Automatically Detecting Criminal Identity Deception: An Adaptive Detection Algorithm

G. Alan Wang, *Student Member, IEEE*, Hsinchun Chen, *Fellow, IEEE*, Jennifer J. Xu, and Homa Atabakhsh

***Abstract*—Identity deception, specifically identity concealment, is a serious problem encountered in the law enforcement and intelligence communities. In this paper, the authors discuss techniques that can automatically detect identity deception. Most of the existing techniques are experimental and cannot be easily applied to real applications because of problems such as missing values and large data size. The authors propose an adaptive detection algorithm that adapts well to incomplete identities with missing values and to large datasets containing millions of records. The authors describe three experiments to show that the algorithm is significantly more efficient than the existing record comparison algorithm with little loss in accuracy. It can identify deception having incomplete identities with high precision. In addition, it demonstrates excellent efficiency and scalability for large databases. A case study conducted in another law enforcement agency shows that the authors' algorithm is useful in detecting both intentional deception and unintentional data errors.**

***Index Terms*—Efficiency, identity deception, missing value, scalability.**

I. Introduction

IDENTITY deception occurs when someone intentionally conceals his/her original identity, impersonates another individual's identity, or uses forged identity documents. One of the problems that identity deception may cause is financial loss. For example, the U.K. reports financial losses of at least £1.3 billion each year due to identity deception [1]. More importantly, criminals or terrorists using false identities may cause casualties and property damages too large to be quantifiable. Thus, the identity deception problem has become a central issue in law enforcement and intelligence agencies.

A fabricated identity is difficult for law enforcement or intelligence agents to uncover. Police officers often rely on computer systems to search a suspect's identity against history records in police databases. Generally, computer systems search using exact match queries. Even if the fabricated identity is similar to the original identity recorded in the law enforcement computer system, an exact-match query is unlikely to bring up that record. Techniques to perform inexact searches have been developed. They can be used to detect deceptive identities by finding records that are similar but not exactly the same. However, most of these techniques are *ad hoc* and cannot be easily applied to real deception detection applications because of problems such as missing values and large volumes of data. Because a police database usually contains millions of criminal identity records, the detection techniques need to be efficient and scalable enough to examine all deceptive identities. In addition, for any large dataset, it is "unlikely that complete information will be present in all cases" [23]. Missing values contained in past criminal records may greatly affect the accuracy of the detection techniques in finding deceptive identities because of the reduced information.

In this paper, we aim to develop an automated approach that looks for inexact matches for fabricated identities. Such a technique is expected to search through past criminal identity records that may contain missing values and to be efficient enough to handle large volumes of data. In Section II, we briefly discuss the identity deception problem and review some existing deception detection techniques. We also review techniques that handle the missing value problem and those that improve algorithm efficiency and scalability. We present our research questions in Section III. In Section IV, we propose an adaptive detection algorithm for identity deception problems. This algorithm is able to utilize records containing missing values and is scalable to large volumes of identity data. We describe our experimental design in Section V and report the results and discussions in Section VI. We conclude our findings and future directions in the last section.

II. Related Work

A. Identity Deception

Identity is a set of characteristic elements that distinguish a person from others [12], [22]. There are three types of basic identity components, namely: 1) attributed identity; 2) biometric identity; and 3) biographical identity [1], [9]. Attributed identity is the information given to a person at birth, such as name and date and place of birth. Biometric identity contains biometric features that are unique to a person, such as fingerprints. Information that builds up over a life span comprises a person's biographical identity, examples of which are credit history and crime history. Among these three types of identity components, attributed and biographical identities are often subject to deception, whereas biometric features of a person are the most difficult to falsify.

Deception is "a sender's knowingly transmitting messages intended to foster a false belief or conclusion in the receiver"

Manuscript received November 18, 2003; revised July 23, 2004. This work was supported by the National Science Foundation, Digital Government Program, "COPLINK Center: Social Network Analysis and Identity Deception Detection for Law Enforcement and Homeland Security," IIS-0429364, 2004–2006.

G. A. Wang, H. Chen, and H. Atabakhsh are with the Department of Management Information Systems, University of Arizona, Tucson, AZ 85721 USA (e-mail: gang@eller.arizona.edu; hchen@eller.arizona.edu; homa@eller.arizona.edu).

J. J. Xu is with the Department of Computer and Information Systems, Bentley College, Waltham, MA 02452 USA (e-mail: xu@bentley.edu).

Digital Object Identifier 10.1109/TSMCA.2006.871799

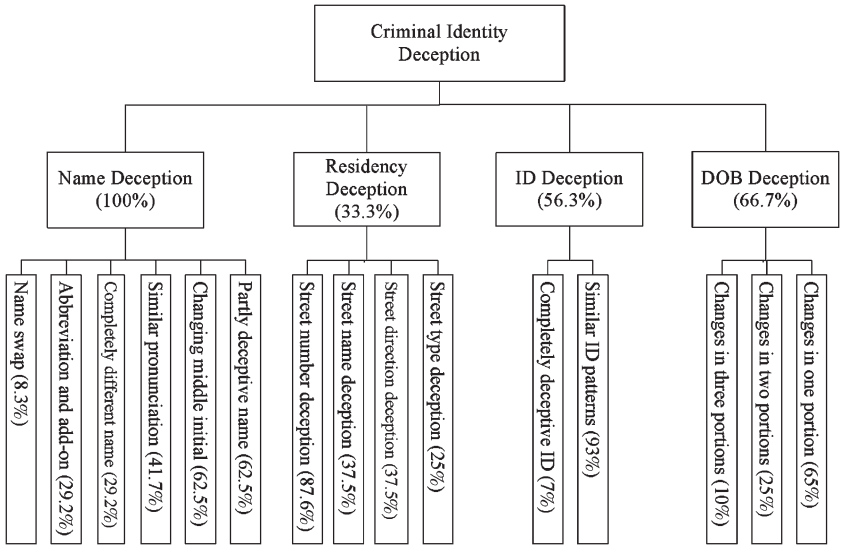


Fig. 1. Taxonomy of identity deception. Each percentage number represents the proportion of cases that contain the particular type of deception.

[7]. This definition originates from the interpersonal communication perspective and also applies to identity deception that usually occurs in an interactive environment (e.g., during an interrogation). We categorize three types of identity deception based on the method of deception, namely: 1) identity concealment; 2) identity theft; and 3) identity forgery.

Identity concealment is deceiving by omitting or changing details of the true identity [11]. For example, a person may report his birth date with an altered month or day or provide a false first name along with his true last name. This type of deception is popular when a subject unexpectedly encounters a law enforcement officer [15]. Concealment could be more advantageous than using a completely fictitious identity to those who lie about their identities. Subjects may recall partially true information more easily than a completely fictitious identity when questioned repeatedly because the true part of the concealed information serves as recall cues and cued recall may reconstruct memory better than recall without cues (i.e., free recall) [10]. Hence, the difficulty of recognizing such a deception (e.g., by law enforcement agents) is substantially increased. Identity theft, also called impersonation, is the action of one person illegally using another person's identity information for fraudulent purposes. Credit card fraud is a good example of identity theft. Identity forgery is committed through the use of forged or faked identity documents such as birth certificates, social security cards, and passports. This is common for illegal aliens who need forged documents to stay unnoticed and, yet, make a living [37].

In this paper, we mainly focus on the problem of identity concealment. We believe a solution to this problem can greatly improve crime investigation by law enforcement and intelligence agencies. We also hope that the solution proposed will be of value in detecting identity theft as well as forgery.

We provided evidence for the existence of identity concealment in [39], in which a taxonomy of identity deception (Fig. 1) was built upon a case study of real criminal identity deception. We found that deception mostly occurs in specific attributes, namely, name, address, date of birth (DOB), and ID number [e.g., the Social Security Number (SSN)]. Name concealment, occurring in most deceptive cases, includes giving a false first name and a true last name or vice versa, changing the middle initial, giving a name pronounced similarly but spelled differently, etc. Concealment made on DOB can consist of, e.g., switching places between the month of birth and the day of birth. Similarly, ID deception is often made by changing a few digits of a social security number or by switching their places. In residency deception, criminals usually change only one portion of the address. For example, the case study found that in about 87% cases, subjects provided a false street number along with the true street direction, name, and type.

