International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 1135-1144

https://doi.org/10.70517/ijhsa46388

Research on Computer Software Copyright Infringement Determination Method and Legal Response Mechanism Combined with Data Mining Algorithm

Shuang Yang[1,*]
[1] School of Political Science and Law, Sichuan University of Arts and Science, Dazhou, Sichuan, 635000, China
Corresponding authors: (e-mail: Yangshuang0814@126.com).

Abstract Under the rapid development of information technology, the problem of computer software copyright infringement is becoming more and more prominent, and the traditional infringement determination methods are facing the challenges of low efficiency and lack of accuracy when dealing with massive data and complex code logic. Considering the different characteristics of computer software copyright infringement determination and traditional works, this paper discusses the important rules of copyright infringement determination (substantial similarity plus contact rule) from the perspective of academic theory. Under the guidance of the rule theory, the copyrighted software and the allegedly infringing software source code data are transformed into TF-IDF vectors using the optimized K-Means mean clustering algorithm. A high-dimensional mixed-attribute data similarity algorithm is used to calculate the similarity of the software source code data vectors and obtain the final mining results. Thereby, a computer software copyright infringement determination method based on the data mining algorithm is proposed. In the infringement determination experiment, the accuracy rate of this method is as high as 85% and above, and the highest recall rate is only 87.60%, which can provide powerful technical support for the determination of computer software copyright infringement.

Index Terms computer software copyright, substantial similarity plus contact, K-Means algorithm, similarity algorithm

I. Introduction

Computer software industry is an important component of the contemporary national economy, and is a fundamental and strategic industry of social informatization [1]. Nowadays, countries around the world have stepped into the information age, and the use of computer software can be said to be ubiquitous. E-commerce platform, taxi software, financial computing, industrial production and other diversified fields are inseparable from the operation of computer software [2], [3]. The influence of computer software technology has long exceeded the situation at the beginning of its birth, and it plays an extremely important role in the whole national economy covering science and technology, culture, national defense and other aspects [4], [5]. Therefore, to promote the healthy development of software industry and national economic informatization is the inevitable pursuit of countries around the world [6].

Along with the high-speed development of China's software industry, corresponding copyright infringement has arisen. Computer software is the result of the brain work of software developers and related personnel, and it is a type of product that contains and condenses a large number of intellectual labor achievements. Computer software from the advance research, actual development, advertising and publicity to put into operation and use to spend a lot of manpower and material resources, while requiring huge financial support [7]. However, computer software is a typical product of information form, which is easy to be copied and the cost of copying is very low [8], [9]. Therefore, the protection of computer software is not to be taken lightly to encourage software developers to research and innovate, and the software industry to flourish, and should be severely punished for software piracy [10]-[12].

With the common use of computer software, developers and enterprises have also found its own commercial value, so there is bound to be a conflict of legal protection between software patent protection and open sharing. For this reason, relevant scholars have carried out research and analysis related to software copyright protection. Arai, Y. showed that the implementation of copyright protection for software effectively prevents reverse engineering such as reverse engineering and software copying, both for users under patent protection and for the benefit of society through the implementation of innovative technologies prompted by copyright protection [13]. Abbott, R. discusses the feasibility of granting patents to computer-generated works, arguing that this type of

product provides a huge creative economy for today's society and that intellectual property laws should be updated in time to provide them with appropriate patent protection [14]. Jaszi, P. et al. also pointed out the drawbacks of software copyright protection, namely, that the irrational code of practice of software copyright protection hinders the dissemination of software for digital culture and expression, and that the norms and beliefs of the law need to be improved in order to realize the innovative work of the code [15]. For this reason, Atanasova, I. proposed the core idea of reducing the problem of digital copyright infringement, emphasizing that reasonable protection of digital copyrights should be carried out in a way that safeguards not only the interests of copyright holders in obtaining fair remuneration, but also the interests of users in obtaining reasonable access to copyrighted material [16]. Basanagoudar, V. and Srekanth, A. examined the issue of copyright disputes over code in open source software, further identifying the infringement involved in generative AI copying of code by discussing the principle of fair use of open source code in terms of economic damages and personal rights [17]. Wu, Y. et al. describe and analyze the license consistency of source code in open source software, showing that discovering and maintaining source code provenance will facilitate the identification of software infringements and contribute to the protection of software copyrights [18].

