Hindawi
Security and Communication Networks
Volume 2021, Article ID 4856265, 7 pages
https://doi.org/10.1155/2021/4856265

*Research Article*

# Matching Cyber Security Ontologies through Genetic Algorithm-Based Ontology Alignment Technique

**Weiwei Lin** [1,2] **and Reiko Haga** [3]

[1]*School of Big Data and Artificial Intelligence, Fujian Polytechnic Normal University, Fuqing 350300, China*
[2]*Engineering Research Center for ICH Digitalization and Multi-source Information Fusion, Fujian Province University, Fuqing 350300, China*
[3]*CommScope Japan KK, Nagatacho, Tokyo 100-0014, Japan*

Correspondence should be addressed to Weiwei Lin; linww_cn@hotmail.com

Received 23 September 2021; Revised 23 October 2021; Accepted 27 October 2021; Published 30 November 2021

Academic Editor: Pei-Wei Tsai



Security ontology can be used to build a shared knowledge model for an application domain to overcome the data heterogeneity issue, but it suffers from its own heterogeneity issue. Finding identical entities in two ontologies, i.e., ontology alignment, is a solution. It is important to select an effective similarity measure (SM) to distinguish heterogeneous entities. However, due to the complex semantic relationships among concepts, no SM is ensured to be effective in all alignment tasks. The aggregation of SMs so that their advantages and disadvantages complement each other directly affects the quality of alignments. In this work, we formally define this problem, discuss its challenges, and present a problem-specific genetic algorithm (GA) to effectively address it. We experimentally test our approach on bibliographic tracks provided by OAEI and five pairs of security ontologies. The results show that GA can effectively address different heterogeneous ontology-alignment tasks and determine high-quality security ontology alignments.

## 1. Introduction

Security ontology builds a shared knowledge model for an information system's security area to facilitate the establishment of trust relationships [1]. Figure 1 shows an example of security ontology. An oval denotes a concept, such as SecurityProtocol or ProtocolEncryption. The arrow between two concepts denotes a subsumptive relationship, for example, ProtocolSignature is subsumed by SecurityProtocol. A concept might have properties, such as the XACML and ACL properties of ProtocolAccessControl. However, security ontologies have different application requirements and bias interest, which causes the ontologies themselves to suffer from the heterogeneity problem. Finding identical entities in two security ontologies, i.e., security ontology alignment, is a solution to this issue [2, 3]. It is important to use a similarity measure (SM) to distinguish heterogeneous entities when aligning security ontologies. However, due to the complex semantic relationships among concepts, no SM is effective in all contexts. Hence, it is important to aggregate SMs so that their advantages and disadvantages complement each other.

The most flexible way to aggregate SMs is the parallel framework, which assigns a weight for each SM to obtain the final alignment. During this procedure, each SM's similarity matrix is calculated, whose rows and columns, respectively, represent two ontologies' entities and whose elements are their similarity values. The aggregated matrix is determined by aggregating all the matrices with the weighted mean strategy. A threshold is used to filter elements with low similarity values to obtain the final matrix, which is decoded to the ontology alignment. It is a complex problem to determine the optimal aggregating weight set for SMs since there are many local optimal solutions. Genetic algorithm (GA) [4, 5] is a classic global optimization algorithm, which is adept at solving the optimization problem without the