Based on this case study, we observed that a concealed identity often partially matched with its original identity. We studied whether a certain technique could utilize such a characteristic and automatically detect this type of identity deception. In the next section, we review techniques that can be used to detect identity deception.

B. Deception Detection Techniques

Detection techniques for general deception have been developed in the behavioral research fields, such as psychology, physiology, and communication. Techniques include the analysis of verbal cues (symptoms of verbal content that are used to determine truth and deception), observing nonverbal cues (indications conveyed through nonverbal communication channels such as facial expression), and measuring physiological reactions (e.g., polygraph lie detector) [3], [14], [38]. However, detection results from these techniques are quite unreliable [11], [13], [24], [25]. Moreover, these techniques are not automated processes and require human operators.

Practical detection techniques for identity deception are developed in law enforcement and intelligence communities. First, police officers often use techniques such as repeated questioning and detailed questioning to validate the truthfulness of a suspect's identity. During the questioning process, inconsistent answers may disclose a false identity. However, those

questioning methods are not reliable techniques, especially when dealing with good liars. Consequently, many deceptive records still exist in law enforcement databases. Second, after talking to the crime analysts of Tucson Police Department (TPD), we find that professional crime analysts can sometimes detect deceptive identities using link analysis techniques. By examining associations among criminals, organizations, and vehicles, a crime analyst is able to build criminal networks. When information about a suspect's identity is incompatible with known relationships represented in the criminal networks, the identity will be flagged as a possible deception. This technique, however, requires great amounts of manual information processing and is very time-consuming. In fact, it often serves as a postinvestigative tool rather than a proactive investigation technique.

Some techniques that were initially designed for crime analysis can be used to detect identity deception. These techniques basically perform data association that links suspects to the crime being investigated, ordered from the most possible to the least possible. Brown and Hagen [5] proposed a similarity-based data association method for associating records of the same suspect or incidents having similar modus operandi (MO). It compares corresponding description attributes of two records and calculates a total similarity measure between the two records. Experiments showed that associations suggested by the algorithm agreed with those made by experts. Both techniques introduced above are automated processes and can be used to detect identity deception by associating a suspect's identity with past criminal records. However, these methods only define similarity measures for categorical (e.g., hair color) and quantitative (e.g., height) attributes, but not for textual noncategorical attributes such as name and address.

A record comparison algorithm specifically targeting the detection of identity deception was proposed in our previous paper [39]. This automated detection method makes use of string comparison techniques and searches for inexact matches of suspects' identities in police databases. This technique examines the attributes of name, address, DOB, and SSN for each identity. It computes a disagreement measure between values in each corresponding attribute of two identities and calculates an overall disagreement value between the two identities as an equally weighted sum of the attribute disagreement measures. The formula for the overall disagreement value is as follows:

$$d = \sqrt{\frac{d_{\mathrm{Name}}^2 + d_{\mathrm{Addr}}^2 + d_{\mathrm{SSN}}^2 + d_{\mathrm{DOB}}^2}{4}} \tag{1}$$

where d_{Name}, d_{Addr}, d_{SSN}, and d_{DOB} represent the disagreement measures in the fields of name, address, SSN, and DOB, respectively. Each field value is considered a string of characters. Disagreement between two field values is computed by a string comparator, namely, the Levenshtein edit distance [26], which calculates the minimum number of single-character insertions, deletions, and substitutions required to transform one string to the other. Dividing the edit distance by the length of the longer string, each disagreement value is normalized between 0 and 1. If an overall disagreement value d between a suspect's identity and a past identity record is less than a threshold, which can be predetermined by a training process, the algorithm suggests that one identity is a deceptive form of the other. Experiments showed that this algorithm achieved high detection accuracy (94%). However, this method is quite inefficient for large-scale datasets. The computational time complexity of the algorithm is $O(N^2)$ because it compares each pair of records in a dataset. The computational time will increase exponentially as the size of the dataset increases. Furthermore, this method is unable to deal with identities that do not have values in all four fields (i.e., containing missing values).

The record comparison algorithm works better than data association algorithms for detecting identity deception because it specifically captures the concealment deception patterns defined in the taxonomy introduced in the previous section. However, the problems with the record comparison algorithm, namely, the inability to handle missing values and the inefficiency in processing large data volumes, prevent it from being used in any real-world applications. In the next two sections, we review techniques that handle the missing value problem and methods that improve the algorithm efficiency.

C. Missing Value Problem

Missing values are defined as values excluded from arithmetic calculations because they are missing [8]. In statistical analysis and data mining fields, there are three major types of strategies that deal with the missing value problem, namely: 1) deletion; 2) imputation; and 3) adaptive data analysis.

Deletion (listwise or pairwise deletion) [6], [16], [18], [23] is the simplest technique to overcome the missing value problem and is easy to implement. Listwise deletion deletes or ignores those data records where missing values occur. Pairwise deletion only excludes records missing information on the variables under examination [17]. Both approaches may result in a great amount of information loss if the fraction of missing values is high [17], [40]. Also, deletion methods may lead to serious statistical biases if the missing values are not randomly distributed [35].

Another alternative is imputation, which fills in missing values with plausible estimates [2], [35]. Such a technique makes use of patterns or statistical associations found in complete records. These patterns are then applied to records with missing values, making estimates of the missing values in each record based on known attribute values. For example, mean imputation [33] replaces a missing value with the mean of nonmissing values of the same attribute. Some imputation methods can be complex due to the process of finding statistical patterns [31]. However, imputation techniques can only make estimates on numeric or categorical attributes, upon which statistical patterns can be built. Textual attributes, such as names or addresses, can hardly be estimated. Another disadvantage of imputation methods is potentially biasing datasets by treating artificially imputed values as real ones in subsequent data analysis [30].

In cases where imputation methods cannot reasonably estimate, adaptive data analysis methods are usually developed to minimize the impact of missing values. Timm and Klawonn [36] gave an example with the fuzzy c-means clustering algorithm, in which missing values were omitted and known ones

were taken into account in calculating the center of each cluster. Quinlan [32] developed an adaptive approach for missing values in decision tree problems. He reduced the information gain from testing an attribute A by the proportion of cases with missing values of A. Experiments showed that this approach performed better than that of dropping all incomplete cases (i.e., listwise deletion).

In conclusion, listwise or pairwise deletion is not always desirable because they lead to great information loss when there are many missing values. For the problem of identity deception, imputation methods are not appropriate because identity attributes such as names and addresses are textual attributes to which imputation techniques simply do not apply. Therefore, an adaptive data analysis method suitable for our scenario needs to be developed to fully utilize the known attribute values and minimize the impact of those that are unknown.

D. Algorithm Efficiency and Scalability

The efficiency and scalability problem impacts many algorithms that process large amounts of data, such as algorithms for finding duplicate records from large databases involving millions of records. To find all duplicate records in a database, the most reliable way is to compare every record with every other record [27]. Such a method apparently is the most inefficient, especially when it is applied to large databases, because of its time complexity $(O(N^2))$.

Much database research has focused on data comparison efficiency. Hernandez and Stolfo [20] presented a sorted neighborhood method (SNM) for the so-called merge/purge problems, in which data were merged from multiple sources. The SNM has three steps, namely: 1) creating sorting keys; 2) sorting data; and 3) merging duplicates. A key is made by extracting a relevant attribute or a combination of relevant attributes. The selection of a key, determined mainly by domain-dependent knowledge, is critical for final merging results [21]. The dataset is then sorted by the selected key in the sorting phase. During the merging phase, a window of a fixed size sequentially moves through the sorted dataset from the top. Every new record entering the window compares with the previous records in the window and looks for matching records. To maintain the fixed window size, the first record in the window is dropped when a new record enters a full window. The time complexity of the SNM is $O(wN)$ (the time complexity of the merging phase) if $w < \log N$, or else $O(N \log N)$ (the time complexity of the sorting phase), where w is the window size and N is the total number of records in the dataset. Experiments showed that the SNM could achieve high detection accuracy and greatly reduce running time. The SNM assumes that duplicate records sorted by an appropriate key are located close to each other, which is not always the case. One may increase the window size to find potential duplicates; however, this may increase the running time as well.

