($O(|E_{known}|)$), as it only requires to run the new

trust model on the 2268 environments to build the new case base. Though the above calculations show that a considerable computation time is involved, adding a new feature or a trust model will lead to an improvement in the performance of the framework. Specifically, adding a new feature can help to more accurately select suitable trust models and adding a new trust model may result in a lower MAE for certain environments, leading to better decision making and thereby improving the utility for the buyers in the environment as shown in Fig. 8. Also, all the above computation need to be done off-line and the online effort is much lower as can be seen in Fig. 9.

## 6. Conclusion and Future Work

In this paper, we propose a case-based reasoning framework to choose suitable trust models for the environments where ground truth about agents' behavior is unknown. In the framework, the case base is built by generating a number of simulated environments and determining the most suitable trust models for the environments. The framework also offers to choose between different techniques ($k$-nearest neighbors, K-d trees and decision trees) for case retrieval. Given an unknown environment, the most similar case(s) are retrieved using the retrieval techniques. Then, the trust model corresponding to the most similar case(s) is chosen as the most suitable one for the unknown environment. Experimental results confirm that our framework can accurately select suitable trust models for various unknown environments (both simulated and real e-marketplaces). We also find that $k$-nearest neighbors and K-d tree retrieval techniques can more accurately choose suitable trust models by determining the most similar case(s) from the case base than decision trees, especially when the number of simulated environments (cases in the case base) is much smaller. It is also demonstrated that K-d trees significantly improve the time complexity in choosing suitable trust models over $k$-nearest neighbors. Experiments also verify that using our framework to choose trust models for unknown environments (using $k$-nearest neighbors and K-d tree retrieval) is better than always applying any single trust model (or the aggregate of all trust models), in terms of the accuracy in evaluating seller trustworthiness.

Currently, the framework achieves the best performance when the number of simulated environments in the case base is as large as 2268 environments; the performance will further increase by adding more simulated environments. While adding more simulated environments is a feasible option to further improve the performance of the framework, it requires tremendous off-line computation to determine suitable trust models for the simulated environments and build the case base. In the future, we will investigate methods to generate simulated environments that are more representative of real world e-marketplaces, such that the performance of the framework is much higher even when the case base contains smaller number of simulated environments. We will also analyze more effective feature selection techniques to accurately select trust models in this regard.

Another important direction of future work is to consider the scenario when the features of the unknown environment deviate from the most similar simulated environment determined by the framework, during its execution time. One possible solution is to use the proposed framework to choose the most suitable trust model over regular intervals of time when the unknown environment operates. We will conduct detailed experiments to analyze the performance of the framework in such scenarios. We will also continue to evaluate our framework by incorporating more sophisticated trust models and involving more real data sets.

## References


Aamodt, A., & Plaza, E. (1994). Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, *7*(1), 39–59.

Ahmed, Y. S. (2004). *Multiple Random Projection For Fast, Approximate Nearest Neighbor Search in High Dimensions*. Ph.D. thesis, University of Toronto.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*, Vol. 3. Wiley.

Fullam, K. K., & Barber, K. S. (2007). Dynamically learning sources of trust information: Experience vs. reputation. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Hang, C. W., Wang, Y., & Singh, M. P. (2009). Operators for propagating trust and their evaluation in social networks. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*, Vol. 2. Springer.

Hoffman, K., Zage, D., & Nita-Rotaru, C. (2009). A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, *42*(1), 1.

Huynh, T. (2009). A personalized framework for trust assessment. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*.

Irissappane, A. A., Jiang, S., & Zhang, J. (2013). A framework to choose trust models for different e-marketplace environments. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*.

Irissappane, A. A., Oliehoek, F. A., & Zhang, J. (2014). A POMDP based approach to optimally select sellers in electronic marketplaces. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Irissappane, A. A., & Zhang, J. (2014). A testbed to evaluate the robustness of reputation systems in e-marketplaces. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Jøsang, A., & Ismail, R. (2002). The Beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*.

Liu, S., Zhang, J., Miao, C., Theng, Y., & Kot, A. (2011). iCLUB: An integrated clustering-based approach to improve the robustness of reputation systems. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.

Noorian, Z., Marsh, S., & Fleming, M. (2011). Multi-layer cognitive filtering by behavioral modeling. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Quinlan, J. R. (1996). Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research (JAIR)*, *4*(1), 77–90.

Quinlan, J. R. (1993). *C4. 5: Programs for machine learning*, Vol. 1. Morgan kaufmann.

Regan, K., Poupart, P., & Cohen, R. (2006). Bayesian reputation modeling in e-marketplaces sensitive to subjectivity, deception and change. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Sabater, J., & Sierra, C. (2005). Review on computational trust and reputation models. *Artificial Intelligence Review*, *24*(1), 33–60.

Soltani, S. (2013). Case-based reasoning for diagnosis and solution planning. *Technical Report, Queen's University*.

Sormo, F., Cassens, J., & Aamodt, A. (2005). Explanation in case-based reasoning–perspectives and goals. *Artificial Intelligence Review*, *24*(2), 109–143.

Tahir, M. A., Bouridane, A., & Kurugollu, F. (2007). Simultaneous feature selection and feature weighting using hybrid tabu search/$k$-nearest neighbor classifier. *Pattern Recognition Letters*, *28*(4), 438–446.

Teacy, W., Patel, J., Jennings, N., & Luck, M. (2006). TRAVOS: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multiagent Systems*, *12*, 183–198.

Vempala, S. S. (2012). Randomly-oriented $k$-d trees adapt to intrinsic dimension. In *Proceedings of the 32nd International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*.

Wang, Y., Hang, C.-W., & Singh, M. P. (2011). A probabilistic approach for maintaining trust based on evidence. *Journal of Artificial Intelligence Research*, *40*(1), 221–267.

Watson, I. (1999). Case-based reasoning is a methodology not a technology. *Knowledge-based systems*, *12*(5), 303–308.

Watson, I., & Marir, F. (1994). Case-based reasoning: A review. *Knowledge Engineering Review*, *9*(4), 327–354.

Wess, S., Althoff, K.-D., & Derwand, G. (1994). *Using $k$-d trees to improve the retrieval step in case-based reasoning*. Springer.

Wettschereck, D., & Aha, D. W. (1995). Weighting features. *Case-based Reasoning Research and Development*, 347–358.

Whitby, A., Jøsang, A., & Indulska, J. (2004). Filtering out unfair ratings in bayesian reputation systems. In *Proceedings of the AAMAS Workshop on Trust in Agent Societies (TRUST)*.

Yang, Y., Feng, Q., Sun, Y. L., & Dai, Y. (2008). RepTrap: A novel attack on feedback-based reputation systems. In *Proceedings of the International Conference on Security and Privacy in Communication Networks (SecureComm)*.

Yu, B., & Singh, M. (2003). Detecting deception in reputation management. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Zhang, J., & Cohen, R. (2008). Evaluating the trustworthiness of advice about seller agents in e-marketplaces: A personalized approach. *Electronic Commerce Research and Applications*, *7*(3), 330–340.

Zhang, J. (2009). *Promoting honesty in e-marketplaces: Combining trust modeling and incentive mechanism design*. Ph.D. thesis, University of Waterloo.