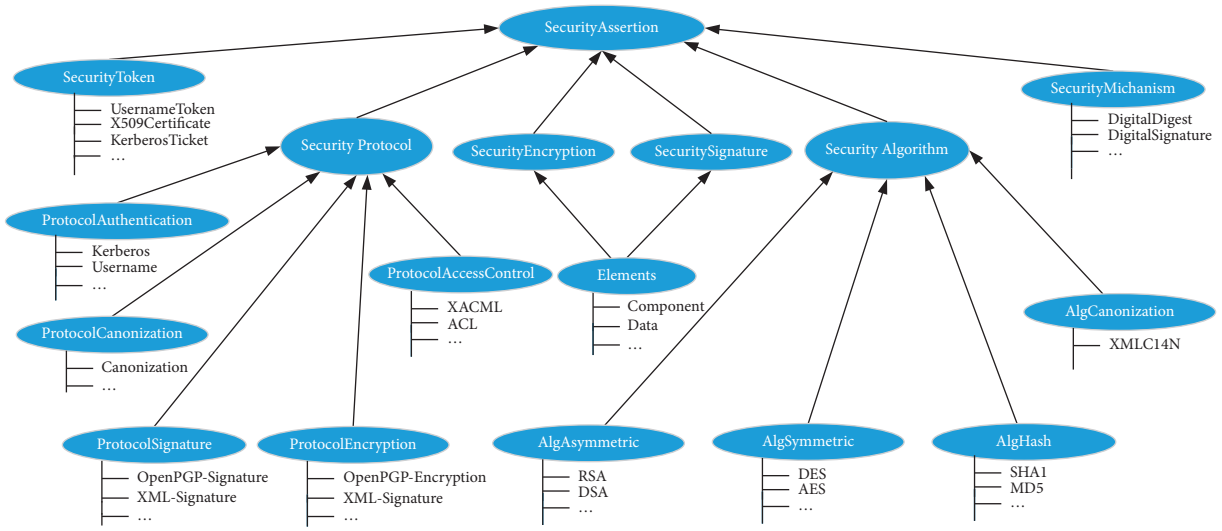


Figure 1: An example of security ontology.

information of the objective's gradient. Being inspired by its success in the complex optimization domains [6, 7], we build a mathematical model under a parallel aggregating framework to define the security ontology alignment problem, propose a problem-specific GA to address it, and determine high-quality security ontology alignments.

The remainder of this paper is arranged as follows. Section "Preliminaries" defines the security ontology alignment and similarity measure. Section "Genetic Algorithm to Integrate Security Ontologies" describes the GA-based alignment technique. Experimental results are discussed in section "Experiment," and section "Conclusion" relates our conclusions.

## 2. Preliminaries

*2.1. Security Ontology Alignment.* Security ontology consists of concepts, properties, and axioms, and an ontology alignment is a mapping set. A mapping is a 3-tuple ($c_1$, $c_2$, sim), where $c_1$ and $c_2$ are two ontologies' entities, and sim is their similarity [8, 9]. Aligning ontologies require us to find the correspondence between two ontology entities to bridge their semantic gap. As shown in Figure 2, the input of ontology alignment is a pair of ontologies. After using different SMs to determine the corresponding similarity matrices, GA is used to optimize their aggregating weights to obtain the final alignment.

A security ontology alignment's quality can be measured with metrics in the information retrieval domain [10]:

$$\begin{aligned} \text{recall} &= \frac{|R \cap RA|}{|R|}, \\ \text{precision} &= \frac{|R \cap RA|}{|A|}, \\ f - \text{measure} &= \frac{2\,\text{precision} \cdot \text{recall}}{\text{recall} + \text{precision}}, \end{aligned} \tag{1}$$

where $A$ and $RA$ are, respectively, an alignment and reference alignment and  denotes a set's cardinalities. Here, $f$-measure is the harmony mean of recall and precision. On this basis, the security ontology alignment problem has the objective to maximize the $f$-measure, and the decision variable is $X = (x_1, x_2, \ldots)^T$, where $x_i \in [0, 1]$, $i = 1, 2, \ldots$, is the $i$th SM's aggregating weight, and $\sum x_i = 1$. In this work, we choose the weighted average strategy to aggregate the SMs, which is the most popular and flexible method in the domain of information fusion of combining SMs. The other aggregating mechanisms, such as those in the field of evidential reasoning and fuzzy reasoning, could be also applied, which is one of our future works.

*2.2. Similarity Measure.* SM can generally be categorized as either syntactic, linguistic, or taxonomy SM [11, 12], which we describe as follows.