Monge [28], [29] proposed an adaptive duplicate detection algorithm that further improved the detection efficiency over the SNM. Like the SNM, this method also starts by creating a sorting key and sorts the dataset with the key. Whereas a window sequentially scans the sorted dataset, it does not compare each newly entering record with all existing records in the window. If there are duplicate records existing in the window, the newly entering record only compares with one of them and others are ignored. Therefore, the actual number of comparisons w' that a newly entering record makes within the window varies. The time complexity of this algorithm is $O(w'N)$, where w' is usually less than the window size w. Consequently, this adaptive detection method is much more efficient than the SNM. Experiments showed that the detection accuracies of both methods were similar [28].

III. Research Questions

In this paper, we aim to develop a technique that can automatically detect deceptive criminal identities in law enforcement and intelligence databases in an effective and efficient way. Such a technique is applicable to the following law enforcement scenarios.

1) Given a suspect's possibly false identity, the algorithm is able to locate relevant identity records of the same individual in police databases. Therefore, the true identity of the suspect may be recovered, and more information becomes available to assist the police investigation.
2) The algorithm detects deceptive identities by examining records currently existing in police databases. This requires an efficient algorithm that deals with large data volumes, especially when data are integrated from different sources.

We have identified a record comparison algorithm that is most appropriate for detecting identity deception. We aim to improve this algorithm using techniques that allow it to deal with missing values and make it efficient and scalable with large data volumes. Our research questions are as follows.

1) Can the improved technique effectively detect deceptive identities with records having missing values?
2) Is the improved technique efficient and scalable enough to handle the large amount of identities in police databases while the detection accuracy is maintained?

IV. Adaptive Detection Algorithm

We aim to develop a detection algorithm that can adapt to real-world applications where missing values are prevalent and data volume is often on the order of millions. In this section, we propose an adaptive detection algorithm for detecting identity deception. We use an improved version of the record comparison algorithm's process, so that identities containing missing values can be compared based on known attributes. The new algorithm also incorporates the heuristics of Monge's adaptive duplicate detection method. We expect the efficiency of the detection process to be highly improved.

We choose to use an adaptive analysis method to handle the problem of missing values. Our intention is to make use of as many known attribute values as possible and to ignore missing values. Deletion methods discard not only attributes that have missing values but also some attribute values that are not missing. Statistics-based imputation methods try to impute missing values based on the statistical relationship between

attribute values that are missing and those that are not. However, they require attributes to be either quantitative or categorical, so that statistical relationship can be established. In our case, most of the attributes (e.g., name and address) are textual. Statistical relationships between these attributes do not make sense (e.g., it would be strange to conclude that people named "George" usually live on "Broadway Blvd.").

In the pairwise record comparison algorithm, identity records containing missing values are simply discarded (i.e., listwise deletion). In the proposed adaptive detection algorithm, only the missing attributes are ignored, whereas other available attributes are used in comparing a pair of identities. Here, we assume that every two identities being compared have at least one nonmissing attribute. We also assume that two matching identities have similar values on all attributes. We modify the original formula given in the previous section as

$$d' = \sqrt{\frac{d^2_{\text{Name}} + d^2_{\text{Addr}} + d^2_{\text{SSN}} + d^2_{\text{DOB}}}{a}} \tag{2}$$

where a is the number of attributes that are available in both identity records being compared. The disagreement measures on missing attributes are set to zero. The heuristic is similar to what police officers would do when they manually compare two identities. It is obvious that the higher the number of missing values, the less confident the overall disagreement is.

We apply Monge's algorithm to our proposed algorithm to improve efficiency. The first step of Monge's algorithm is to sort the dataset according to a key attribute. Sorting on some attributes may lead to better results than sorting on the others. The key attribute can be determined by a training process. However, no single key will be sufficient to catch all matching records in general [21]. Hernandez and Stolfo suggested a multipass approach that executes several independent runs of the algorithm, each time using a different key attribute. On the other hand, the multipass approach will increase the computation time. In this study, we only consider the single-pass approach.

The procedure for the revised detection method is shown in Fig. 2. First, the whole dataset is sorted by a chosen key attribute. The window size w is set in step 2, which defines the range of nearby records being compared. The window is represented as a priority queue, which can contain at most w elements (i.e., clusters). The algorithm sequentially examines each record R_i in the sorted dataset starting from the top. In step 7, R_i is first compared with the representative record (the record that represents the cluster; we use the first record of each cluster to simplify the process) of each existing cluster C_j in a priority queue q. If a comparison suggests a match (i.e., the disagreement value of the two records is less than a given threshold) between R_i and C_j's representative, R_i will be merged into C_j. If R_i fails to find a match, it will continue to compare with the nonrepresentative records (i.e., records except the first one) of each C_j in q. If a match is found, R_i will be merged into the cluster where the matched record belongs. If R_i cannot be merged into any cluster in q (such as in the beginning when clusters do not exist in q), a singleton cluster is created for R_i in step 19 and is inserted into q in step 23. The lowest priority cluster in q (i.e., the cluster first put in the queue) will be dropped from q if a new cluster is inserted into an already full queue. If a dropped cluster contains more than one identity record, this indicates that deceptive identities are found.

```
procedure AdaptiveDetection ()
1:  Sort the data set according to a key field;
2:  Set a window size w;
3:  Create a priority queue q of size w;
4:  LOOP: record Ri in sorted dataset   //scan the sorted dataset sequentially
5:      IF: Ri is not a member of any clusters in q
            //Compare Ri with the representative record of each cluster in q
6:          LOOP: cluster Cj in q
7:              IF Distance(Ri , Representative(Cj)) < threshold
8:                  Union(Ri, Cj);   //include Ri to the cluster C
9:                  GOTO step 4
10:             END IF
11:         END LOOP
            //if no match is found,
            //compare Ri with the rest records of each cluster in q,
12:         LOOP: cluster Cj in q
13:             LOOP: R in cluster Cj
14:                 IF Distance(Ri , R) < threshold
15:                     Union(Ri, Cj);   //include Ri to the cluster Cj
16:                     GOTO step 4
17:                 END IF
                END LOOP
18:         END LOOP
            //if no match is found, create a new cluster for Ri and enque
19:         Cnew = NewCluster(Ri);
            //if q is full, dequeue the cluster that first entered q
20:         IF: Size(q) = w
21:             q.Deque();
22:         END IF
23:         q.Enque(Cnew);
24:     END IF
25: END LOOP
end AdaptiveDetection
```

Fig. 2. Procedures of the adaptive detection algorithm.

An example would make this clustering process much easier to understand. Suppose the dataset is sorted on name and the window size w (i.e., the capacity of the priority queue q) is set to 4. We start to look at the first record R_0 from the top of the sorted dataset. Because q is empty at the beginning, we do not have any clusters to compare against. Therefore, a new cluster C_0 is created with R_0 as its only record and is put in q. We then examine the next record R_1. We first compare R_1 with the representative record (R_0) of the only cluster C_0 in q (step 7). Suppose R_1 matches R_0 (i.e., the disagreement value of the two records is less than a given threshold), we include R_1 in C_0 (step 8) and go back to step 4 to examine the next record R_2. Similarly, R_2 is first compared with R_0, the representative record of cluster C_0. If the two records do not match, R_2 is compared with R_1, the nonrepresentative record in C_0 (step 14). If R_2 and R_1 match, R_2 is included in C_0. If they do not match, a new cluster C_1 is created with R_2 as its only record and becomes the second element in q. This procedure is repeated until all records are examined. The first cluster (e.g., C_0) will be removed from q when q is full (i.e., the number of clusters in q is equal to w). Therefore, a new record will only be able to compare the records contained in q.