It is not difficult to find, at present, the computer software infringement rules in the legislation does not make clear provisions, but borrowed in practice is usually used in the “substantial similarity + contact” judgment mode as the software infringement rules [19]-[21]. Therefore, only by clarifying how to determine the rules in the field of computer software infringement determination, in order to make computer software infringement determination more standardized and normalized [22], [23]. In this context, data mining algorithms in the determination of computer software infringement determination method and legal response mechanism, in theory and practice has great practical value.

This paper firstly discusses the substantial similarity plus contact rule in detail, and makes it clear that many elements of computer software, such as function, interface, program structure, algorithm and data structure, should be included in the scope of software copyright protection. Secondly, K-Means mean clustering algorithm is introduced to transform the source code data in computer software into TF-IDF feature vectors and optimize the K-Means algorithm. At the same time, applying the principle of spatial dimension transformation, the high-dimensional mixed attribute data of computer software source code methods, statements and expressions are transformed, and similarity calculation is carried out, so as to realize the mining of source code data of copyrighted software and software accused of infringing copyright. By defining the scope of computer software copyright and designing mathematical calculation methods, a method for determining computer software copyright infringement based on data mining algorithm is established. Finally, the computer software copyright infringement cases accepted by Court F are selected as the research object to analyze the main discretionary factors of infringing damages and the reference of damages, and to evaluate the operational performance of the method in this paper.

II. Substantive similarity plus rules of engagement

II. A. Overview of the substantive similarity plus contact rule

The “substantial similarity plus access” rule is an important rule for the determination of copyright infringement, which is based on the theory of information dissemination, adjusts the interrelationship between the creator's access to known conditions and the originality of the new creation, and provides theoretical support for the reasonable allocation of the burden of proof in judicial practice. The “substantial similarity plus exposure” rule covers three main aspects, namely, that the two works constitute substantial similarity, that the alleged infringer was exposed to the prior copyrighted work, and that there are no reasonable defenses.

The proof of the contact element in the substantial similarity plus contact rule includes, first of all, that the prior work alleged to have infringed has been made public, i.e., published in the sense of copyright, and the published work has entered the public domain so that all persons can freely access it. In addition, if there is a clear similarity between the later work and the prior work, and the alleged infringer cannot give a reasonable explanation for the similar part or excludes the possibility that the latter work is an independent creation, the condition of "contact" can also be determined to be met The importance of the elements is not equivalent, "substantial similarity" occupies the primary position in the judgment of infringement, and the fact of "contact" is judged only after the two works have been found to be similar, and unless there is evidence to the contrary, it is often possible to directly determine that the infringer has contact behavior based on the objective similarity, so it can be seen that the "contact" condition is in an auxiliary position in the substantive similarity plus contact rule, and the two are in a progressive relationship when judging the infringing work.

"Exclusion of reasonable interpretation" is a supplement to the principle of substantial similarity plus contact, in the case that the alleged infringing work is judged to be substantially similar and the defendant has been exposed to the prior work, the defendant is given a final opportunity for interpretation, the so-called "reasonable

interpretation" is not the subjective rationality of the literal understanding, but to find relevant legal support in the legal provisions, first of all, the most common situation is that the expression is limited, although the copyright law only protects the author's expression, but when a certain idea has only the only way to express it, If the only expression is protected, then it is equivalent to the protection of the thought itself, which is obviously in conflict with the dichotomy of thought expression, and another common situation is the case of fair use.