Syntactic SM calculates the similarity of two strings through their edit distance. We use the Levenshtein distance [13]:

$$\text{Levenshtein}(s_1, s_2) = \frac{\max 0,\ \min\left(|s_1|, |s_2|\right) - d(s_1, s_2)}{\min\left(|s_1|, |s_2|\right)}, \tag{2}$$

where $|s_1|$ and $|s_2|$ are the respective character numbers of strings $s_1$ and $s_2$ and $d(s_1, s_2)$ is their edit distance.

Linguistic SM utilizes an electronic dictionary to measure the similarity of two words. We use WordNet [14, 15] as the electronic knowledge base. Linguistic similarity is defined as

$$\text{Linguistic}(w_1, w_2) = \max_{c_1 \in \text{sen}(w_1), c_2 \in \text{sen}(w_2)} [\text{sim}(c_1, c_2)], \tag{3}$$

where $w_1$ and $w_2$ are words derived from two entities and $\text{sen}(w_i)$ denotes the number of meanings of $w_i$.

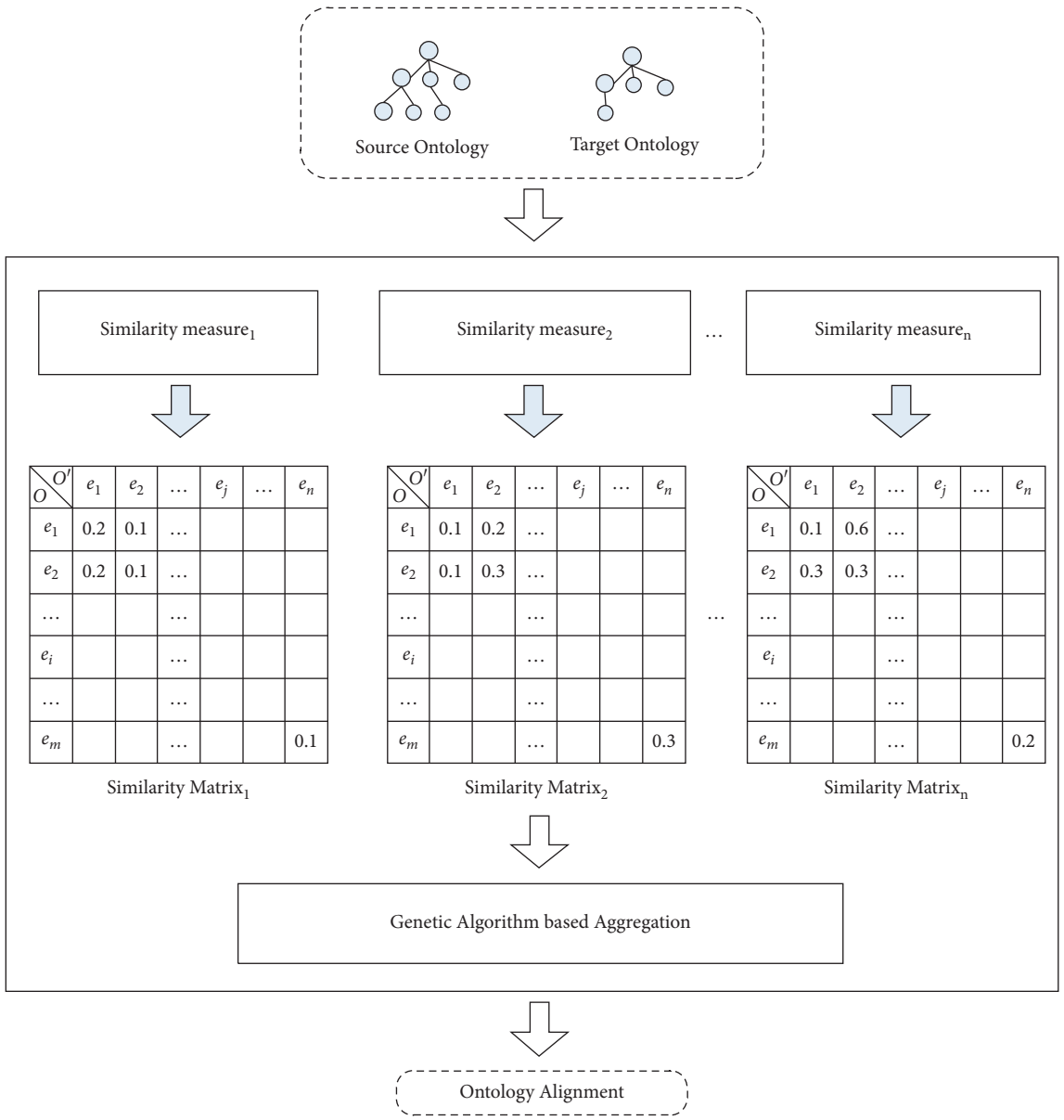


Figure 2: Framework of ontology alignment.

Taxonomy SM uses the context of concepts $c_1$ and $c_2$ to determine their similarity [16, 17]:

$$\text{Taxonomy}(c_1, c_2) = \frac{\text{Levenshtein}(\text{super}_1, \text{super}_2) + \text{avg}\{\text{Levenshtein}(\text{sub}_i, \text{sub}_j)\}}{2}, \tag{4}$$

where $\text{super}_1$ and $\text{super}_2$ are the superclasses of $c_1$ and $c_2$, respectively, and $\text{sub}_i$ and $\text{sub}_j$ are, respectively, their $i$ th and $j$ th subclasses. In particular, the taxonomy SM determines the similarity value by calculating the average similarity of two concepts' parent pair and all their direct subclass pairs.

## 3. Genetic Algorithm to Integrate Security Ontologies

*3.1. Encoding Mechanism.* In this work, we use binary coding [18] to reduce the evolutionary operation's computational complexity. Considering that the coding

information must contain the weight set of SMs, we store them in disguised form by storing the cutting points in the coding information. We sort a set of cutting points $C\prime = (c_1', c_2', \ldots, c_n')$ in the ascending order as $C = (c_1, c_2, \ldots, c_n)$, and then we can get the corresponding weight set:

$$w_k = \begin{cases} c_1, & k = 1, \\ c_{k-1} - c_k, & 1 < k < n+1, \\ 1 - c_n, & k = n+1. \end{cases} \quad (5)$$

Through calculation, we can use $n$ cutting points to obtain $n+1$ aggregating weights. This work selects three SMs, so we need to encode the information of two cutting points. We use 10 gene bits to represent a cutting point; hence, the length of a chromosome is 20 gene bits. Figure 3 shows an example of the encoding mechanism, where two cutting points represent the aggregating weights of the SM, and five gene bits are used to encode each cutting point. As shown in the figure, a chromosome is decoded to decimal to obtain the cutting point set $C\prime$, which is sorted to obtain the cutting point set $C$. Then, weights $w_1$, $w_2$, and $w_3$ are calculated according to formula (5).

*3.2. Selection.* The selection operator is the kernel component of GA, which decides whether a solution's gene information can persist. A solution with a higher fitness value should have a greater probability of selection, but one with a lower fitness value should also have a certain opportunity. This work empirically chooses the classic roulette selection operator. The probability of selecting an individual is the ratio of its fitness value to the sum of the fitness values of all solutions; hence, each individual has the opportunity to be selected. If the $i$ th solution has fitness value $f_i$, its selection probability is $f_i/\sum f_i$.

*3.3. Crossover.* The crossover operator mixes the genes of two parent solutions according to a crossover probability. We randomly select a cutting point using the single-point crossover operator [19], and two children are generated by swapping the right parts of two parents' genes.

*3.4. Mutation.* The mutation operator aims to maintain population diversity, which is critical to the algorithm's searching ability. This work selects the locus mutation operator [20], which judges whether a gene value should be flipped by generating a random number in [0, 1] and comparing it with the mutation probability.