The time complexity of the proposed adaptive detection method becomes $O(w'N)$ (the time complexity of the merging phase) if $w' < \log N$, or otherwise $O(N \log N)$ (the time complexity of the sorting phase), where w' is the window size and N is the total number of records in the dataset. Compared to the

pairwise comparison algorithm, the adaptive detection method is expected to be much more efficient.

V. Experiments

In this section, we aim to test the effectiveness and the efficiency of the proposed adaptive detection algorithm. Experiments are conducted to answer the following questions.

1) Will the detection accuracy be maintained when employing the adaptive detection algorithm?
2) Can the adaptive detection algorithm detect deceptive identity records that contain missing values?
3) How does the adaptive detection algorithm perform with large datasets?

A. Performance Matrix

Algorithm performance is measured in terms of detection effectiveness and efficiency.

1) Detection Accuracy: We evaluate the algorithm's detection accuracy by using three kinds of measures, namely: 1) recall; 2) precision; and 3) F-measure. Those measures are widely used in information retrieval [34]. Precision, in this scenario, is defined as the percentage of correctly detected deceptive identities in all deceptive identities suggested by the algorithm. Recall is the percentage of deceptive identities correctly identified. F-measure is a well-accepted single measure that combines recall and precision.

Suppose a set of identities D contains m unique individuals and each individual has at least one identity. Each individual may have a set of different identities denoted as D_i ($1 \leq i \leq m$ and $|D_i| \geq 1$). Let d_{ij} ($1 \leq i \leq m, j \geq 1$) denote the jth identity of the ith individual. The detection algorithm groups all identities into n clusters based on identified identity deception. That is, deceptive identities that are considered as referring to the same individual by the detection algorithm are grouped into the same cluster. Each cluster identified by the algorithm is denoted as C

$$C_k = \{d_{ij} | d_{ij} \in D \text{ and } d_{ij} \text{ referring to the } k\text{th individual}\} \quad (3)$$

where $k = 1, 2, \ldots, n$. The clusters have the following properties:

$$C_k \cap C_{k\prime} = \varnothing$$
$$\bigcup_k C_k = D. \quad (4)$$

Identities of the same cluster are considered to refer to the same person, whereas identities of different clusters are considered irrelevant. To make performance measures of clustering results comparable to those of the pairwise comparison method, we convert the clustering results to a matrix that is often generated by the pairwise comparison method. For example, suppose person A has two different identities $\{A_1, A_2\}$, whereas person B has three identities $\{B_1, B_2, B_3\}$. Suppose the adaptive detection algorithm identifies two clusters, namely: $\{A_1, A_2, B_1\}$ and $\{B_2, B_3\}$. A pairwise comparison matrix is constructed from the clusters as shown in Fig. 3. Each

	A1	A2	B1	B2	B3
A1		1	1	0	0
A2			1	0	0
B1				0	0
B2					1
B3					

Fig. 3. Pairwise comparison matrix constructed from the two clusters.

TABLE I
Classification of Algorithm Outcomes

	Identities of the same person	**Identities of different persons**
Identities considered to refer to the same person	True Positive (TP)	False Positive (FP)
Identities considered not to refer to the same person	False Negative (FN)	True Negative (TN)

superdiagonal element in the matrix represents the comparison result between any two identity records. It is labeled as one when two identity records are grouped in the same cluster by the algorithm; otherwise, it is labeled as zero. We will have four outcomes defined in Table I. In this example, we have TP $= 2$, FP $= 2$, TN $= 4$, and FN $= 2$.

Based on the algorithm outcomes, we compute recall and precision as the following:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (6)$$

F-measure is defined as

$$F\text{-measure} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \quad (7)$$

2) Efficiency and Scalability: Efficiency is measured by the number of comparisons that the algorithm requires to detect all deceptive identities within a dataset. Algorithm completion time is a supplementary efficiency measure.

According to the Longman Web Dictionary, scalability of an algorithm can be defined as the degree to which the algorithm becomes more efficient as the data volume increases. We define scalability to be proportional to the number of identities processed per unit of time, i.e.,

$$\text{Scalability} \propto \frac{\text{Number of records in a dataset}}{\text{Completion time}}. \quad (8)$$

B. Experimental Design

In our experiments, we compared the performance of the proposed adaptive detection algorithm with that of the record comparison algorithm. We did not compare with the performance of other deception detection techniques because they are not directly comparable. We aim to examine how the algorithm's performance improves when incorporating techniques that handle the problems of missing values and large volumes of data. We expect that those techniques developed in the proposed algorithm will also apply to other computational deception

TABLE II
DIFFERENT MISSING TYPES IN IDENTITY RECORDS OF THE TPD

Missing type	Number of identity records	Percentage
Complete	311,151	24.139%
SSN-missing	540,849	41.960%
Address-missing	298	0.023%
DOB-missing	1,470	0.114%
SSN-Address-missing	293,595	22.777%
SSN-DOB-missing	83,952	6.513%
Address-DOB-missing	25	0.002%
SSN-Address-DOB-missing	57,634	4.471%
Total:	**1,288,974**	**100%**

detection techniques reviewed in Section II-B and will improve their performance.

The datasets of deceptive identities used in our experiments were manually extracted by our police detective expert who has served law enforcement for 30 years. The sampling method the expert used was convenience sampling, in which he looked through the list of all identity records and chose the deceptive identity records that he ran into. Because deceptive identities are sparsely distributed in the criminals' database, convenience sampling is more feasible than random sampling to locate deceptive identity records for experimental purposes.

1) Test Bed: We chose criminal identity records stored in the TPD as our test bed. According to the U.S. Census Bureau, Tucson's population ranked 30th among U.S. cities with populations of 100 000 and greater. The Federal Bureau of Investigation also reported that Tucson's crime index ranked 20th highest among U.S. cities in 2001 and was higher than the national average. Therefore, data kept in the TPD are representative of those stored in other agencies in terms of variety and data volume.

The TPD maintains about 1.3 million person records in the database. Each record uniquely identifies a person by a set of identity attributes. In this experiment, we only focus on four attributes in which identity deception usually occurs, namely: 1) name; 2) address; 3) DOB; and 4) SSN. The name attribute of each identity record is mandatory in the TPD and always has a value. We found a large number of missing values in the other three attributes; 76% of these records contain missing values in at least one attribute. Among these incomplete records, we found that 42% contain one missing attribute, 29% have two missing attributes, and 4% of the records were missing all attribute values except for name. The distribution of different missing types is shown in Table II. Certain missing types, such as address-missing, DOB-missing, and address-DOB-missing, are rare in the TPD database. Inasmuch as all fields except name can be missing in the TPD database, we chose name as the sorting key for the adaptive detection algorithm in hypotheses testing.

2) Hypotheses Testing: We expect the proposed adaptive detection algorithm, as compared with the pairwise record comparison algorithm, to improve its efficiency in detecting deceptive identities without losing detection accuracy. Although we do not expect detection accuracy to maintain when a dataset has several missing attributes and a large percentage of missing values, we want to find out what circumstances could cause significantly lower accuracy rates for incomplete datasets. We also aim to find out whether the adaptive detection algorithm can find deceptive identities within an acceptable time (e.g., in minutes) when the dataset is large (e.g., in the order of millions). The hypotheses for testing the above objectives are discussed below.

a) Evaluating accuracy and efficiency: We compare the performance of the adaptive detection algorithm with that of the record comparison algorithm. Two hypotheses are proposed to compare the efficiency and the detection accuracy of the two algorithms. We use statistical t-tests in the comparisons to indicate the significance of any differences.