II. B.Substantial similarity plus contact rules software domain review

Substantial similarity determination of computer software should firstly distinguish between two situations, firstly, for textual similarity, the alleged infringing software should be compared with the source code or target code of the prior copyrighted software, and the proportion of overlap between the two program codes should be used to determine whether it constitutes substantial similarity, and this comparison can also be adopted to determine the infringing works of the manual documents of the computer software. Secondly, for the non-text part of the similarity of the two software should be compared from the overall comparison of the two software and can not be limited to the comparison of the code, however, for the comparison of the similarity of the non-text part of the comparison of the scope of the comparison can not be unlimited expansion of the two software can not be realized on the basis of the same function of the two software to claim that the two constitute substantial similarity, the function of the software belongs to the scope of the idea of not subject to copyright protection, and at the same time, for the part of the structure of the software Although the copying of the content may form two output content and technical effect of completely different software, but if there is substantial similarity may still constitute an infringement of works, so the judgment of the substantial similarity of the software must not be the same as the final function of the judgment of substantial similarity, in the preliminary comparison of the program code after the software process structure and other components of the analysis and comparison of the substantial similarity of the judgment.

The dissemination of computer software is often to the Internet as a medium, with convenient storage, dissemination of a wide range of characteristics, so the infringer wants to contact the software copyright holder's software is very easy, according to the principle of civil procedure law who claims who proves that the infringer has been in contact with the software in the case of the burden of proof borne by the right to the infringer, due to the previously mentioned software contact in today's era is very simple, so for the The degree of “contact” that the right holder needs to prove should be relaxed, and the plaintiff is required to show that the defendant has actually touched the original work of fact in practice, there are great difficulties, if we strictly follow this requirement is not conducive to the protection of the rights of copyright holders, and it is easy to cause the phenomenon of unfairness. Therefore, the degree of proof of “contact” should be limited to proving that the defendant has the possibility of contacting the prior work, and it should be more reasonable to prove that the defendant constitutes contact by reaching such a standard.

III. Methods for determining copyright infringement based on data mining algorithms

III. A. K-Means Algorithm Optimization

In order to deeply analyze the correlation between the copyrighted software and the alleged copyright infringing software, the K mean clustering algorithm (K-means) is used to classify them, and the data of their software source code and software content are converted into TF-IDF vectors, and optimized for the K-means algorithm by randomly determining the number of clusters in the dataset and the center of clustering. First of all, the introduction of variance ratio criterion (CH) index to determine the number of clusters, CH index is a kind of index used to assess the quality of clustering, through the calculation of the variance within the cluster and the variance between the clusters to measure the closeness of the clusters and the degree of separation, the larger the CH index represents the more tightly clustered, the more dispersed the clusters are between the clusters, the better the clustering effect. The calculation steps are as follows:

(1) Calculate the inter-cluster sum of squares (BCSS). $BCSS$ is used to measure the separation of samples between different clusters, the larger the $BCSS$, the more separated the clusters are, and the better the clustering effect. The formula for $BCSS$ is shown in equation (1):

$$BCSS = \sum_{i=1}^{k} l_i \left\| m_i - m \right\|^2 \tag{1}$$

where k denotes the number of clusters. l_i denotes the number of data points in cluster i. m_i denotes the centroid of cluster i; m denotes the centroid of the overall data set. $\left\| m_i - m \right\|^2$ denotes the square of the Euclidean distance between the center point m_i of the cluster and the center point m of the overall data set.

(2) Calculate the within-cluster sum of squares (WCSS), $WCSS$ is used to measure the tightness of the data points within each cluster, the smaller $WCSS$ is, the tighter the clusters are, and the better the clustering effect is. The formula for $WCSS$ is shown in equation (2):

$$WCSS = \sum_{i=1}^{k} \sum_{x \in c_i} \left\| x - m_i \right\|^2 \tag{2}$$

where x denotes each data point within the cluster. c_i denotes the cluster to which data point x belongs. $\left\| x - m_i \right\|^2$ denotes the square of the Euclidean distance between the data point x in the cluster and the cluster center point m_i.

(3) Based on the values of $BCSS$ and $WCSS$, the value of CH index is calculated as in equation (3):

$$CH(k) = \frac{BCSS}{k-1} \Big/ \frac{WCSS}{l-k} \tag{3}$$

where l denotes the number of all data points.