*3.5. Pseudocode of Genetic Algorithm.* Given the maximum generation *maxGeneration*, we present the GA pseudocode:

```
******* Initialization**********
for i = 0; i < population.length; i + +do
  for j = 0; j < population.length; j + +do
    gene_{i,j} = random{0, 1};
  end for
end for
******* Evaluation**********
for i = 0; i < population.length; i + +do
  evaluation();
end for
******* Evolution**********
generation = 0;
while generation < max Generation do
  crossover();
  mutation();
  for i = 0; i < population.length; i + +do
    evaluation();
  end for
  selection();
  saveElite();
  generation = generation + 1;
end while
```

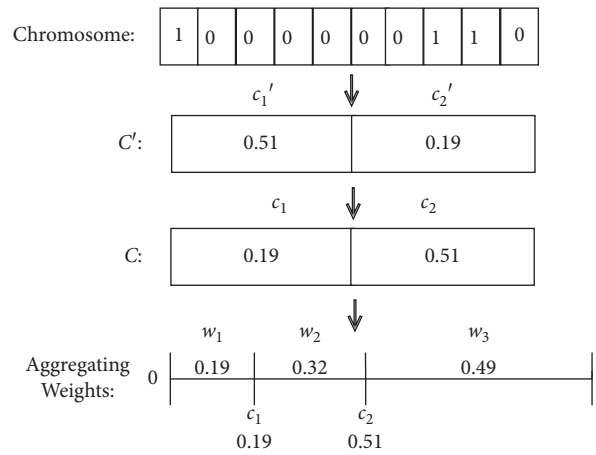


Figure 3: Example of encoding mechanism.

The gene values of each individual are initialized as 1 or 0, and then the population's solutions are evaluated. In each generation, the crossover and mutation operators are successively applied, and all solutions are re-evaluated. The selection operator is then used to determine the population of the next generation. Finally, the worst solution is replaced by the best one in the history (i.e., the elite solution).

*3.6. Experiment.* We utilized the Bibliographic track from OAEI (http://oaei.ontologymatching.org) to test the performance of our proposal. In particular, 1XX and 2XX are the respective testing cases with IDs beginning with 1 and 2. In 1XX, two ontologies under alignment are exactly the same except for different OWL restrictions, while in 2XX, they are heterogeneous in terms of the entity name and/or the concept's hierarchical structure. We also chose four pairs of specialized security ontologies for testing: (1) Network Security Ontologies—Network Attack Ontology (NAO) [21] and Ontology-Based Attack Model (NAM)

Table 1: Comparison on OAEI's bibliographic track in terms of recall and precision.

| Matching system | 1XX | | 2XX | |
|---|---|---|---|---|
| | Precision | Recall | Precision | Recall |
| Edna | 0.64 | 1.00 | 0.62 | 0.84 |
| LogMap | 0.94 | 0.96 | 0.90 | 0.81 |
| LogMapLt | 0.56 | 0.99 | 0.53 | 0.83 |
| LogMapBio | 0.50 | 0.56 | 0.52 | 0.65 |
| GMap | 0.97 | 1.00 | 0.88 | 0.85 |
| LogMap-C | 0.58 | 0.96 | 0.57 | 0.81 |
| Mamba | 0.90 | 0.84 | 0.79 | 0.76 |
| AOT 2014 | 0.97 | 0.97 | 0.93 | 0.83 |
| OReasoner | 0.87 | 1.00 | 0.74 | 0.84 |
| CIDER-CL | 1.00 | 1.00 | 0.78 | 0.91 |
| HerTUDA | 0.89 | 1.00 | 0.90 | 0.85 |
| MapSSS | 0.89 | 0.34 | 0.87 | 0.27 |
| RIMOM 2013 | 0.84 | 1.00 | 0.63 | 0.88 |
| ServOMap | 0.95 | 1.00 | 0.67 | 0.56 |
| StringsAuto | 0.89 | 0.34 | 0.87 | 0.27 |
| Synthesis | 0.94 | 1.00 | 0.81 | 0.86 |
| XMapGen | 0.84 | 1.00 | 0.67 | 0.78 |
| XMapSig | 0.84 | 1.00 | 0.70 | 0.84 |
| ASE | 0.58 | 1.00 | 0.61 | 0.85 |
| GOMMA | 0.84 | 1.00 | 0.70 | 0.87 |
| MEDLEY | 0.72 | 1.00 | 0.68 | 0.84 |
| Optima | 1.00 | 1.00 | 0.85 | 0.83 |
| ServOMap | 1.00 | 0.98 | 0.91 | 0.76 |
| ServOMaplt | 1.00 | 0.28 | 1.00 | 0.45 |
| WMatch | 0.84 | 1.00 | 0.73 | 0.85 |
| GA | 1.00 | 1.00 | 0.95 | 0.85 |