— Hypothesis 1 (H1): There is no significant difference in detection effectiveness between the adaptive detection algorithm and the record comparison algorithm.
— Hypothesis 2 (H2): There is no significant difference in detective efficiency between the adaptive detection algorithm and the record comparison algorithm.
  - Testing dataset: A police detective with 30 years of experience helped us identify 210 deceptive criminal identity records from the TPD database. The dataset involved 75 criminal individuals, each of whom had an average of three identity records. These identity records contain no missing values. All the addresses were manually converted to a standard format consisting of a street number, a street direction, a street name, and a street type.
  - Testing procedure: A ten-fold validation method was employed to validate the performance of the two algorithms. The dataset was randomly equally divided into ten folds. Each time, we used nine folds for training and one fold for testing. In each training session, we determined an optimal threshold that distinguished between similar (i.e., deceptive) and dissimilar (i.e., irrelevant) records, when the highest F-measure was achieved. The threshold was then applied to the next testing session. Accuracy measures, as well as the number of comparisons and the completion time, were recorded for each testing session. Performance measures of the two algorithms were compared using a statistical t-test.

b) Evaluating the effects of missing values: We compare the detection accuracy of the algorithm when using a complete dataset and when using an incomplete dataset. Again, t-tests were used to indicate whether there was a significant difference in the algorithm's detection accuracy. To examine how different types of incomplete datasets may affect the algorithm's detection accuracy, we varied the missing attribute(s) (i.e., attributes where missing values may occur) in the dataset and the percentage of incomplete records in the dataset. We learned from the TPD database that identity records missing more than two attribute values are rare. Therefore, we tested with incomplete datasets having no more than two attributes containing missing values.

— Hypothesis 3 (H3): With the adaptive detection algorithm, there is no significant difference in detection effectiveness

between identities having all attribute values and identities having at most two missing attribute values.

- Testing datasets: First, we conducted experiments using artificial incomplete datasets. In the TPD database, deceptive identities with certain missing attributes (e.g., DOB-missing or address-DOB-missing) are rare. With artificially generated incomplete datasets, we constructed various types of incomplete datasets by adjusting the composition of missing attributes as well as the percentage of incomplete records in each dataset. Incomplete datasets were derived from the complete dataset used in the previous experiment. For each dataset, we randomly chose a percentage (from 10% to 90% with an increment of 10%) of records from which we removed values in the intended missing attribute(s). Second, we used a real incomplete dataset that was directly extracted from the TPD database by our police detective. Our intention is to avoid any systematic errors that might be caused by the artificially generated incomplete datasets. From the TPD database, we were able to draw a dataset of 210 deceptive records in which missing values occurred in SSN only. Deceptive records missing values in other fields were not identified, either because certain missing types (e.g., address-missing, DOB-missing) were rare in the TPD database or because the police expert was not able to identify deceptive identities based on limited available values (e.g., SSN-Address-missing and SSN-DOB-missing).
- Testing procedure: For each missing type, we tested the proposed algorithm for several iterations, each of which had a different percentage (ranging from 10% to 90%) of missing values in the dataset for the intended field(s). During each iteration, we used a ten-fold validation method to test the algorithm's detection accuracy. As in the previous experiments, an optimal threshold value was determined when the highest F-measure was achieved during the training session. The detection accuracy measures of the algorithm were recorded during the testing session. T-tests were used to compare F-measures achieved by the algorithm using incomplete datasets to those acquired using a complete dataset.

c) Evaluating scalability: In terms of scalability, we compare the adaptive detection algorithm to the record comparison algorithm when detecting deception in large datasets (e.g., on the order of millions).

— Hypothesis 4 (H4): There is no significant difference in scalability between the adaptive detection algorithm and the record comparison algorithm.

- Testing datasets: We randomly selected 10 000 criminal identity records from the TPD database as the starting dataset for our scalability testing. We then increased the size of the selection by 10 000 at a time until all identity records in the TPD database (about 1.3 million) were included.

TABLE III
COMPARISON BETWEEN DETECTION EFFECTIVENESS OF THE ADAPTIVE DETECTION ALGORITHM AND THE RECORD COMPARISON ALGORITHM. (a) ALGORITHM EFFECTIVENESS IN TERMS OF F-MEASURE. (b) ALGORITHM EFFICIENCY IN TERMS OF NUMBER OF COMPARISONS AND COMPLETION TIME

(a)

Fold	F Measure	
	Adaptive Detection	Record Comparison
1	1.000	1.000
2	1.000	1.000
3	0.906	1.000
4	1.000	1.000
5	1.000	1.000
6	1.000	1.000
7	1.000	0.977
8	1.000	0.963
9	1.000	0.962
10	0.936	0.864
Avg	**0.984**	**0.977**

(b)

Fold	Number of Comparisons		Completion Time (msec)	
	Adaptive Detection	Record Comparison	Adaptive Detection	Record Comparison
1	116	210	5.950	9.432
2	123	210	5.731	8.311
3	105	210	5.033	8.404
4	144	210	7.337	8.615
5	97	210	4.647	8.509
6	132	210	7.516	8.480
7	132	210	5.868	8.249
8	151	210	7.229	8.725
9	105	210	5.003	17.280
10	134	210	6.775	8.843
Avg.	**123.9**	**210**	**6.109**	**9.485**

- Testing procedure: For each selected dataset, we detected deceptive identities using the adaptive detection algorithm and the record comparison algorithm, respectively. The scalability of each algorithm, as defined earlier, was computed for each test. A t-test was performed to compare the scalability difference between the two algorithms over different sizes of datasets.

VI. RESULTS AND DISCUSSIONS

A. Effectiveness of the Adaptive Detection Algorithm (H1 and H2)

Table III shows the detection accuracy, in terms of F-measure, achieved by the adaptive detection algorithm and the record comparison algorithm, respectively. A t-test showed that there was no significant difference between the two algorithms (p-value $= 0.659$).

Algorithm efficiency measures achieved by the two algorithms, in terms of number of comparisons and completion time, are also listed in Table III. H2 was also tested with a t-test and was rejected at a significant level (p-value $\ll 0.05$). The result showed that the adaptive detection algorithm is more efficient than the pairwise record comparison algorithm.

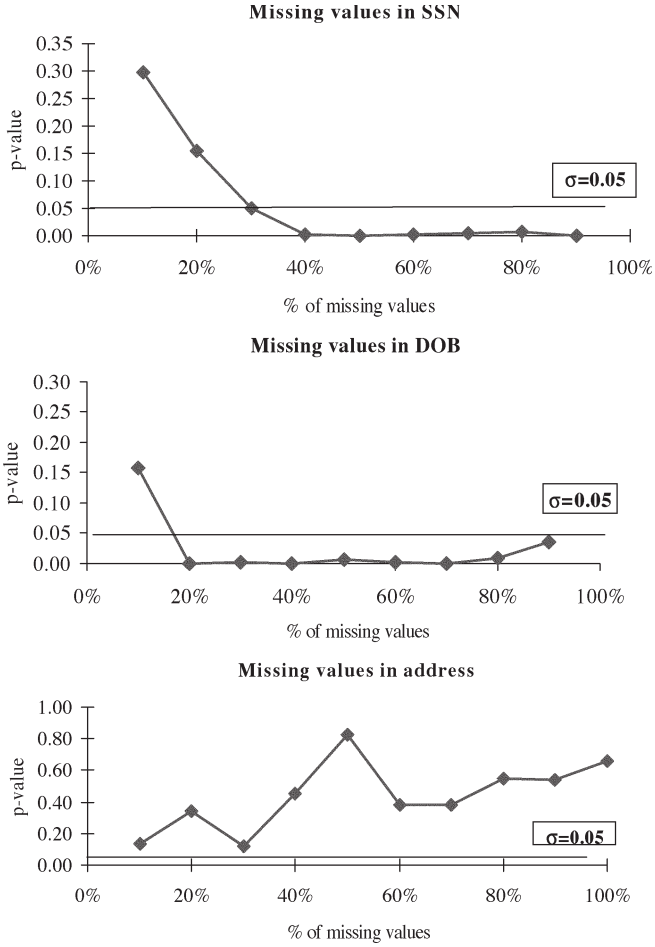


Fig. 4. Performance comparison between the complete dataset and the datasets missing values in one attribute (σ is the significance level of the t-test).