Next, the average distance is introduced to optimize the clustering center selection. The steps are as follows:

(1) Calculate the Euclidean distance $d\left(x_i, x_j\right)$ between the data points, which is calculated as equation (4):

$$d\left(x_i, x_j\right) = \sqrt{\sum_{i \neq j, i, j=1}^{n} \left(x_i - x_j\right)^2} \tag{4}$$

where $x_i = \left(x_{i1}, x_{i2}, \cdots, x_{iz}\right)$ and $x_j = \left(x_{j1}, x_{j2}, \cdots, x_{jz}\right)$ denotes any two data points with dimension equal to z.

(2) Calculate the average distance $d_{avg}\left(x_i\right)$ from each data point to the existing clustering center, which is calculated as in Equation (5):

$$d_{avg}\left(x_i\right) = \frac{1}{n} \sum_{j=1}^{n} d\left(x_i, T_j\right) \tag{5}$$

where n denotes the number of existing clustering centers. $d\left(x_i, T_j\right)$ denotes the Euclidean distance from data point x_i to clustering center T_j. T_j denotes the existing clustering centers $T_{j1}, T_{j2}, \cdots, T_{jz}$.

The steps of the optimized K-means algorithm are as follows:

(1) Calculate the value of CH index for different number of clusters in the dataset and select the number of clusters k with the highest CH index value as the optimal value.

(2) Calculate the Euclidean distance $d\left(x_i, x_j\right)$ between any two data points to measure their similarity.

(3) Compare each $d\left(x_i, x_j\right)$ and select one of the data points with the smallest distance as the first initial clustering center.

(4) Calculate the average distance $d_{avg}\left(x_i\right)$ from each data point to the currently selected clustering center, and select the data point with the largest average distance as the next initial clustering center.

(5) Repeat step (4) until k clustering centers are selected.

(6) Perform an optimized K-means algorithm based on the best k values determined by step (1) and the clustering centers determined by step (5) to arrive at a clustering result.

III. B. Similarity Calculation for High-Dimensional Mixed Attribute Data

Computer software high-dimensional mixed-attribute data in the database are usually characterized by three kinds of attributes, including methods, statements and expressions. Calculating software source code data similarity is the necessary basis for data mining, therefore, the similarity of multi-attribute data is first calculated to obtain the final mining results.

In order to alleviate the memory pressure of the database in the network system and improve the stability of the external algorithm, based on the data denoising results, the matrix transformation of the data is performed by utilizing the principle of spatial dimension transformation. Suppose D_j denotes the structure matrix of the j th column in the database, and $d_{ji}\left(i = 1, 2, \cdots, m\right)$ is the data object in it, which denotes the data in the i th column and the j th row. If there are a total of m high-dimensional mixed attribute data in each row of the matrix, D_j can be expressed as equation (6):

$$D_j = \left(d_{j1}, d_{j2}, \cdots, d_{jm}\right) \tag{6}$$

If the number of data objects in the structure matrix is n, the total matrix of high-dimensional mixed-attribute data in the database D is represented using the transpose transformation of the matrix as in equation (7):

$$D = \left(D_1, D_2, \cdots, D_n\right)^T \tag{7}$$

Now, mine the high-dimensional mixed-attribute data x of the set X in the total matrix D and set the correlation of its individual attributes to $sim(X,Y)$ (the corresponding sets are denoted by Y and X). The mined sample data is represented as s, then s should meet the following conditions as in equation (8):

$$s = sim(X,Y)\left[Freq(x,X) - Freq(y,Y)\right] / D \tag{8}$$

In the above equation, $Freq$ denotes the probability of the condition occurring.

The scale characteristics of high-dimensional mixed-attribute data include both categorical and numerical types, assuming that there are two mutually independent data in the dataset X $X_i\left(x_{i1}, x_{i2}, \cdots, x_{iq}\right)$ and $X_j\left(x_{j1}, x_{j2}, \cdots, x_{jq}\right)$, then the distance between the two data is calculated by equation (9):

$$\left|X_i, X_j\right| = s \cdot \sum_{k=1}^{q}\left(x_{ik} - x_{jk}\right)^2 \tag{9}$$

In the above equation, q denotes the dimension of the data set X. s denotes the conditional sample data. x_{ik}, x_{jk} denote the k th dimension reference samples of X_i and X_j, respectively.