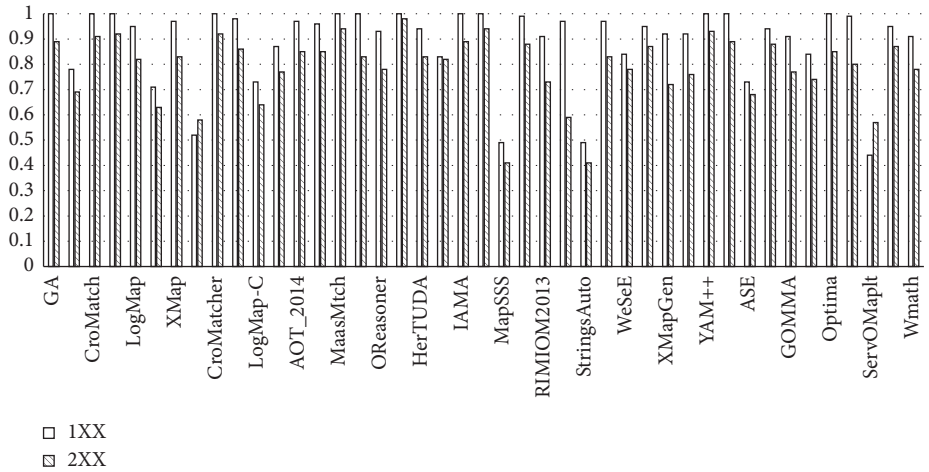


Figure 4: Comparison of OAEI's Bibliographic track in terms of *f*-measure.

[22]; (2) Security Requirement-Related Ontologies—Security and Domain Ontology for Security Requirement Analysis (SDOSRA) [23] and Extended Ontology for Security Requirements (EOSR) [24]; (3) Miscellaneous Security Ontologies—Ontological approach toward Cyber Security in Cloud Computing (OCSCC) [25] and Ontology in Cloud Computing (OCC) [26]; (4) Application-Based Security Ontologies—Security Ontology for Mobile Applications (SOMA) [27] and Security Ontology for Mobile Agents Protection (SOMAP) [28], and Cloud Security Policy (CSP) [29] and Cloud Ontology (CO) [30]. The threshold for filtering the final alignment was set as 0.85, and the configuration of GA was empirically set to a maximum 3000 generations, crossover rate 0.6, and mutation rate 0.02. In the experiment, we compared our approach with OAEI's participants, Table 1 compares the results in terms of recall and precision, and Figure 4 compares the *f*-measures. Table 2 shows the results of using GA to align the security ontologies. The results of our approach were the mean values of 30 independent runs.