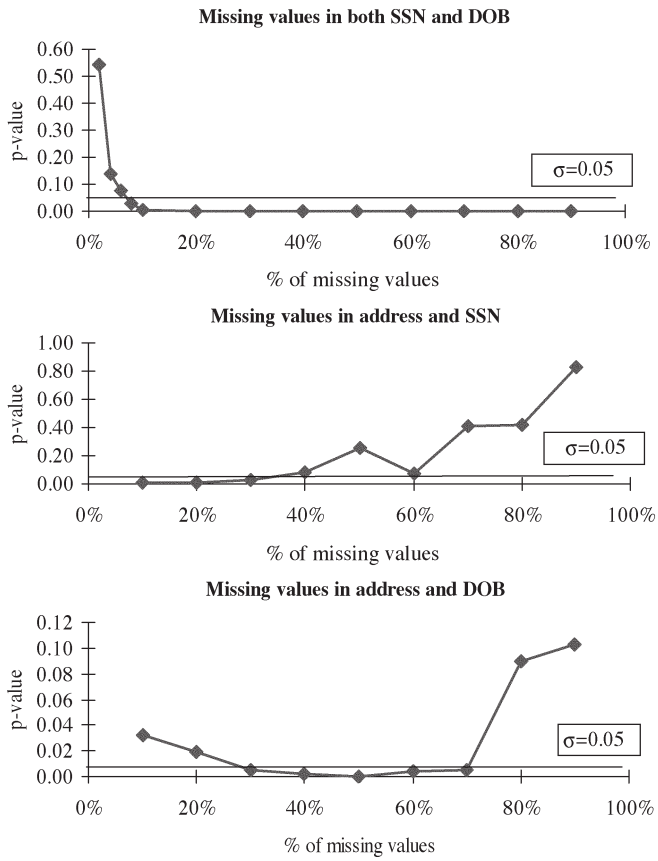


Fig. 5. Performance comparison between the complete dataset and the datasets missing values in two attributes (σ is the significance level of the t-test).

B. Adaptive Detection Algorithm in Handling Missing Values (H3)

1) Testing With Artificially Generated Missing Values: We used p-values of t-tests to indicate whether there was a significant difference in detection accuracy between using a complete dataset and using a dataset that contained a certain percentage of missing values in certain attributes. For each type of incomplete dataset (i.e., values missing in certain attributes), we plotted p-values against the percentage of incomplete identity records contained in a dataset to indicate the significant changes in the algorithm's effectiveness. The effect of the amount of missing values on detection accuracy is clearly visible.

P-values in Fig. 4 indicate the adaptive detection algorithm's performance differences between using a complete dataset and using a dataset in which identity records contain missing values for one attribute. When values were only missing for SSN, the detection accuracy (F-measure) of the adaptive detection algorithm did not significantly decrease if the percentage of incomplete records was less than 30%. Similarly, when values were only missing for DOB, the detection accuracy of the adaptive detection algorithm did not lower significantly if the percentage of incomplete records was less than 18%. However, there were significant variations in the detection accuracy when values were missing in the address attribute, regardless of the percentage of incomplete records.

P-values in Fig. 5 show the adaptive detection algorithm's performance differences between using a complete dataset and using a dataset in which identity records contain missing values for two attributes. When values were missing exclusively in SSN and DOB, the detection accuracy of the adaptive detection algorithm did not significantly decrease if the percentage of incomplete records was less than 12%. Similar to the one-attribute-missing case, detection accuracy varied when there were missing values in the address field.

To explain why the existence of missing values in the address field brought variations to the algorithm's detection accuracy, we examined the characteristics of address values in the complete dataset and compared them with the SSN and the DOB. For each attribute, the distribution of disagreement values between related identities (i.e., different identities referring to the same individual) is shown in Fig. 6. We noticed that the distribution for the address attribute is very different from that for DOB or SSN. DOB and SSN both have a skewed distribution such that identities pointing to the same person mostly have very similar DOB or SSN values. Address, however, has a bipolar distribution of disagreement values. In our dataset, identities of the same individual sometimes have similar address values and sometimes have very different address values. Such a difference between address and the other two attributes might explain the difference in the algorithm's detection accuracy.

2) Testing With Real Missing Values: This dataset extracted from the TPD database had missing values in the SSN field

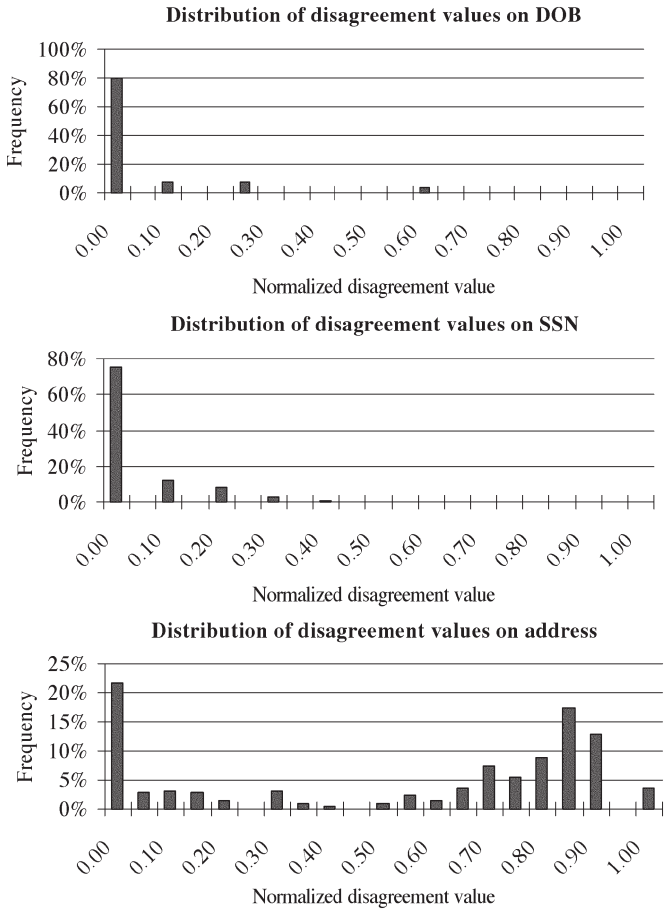


Fig. 6. Distribution of disagreement values on each attribute.

only. As shown in Table IV, the adaptive detection algorithm was able to achieve on average a high precision of 93.7% and a recall of 73.6%. Compared to the detection performance using complete records, the detection precision was decreased for records with values missing in SSN. However, there was a significant decrease in the detection recall, which led to a significant drop in the overall F-measure. Two possible reasons may cause low detection recalls, namely: either two identity records of the same individual are located too far apart (e.g., much larger than the size of the sliding window in the adaptive detection algorithm), or the threshold value is too strict in determining deceptive identities.

C. Efficiency and Scalability (H4)

Scalability measures of the two algorithms are shown in Fig. 7. The adaptive detection algorithm took 6.5 min for the adaptive detection algorithm to finish detecting deceptive identity in 1.3 million records. As the data volume increased, it maintained a gentle slope in the time it needed to finish detections. Note that the 6.5 min did not include the sorting time. Sorting was performed within the database. It would add very minor overhead to the overall running time if the database was appropriately indexed. However, the detection time of the record comparison algorithm increased dramatically. It would have spent 87 days on the same task. Both algorithms were implemented in Java. Experiments were conducted on an HP PC with a Pentium III 800-MHz CPU and 256-MB RAM.

TABLE IV
DETECTION PERFORMANCE WITH REAL MISSING VALUE. (a) DETECTION PERFORMANCE WITH RECORDS CONTAINING REAL MISSING VALUES. (b) DETECTION PERFORMANCE WITH COMPLETE RECORDS

(a)

Fold	Recall	Precision	F-Measure
1	0.786	1.000	0.880
2	0.500	1.000	0.667
3	1.000	1.000	1.000
4	0.417	1.000	0.588
5	0.750	1.000	0.857
6	0.846	0.846	0.846
7	0.846	0.786	0.815
8	0.615	1.000	0.762
9	0.688	1.000	0.815
10	0.917	0.733	0.815
Avg.	0.736	0.937	0.804

(b)

Fold	Recall	Precision	F- Measure
1	1.000	1.000	1.000
2	1.000	1.000	1.000
3	1.000	1.000	1.000
4	1.000	1.000	1.000
5	1.000	1.000	1.000
6	1.000	1.000	1.000
7	1.000	1.000	1.000
8	0.857	1.000	0.923
9	1.000	1.000	1.000
10	0.880	1.000	0.936
Avg.	**0.974**	**1.000**	**0.986**

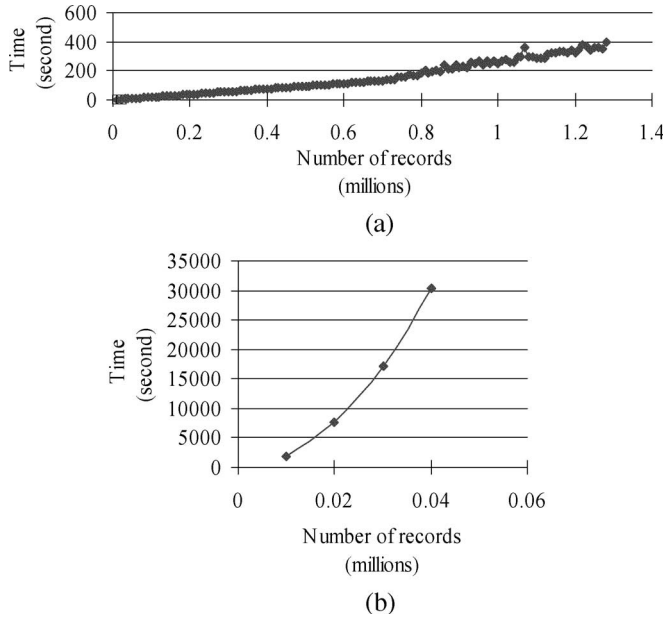


Fig. 7. Efficiency and scalability performance. (a) Scalability of the adaptive detection algorithm. (b) Scalability of the record comparison algorithm.

D. Case Study

To further evaluate the implication of our proposed algorithm, we tested it with another real dataset provided by the Pima County Sheriff Department (PCSD). PCSD serves

330 000 people living in the seventh largest county in the nation. We consider it as a representative of law enforcement agencies in the U.S. The PCSD dataset contained over 1.3 million identity records. Residential address and SSN information was not available in the dataset and was considered missing. We ignored those records that only had names because it is not reliable to determine deception solely by names. There were 700 686 identity records remaining in the testing dataset, each of which has values in the attributes of first name, last name, and DOB. With a window size of 10, our algorithm was able to identify 16 912 clusters. Identities of each cluster were considered to refer to the same person. We randomly chose 20 clusters and asked our police detective expert to evaluate each of them. The expert from the TPD confirmed that 11 out of 20 clusters were correctly grouped. There were six clusters that the expert from the TPD could not verify because of limited information. Three clusters were incorrectly clustered due to the use of common names and similar DOBs.

The expert from the TPD found this algorithm useful in finding both deceptive identity records and records that have data errors such as misspellings. Currently, the record management system used by this agency is not able to automatically group the identity records that refer to the same person. The six clusters that the expert from the TPD was unable to verify could also be useful in providing additional leads during investigation processes.

VII. Conclusion and Future Work

In this paper, we discussed algorithmic approaches to automatically detecting criminal identity deception. We proposed an adaptive detection algorithm that improved the record comparison algorithm in terms of efficiency, scalability, and ability to handle incomplete identities. Experiments showed that the proposed algorithm greatly improved detection efficiency and achieved detection accuracy comparable with that of the pairwise record comparison algorithm. Our experiments also showed that the detection accuracy of the adaptive detection algorithm was not affected when there was a small percentage of attribute values missing (less than 30% for missing values on SSN or less than 18% for missing values on DOB). In cases where there was a larger percentage of attribute values missing, the adaptive detection algorithm could still maintain detection precision of around 95%.

However, limitations exist in this paper. The testing dataset is relatively small. The changing data characteristics of the testing dataset may affect the algorithm's performance. The algorithm's parameters (e.g., window size of the priority queue and/or threshold values) may be adjusted when running the algorithm in a different dataset.

Our proposed algorithm assumes that all attributes are equally important. Therefore, it assigns an equal weight to each attribute when combining disagreement measures of the four attributes into an overall measure between two identity records. We may consider a different weighting schema. For example, in the future, we may assign less weight to the address attribute because disagreement measures among related addresses introduce noise rather than contribute to the detection of deceptive identities. The assumption would also lead to the conclusion that two records, in which only the first name "John" was recorded, would have the same probability of describing the same person as two records, in which all of the fields exist. Intuitively, if name is the only available field to compare, one can only judge the probability that two identities describe the same person solely by the names. However, the confidence in the match increases as more fields are available to compare.

One of the intentions of our proposed algorithm is to avoid pairwise comparisons, so that detection efficiency can be improved. However, detection effectiveness may be affected, whereas the efficiency is improved under the assumption that two identities of the same individual sorted by an appropriate key are located close to each other. That assumption is, however, not guaranteed. It is possible that the two identities are located too far apart to be grouped into the same cluster. Although the algorithm did not cause a significant drop of detection efficacy in our experiments, we will consider more advanced clustering algorithms such as mixture models to avoid the assumption in future work.

In addition to detecting intentional deception, both record comparison algorithm and the proposed adaptive detection algorithm are capable of dealing with identity records having unintentional data errors such as misspellings. It might be interesting to differentiate between the patterns of deception and errors. However, we do not perceive any difference in terms of the algorithm's effectiveness.

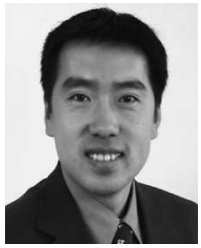
In the future, we intend to consider other identity-related information, such as biometrics, behavior characteristics, and social context. A good example of behavior characteristics is MO, which is often used to identify a criminal in crime investigation. The social context is a set of characteristics of the social system that a person usually lives. These types of information can also be helpful in determining a person's identity. The core function of our proposed algorithm is to combine the disagreement measure of each of the four attributes and to determine the disagreement (or similarity) between two identity records. It is open to include more identity attributes when a disagreement measure can be defined for each attribute. A more comprehensive model that encompasses more identity attributes is desirable in future research.

The proposed automated deception detection system will be incorporated into our ongoing COPLINK project [19], which has been under development at the University of Arizona's Artificial Intelligence Lab, in collaboration with the TPD, and PCSD, since 1997. Such a system can also be used in merging customer profiles for marketing purposes.

References

[1] *Identity Fraud: A Study*. (2002). London, U.K.: Home office. [Online]. Available: http://www.homeoffice.gov.uk/cpd/id_fraud-report.pdf
[2] P. D. Allison, *Missing Data*. Thousand Oaks, CA: Sage, 2001.
[3] A. S. J. Aubry, *Criminal Interrogation*, 3rd ed. Springfield, IL: Charles C. Thomas, 1980.
[4] A. B. Badiru, J. M. Karasz, and B. T. Holloway, "AREST: Armed robbery eidetic suspect typing expert system," *J. Police Sci. Admin.*, vol. 16, no. 3, pp. 210–216, Sep. 1988.
[5] D. E. Brown and S. Hagen, "Data association methods with applications to law enforcement," *Decis. Support Syst.*, vol. 34, no. 4, pp. 369–378, 2003.
[6] S. F. Buck, "A method of estimating missing values in multivariate data suitable for use with an electronic computer," *J. R. Statist. Soc.*, vol. B22, no. 2, pp. 302–306, 1960.