Table 2: Experimental results on security ontology alignment.

| Category | Testing case | Recall | Precision | $f$-measure |
|---|---|---|---|---|
| Network security ontologies | NAO-NAM | 0.82 | 0.75 | 0.78 |
| Security requirement-related ontologies | SDOSRA-EOSR | 0.76 | 0.90 | 0.82 |
| Miscellaneous security ontologies | OCSCC-OCC | 0.88 | 0.95 | 0.91 |
| Application-based security ontologies | SOMA-SOMAP | 0.84 | 0.88 | 0.85 |
| | CSP-CO | 0.86 | 0.82 | 0.83 |

As shown in Table 1, the recall and precision of our approach were generally higher than those of OAEI. This is because GA is able to effectively jump out of lots of local optimas, and find the optimal aggregating weights from large-scale feasible solutions. In particular, the precision of our approach was high, which shows that aggregating different similarity measures can effectively distinguish heterogeneous entities.

As can be seen from Figure 4, the results of our approach were the best on 1XX testing cases, which shows that GA can effectively align two ontologies with the same entities and structures. In addition, with respect to different heterogeneous tasks on 2XX testing cases, our approach was also effective, which shows that our approach is able to address the matching problem with different heterogeneity characteristics.

Table 2 depicts the results of approaches to aligning five pairs of real security ontologies, which show our approach can achieve a high capacity on all testing cases in terms of the $f$-measure. To sum up, our approach was robust at addressing different alignment tasks and could determine high-quality security ontology alignments.

## 4. Conclusions

To ensure communication and cooperation among different security applications built on security ontologies, we proposed a GA-based ontology alignment technique to address the security ontology heterogeneity problem. We defined the problem, discussed its challenges, and presented a problem-specific GA to effectively address it. Bibliographic tracks provided by OAEI and five pairs of security ontologies were used to test our approach's performance. The experimental results show that our approach is able to align different heterogeneous ontologies and determine high-quality security ontology alignments.

In the future, we are interested in adaptive similarity selection, which determines effective and nonconflicting similarity measures according to the heterogeneous features of two ontologies under alignment. Moreover, when the number of similarity measures is large, some strategies to improve efficiency should be introduced to improve GA's performance.

## Data Availability

The data used to support this study can be found in the corresponding footnotes.

## Conflicts of Interest

The authors declare that they have no conflicts of interest in the work.

## Acknowledgments

The authors thank LetPub (https://www.letpub.com) for its linguistic assistance during the preparation of this manuscript. This work was supported by the Natural Science Foundation of Fujian Province, China (grant no. 2019J01889); the "Tiancheng Huizhi" Innovation and Education Promotion Fund, China (grant no. 2018A02005); and the Education-Scientific Research Project for Middle-Aged and Young of Fujian Province, China (grant no. JT180626).

## References

[1] S. Hacini and R. Lekhchine, "Security ontology for mobile agents protection," *International Journal of Computer Theory and Engineering*, vol. 4, no. 3, pp. 426–428, 2012.

[2] P. Shvaiko and J. Euzenat, "Ontology matching: state of the art and future challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 1, pp. 158–176, 2013.

[3] I. Osman, S. Ben Yahia, and G. Diallo, "Ontology integration: approaches and challenging issues," *Information Fusion*, vol. 71, pp. 38–63, 2021.

[4] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[5] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.

[6] X. Xue and Y. Wang, "Optimizing ontology alignments through a memetic algorithm using both MatchFmeasure and unanimous improvement ratio," *Artificial Intelligence*, vol. 223, pp. 65–81, 2015.

[7] C. Kim, R. Batra, L. Chen, H. Tran, and R. Ramprasad, "Polymer design using genetic algorithm and machine learning," *Computational Materials Science*, vol. 186, pp. 1–6, 2021.

[8] G. Acampora, V. Loia, and A. Vitiello, "Enhancing ontology alignment through a memetic aggregation of similarity measures," *Information Sciences*, vol. 250, pp. 1–20, 2013.

[9] X. Xue and Y. Wang, "Using memetic algorithm for instance coreference resolution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 2, pp. 580–591, 2016.

[10] C. J. Van Rijsberge, *Information Retrieval*, University of Glasgow, London, UK, 1975.