[7] J. K. Burgoon, D. B. Buller, L. K. Guerrero, W. Afifi, and C. Feldman, "Interpersonal deception: XII. Information management dimensions underlying deceptive and truthful messages," *Commun. Monogr.*, vol. 63, no. 1, pp. 50–69, Mar. 1996.
[8] K. C. Clarke, *Getting Started With Geographic Information Systems*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.
[9] R. Clarke, "Human identification in information systems: Management challenges and public policy issues," *Inf. Technol. People*, vol. 7, no. 4, pp. 6–37, Dec. 1994.
[10] J. Cohen, "Errors of recall and credibility: Can omissions and discrepancies in successive statements reasonably be said to undermine credibility of testimony?" *Med.-Leg. J.*, vol. 69, no. 1, pp. 25–34, 2001.
[11] B. M. DePaulo and R. L. Pfeifei, "On-the-job experience and skill at detecting deception," *J. Appl. Soc. Psychol.*, vol. 16, no. 3, pp. 249–267, 1986.
[12] J. S. Donath, "Identity and deception in the virtual community," in *Communities in Cyberspace*, P. Kollock and M. Smith, Eds. London, U.K.: Routledge, 1998.
[13] P. Ekman, M. O'Sullivan, "Who can catch a liar?" *Amer. Psychol.*, vol. 46, no. 9, pp. 913–920, Sep. 1991.
[14] P. Ekman, *Telling Lies: Clues to Deceit in the Marketplace, Politics and Marriage*, 3rd ed. New York: Norton, 1992.
[15] GAO, "Law enforcement: Information on timeliness of criminal fingerprint submissions to the FBI," U.S. Gen. Accounting Off. (GAO), Washington, DC, GAO-04-260, 2004.
[16] M. Glasser, "Linear regression analysis with missing observations among the independent variables," *J. Amer. Statist. Assoc.*, vol. 59, no. 307, pp. 834–844, Sep. 1964.
[17] S. O. Gyimah, "Missing data in quantitative social research," Dept. Sociology, Univ. Western Ontario, London, ON, Canada, Rep. 01-14, 2001.
[18] Y. Haitovsky, "Missing data in regression analysis," *J. R. Statist. Soc.*, vol. B30, no. 1, pp. 67–82, 1968.
[19] R. V. Hauck, H. Atabakhsh, P. Ongvasith, H. Gupta, and H. Chen, "Using COPLINK to analyze criminal-justice data," *Computer*, vol. 35, no. 3, pp. 30–37, Mar. 2002.
[20] M. A. Hernandez and S. J. Stolfo, "The merge/purge problem for large databases," in *Proc. ACM SIGMOD Int. Conf. Management Data*, San Jose, CA, 1995, pp. 127–138.
[21] ——, "Real-world data is dirty: Data cleansing and the merge/purge problems," *Data Mining Knowl. Discov.*, vol. 2, no. 1, pp. 9–37, 1998.
[22] G. Jones. (2001). *E-Commerce and Identity Fraud*, Nottingham, U.K.: Experian Co. [Online]. Available: http://press.experian.com/documents/e-comm.pdf
[23] J. Kim and J. Curry, "The treatment of missing data in multivariate analysis," *Sociol. Methods Res.*, vol. 6, no. 2, pp. 206–240, 1977.
[24] G. Kohnken, "Training police officers to detect deceptive eyewitness statements: Does it work?" *Soc. Behav.*, vol. 2, no. 1, pp. 1–17, 1987.
[25] R. E. Kraut and D. Poe, "On the line: The deception judgements of customs inspectors and laymen," *J. Pers. Soc. Psychol.*, vol. 39, no. 5, pp. 784–798, 1980.
[26] V. L. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Phys. Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.
[27] W. L. Low, M. L. Lee, and T. W. Ling, "A knowledge-based approach for duplicate elimination in data learning," *Inf. Syst.*, vol. 26, no. 8, pp. 585–606, Dec. 2001.
[28] A. E. Monge, "Adaptive detection of approximately duplicate database records and the database integration approach to information discovery," Ph.D. dissertation, Dept. Comput. Sci. Eng., Univ. California, San Diego, 1997.
[29] A. E. Monge and C. P. Elkan, "An efficient domain-independent algorithm for detecting approximately duplicate database records," in *Proc. ACM-SIGMOD Workshop Research Issues Knowledge Discovery Data Mining*, Tucson, AZ, 1997, pp. 23–29.
[30] L. Myrtveit, E. Stensrud, and U. H. Olsson, "Analyzing data sets with missing data: An empirical evaluation of imputation methods and likelihood-based methods," *IEEE Trans. Softw. Eng.*, vol. 27, no. 11, pp. 999–1013, Nov. 2001.
[31] J. R. Quinlan, "Induction of decision tree," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986.
[32] ——, "Unknown attribute values in induction," in *Proc. 6th Int. Machine Learning Workshop*, 1989, pp. 164–168.
[33] D. B. Rubin, *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley, 1987.
[34] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA: Addison-Wesley, 1988.
[35] J. L. Schafer, *Analysis of Incomplete Multivariate Data*. London, U.K.: Chapman & Hall, 1997.
[36] H. Timm and F. Klawonn, "Different approaches for fuzzy cluster analysis with missing values," presented at the 7th Eur. Congr. Intelligent Techniques and Soft Computing, Aachen, Germany, 1999.
[37] S. Toth. (2003). *Need Fuels Demand for False IDs: For Jobs, Documents are the Key*, South Bend, IN: South Bend Tribune. [Online]. Available: http://www.southbendtribune.com/stories/2003/07/27/local.20030727-sbt-FULL-A1-Need_fuels_demand_fo.sto
[38] A. Vrij, *Detecting Lies and Deceit: The Psychology of Lying and the Implication for Professional Practice*. Hoboken, NJ: Wiley, 2000.
[39] G. Wang, H. Chen, and H. Atabakhsh, "Automatically detecting deceptive criminal identities," *Commun. ACM*, vol. 47, no. 3, pp. 71–76, Mar. 2004.
[40] A. P. White, W. Z. Liu, M. T. Hallissey, and J. W. L. Fielding, "A comparison of two classification techniques in screening for gastro-esophageal cancer," in *Proc. Appl. Innovations Expert Syst. IV*, 1996, pp. 83–97.



G. Alan Wang (S'05) received the B.S. degree in industrial management engineering from Tianjin University, Tianjin, China, in 1995, the M.S. degree in industrial engineering from Louisiana State University, Baton Rouge, in 2001. He is currently working toward the Ph.D. degree in management information systems at the University of Arizona, Tucson.

He has published papers in the *Communications of the ACM*, the *Journal of the American Society for Information Science and Technology*, IEEE COMPUTER and *Group Decision and Negotiation*. His research interests include data heterogeneity and uncertainty, data mining, and knowledge management.

Hsinchun Chen (M'92–SM'04–F'06) received the Ph.D. degree in information systems from New York University, New York, in 1989.

He is currently a McClelland Endowed Professor at the Department of Management Information Systems, University of Arizona, Tucson. He has authored or coauthored over 70 papers concerning semantic retrieval, search algorithms, knowledge discovery, and collaborative computing. He is an expert in digital library and knowledge management research, and his research has been featured in scientific and information technology publications.

Jennifer J. Xu received the M.S. degree in computer science and the M.A. degree in economics from the University of Mississippi, University, in 1999 and 2000, respectively. She is currently working toward the Ph.D. degree in management information systems.

She is currently an Assistant Professor of computer information systems at Bentley College, Waltham, MA. Her research interests include knowledge management, social network analysis, information retrieval, human–computer interaction, and information visualization.

Homa Atabakhsh received the M.S. and Ph.D. degrees from the University of Toulouse, Toulouse, France, in 1984 and 1987, respectively, all in computer science.

She was an Assistant Professor with the University of Toulouse from January 1988 to January 1989. From January 1989 to 1996, she was a Research Scientist at the National Research Council of Canada, Ottawa, ON, Canada, where she worked in areas such as knowledge-based systems, object-oriented design and programming, graphical user interface, and applications in manufacturing and business. She has also been an Adjunct Lecturer at the University of Ottawa. Currently, she is the Associate Director of the COPLINK Center for Excellence and a Principal Research Specialist at the Department of Management Information System, University of Arizona, Tucson.