[11] S. Mani and S. Annadurai, "Explicit link discovery scheme optimized with ontology mapping using improved machine learning approach," *Studies in Informatics and Control*, vol. 30, no. 1, pp. 67–75, 2021.

[12] X. Xue and J. Chen, "Matching biomedical ontologies through compact differential evolution algorithm with compact adaption schemes on control parameters," *Neurocomputing*, vol. 458, pp. 526–534, 2021.

[13] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[14] G. A. Miller, "WordNet," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[15] E. Geller, M. Gajek, A. Reibach, and Z. Łapa, "Applicability of wordnet architecture in lexical borrowing studies," *International Journal of Lexicography*, vol. 34, no. 1, pp. 92–111, 2021.
[16] M. AlMousa, R. Benlamri, and R. Khoury, "Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in WordNet," *Knowledge-Based Systems*, vol. 212, pp. 1–27, 2021.
[17] X. Xue and J. Zhang, "Matching large-scale biomedical ontologies with central concept based partitioning algorithm and adaptive compact evolutionary algorithm," *Applied Soft Computing*, vol. 106, pp. 1–11, 2021.
[18] Y. Xue, H. Zhu, J. Liang J, and A. stowik, "Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification," *Knowledge-Based Systems*, vol. 227, pp. 1–17, 2021.
[19] F. A. Zainuddin and M. F. Abd Samad, "Comparison of crossover in genetic algorithm for discrete-time system identification," *International Review of Mechanical Engineering (IREME)*, vol. 15, no. 2, pp. 59–66, 2021.
[20] J. Al-Afandi and A. Horvath, "Adaptive gene level mutation," *Algorithms*, vol. 14, no. 1, pp. 1–18, 2021.
[21] R. P. Van Heerden, B. Irwin, and I. Burke, "Classifying network attack scenarios using an ontology," in *Proceedings of the 7th International Conference on Information-Warfare & Security*, pp. 311–324, Seattle, WA, USA, March, 2012.
[22] J.-b. Gao, B.-w. Zhang, X.-h. Chen, and Z. Luo, "Ontology-based model of network and computer attacks for security assessment," *Journal of Shanghai Jiaotong University*, vol. 18, no. 5, pp. 554–562, 2013.
[23] A. Souag, C. Salinesi, I. Wattiau, and H. Mouratidis, "Using security and domain ontologies for security requirements analysis," in *Proceedings of the 2013 IEEE 37th Annual Computer Software and Applications Conference Workshops*, pp. 101–107, Washington, DC, USA, July 2013.
[24] F. Massacci, J. Mylopoulos, F. Paci, T. T. Tun, and Y. Yu, "An extended ontology for security requirements," in *Proceedings of the International Conference on Advanced Information Systems Engineering*, pp. 622–636, London, UK, June 2011.
[25] T. Takahashi, Y. Kadobayashi, and H. Fujiwara, "Ontological approach toward cybersecurity in cloud computing," in *Proceedings of the 3rd International Conference on Security of Information and Networks*, pp. 100–109, New York, NY, USA, September 2010.
[26] L. Youseff, M. Butrico, and D. Da Silva, "Toward a unified ontology of cloud computing," in *Proceedings of the Grid Computing Environments Workshop*, pp. 1–10, Austin, TX, USA, November 2008.
[27] S. Beji and N. El Kadhi, "Security ontology proposal for mobile applications," in *Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pp. 580–587, Washington, DC, USA, May 2009.
[28] H. Razouki, "Security policy modelling in the mobile agent system," *International Journal of Computer Network and Information Security*, vol. 11, no. 10, pp. 26–36, 2019.
[29] C. Choi, J. Choi, and P. Kim, "Ontology-based access control model for security policy reasoning in cloud computing," *The Journal of Supercomputing*, vol. 67, no. 3, pp. 711–722, 2014.
[30] K. Arbanas and M. Cubrilo, "Ontology in information security," *Journal of Information and Organizational Sciences*, vol. 39, no. 2, pp. 107–136, 2015.