The similarity between data X_i and cluster U_j is defined by the formula in equation (10):

$$\left|X_i, U_j\right| = \left|X_i, X_j\right| \sum_{k=1}^{q}\left(x_{ik} - x_{jk}\right)^2 \tag{10}$$

In the above equation, U_j denotes the frequency vector of the data clusters' fetch values. U_{jk} denotes the *k*-dimensional fetch. The distance between data X_i and the reference sample data is calculated by equation (11):

$$H(\upsilon) = \left|X_i, U_j\right|\left(\frac{r_k}{2\sigma^2}\right) \tag{11}$$

In the above equation, r_k denotes the independent components of the data. σ denotes the candidate information target parameter.

For the form of categorical data similarity measure, if two categorical data $X_{\mathrm{i}}(x_{i1}, x_{i2}, \cdots, x_{iq})$ and $X_j(x_{j1}, x_{j2}, \cdots, x_{jq})$ can exist in a data matrix at the same time, then the similarity between these two data is calculated as equation (12):

$$S(X_i, U_j) = \frac{\sum_{k=1}^{q} \delta_{ij}(x_{ik}, x_{jk})}{H(\upsilon)} \tag{12}$$

In the above equation, δ_{ij} denotes the probability of the simultaneous occurrence of methods, statements and expressions of the high-dimensional mixed-attribute data, which is computed as in equation (13):

$$\delta_{ij}(x_{1i}, x_{2i}) = \begin{cases} 0, x_{1i} = x_{2i} \\ 1, x_{1i} \neq x_{2i} \end{cases} \tag{13}$$

If the similarity between the categorical data X_i and the categorical clustering A is considered to be the average similarity between the data and the other data, the definition of the similarity is expressed as Equation (14):

$$M(X_i, A) = \frac{\left\| S(X_i, X_j) \right\|}{A \times q} \tag{14}$$

By extending the above definition, the similarity of the clustering of two types of data can be obtained by setting A, B as the clusters of this type of data, respectively, then the similarity bounds are as in equation (15):

$$N(A, B) = \frac{\left\| M(X_i, A) \right\|}{A \times B \times q} \tag{15}$$

The clustering boundaries are used as an extraction criterion for the data similarity measure and are described by equation (16):

$$l(x_i, U_j) = \frac{\left\| N(A, B) \right\|}{l_n + \omega_i l_c} \tag{16}$$

In the above equation, U_j denotes the fetch frequency vector of the data cluster. ω_i denotes the discrete point representation degree of the i th data object. l_n, l_c denote the similarity of n, c class of data attribute types as in Eqs. (17)-(18), respectively:

$$l_n = l(x_i, U_j) \sum (x_{ik} - U_{jk})^2 \tag{17}$$

$$l_c = l(x_i, U_j) \sum \sigma(x_{ik} - U_{jk})^2 \tag{18}$$

Based on the attribute classification of high-dimensional mixed-attribute data, we analyze and calculate the similarity of numerical and sub-type association of high-dimensional mixed-attribute data, which is convenient to realize the mining of software source code data.

IV. Performance Evaluation of Infringement Determination Methods

This chapter takes the computer software copyright infringement cases accepted by the Intellectual Property Court of Intermediate People's Court of F during the period of 2018-2024 as the research data, and combs through the changes in the number of cases accepted annually as well as the changes in the number of cases of right holders. Combined with this research data, it analyzes the discretionary factors of computer software copyright infringement damages as well as the differences in damages, and then launches a comparison of the operational performance of this mining algorithm-based computer software copyright infringement determination method with similar methods.

IV. A. Research Objects and Data Sets

The Intellectual Property Court of Intermediate People's Court of F (hereinafter referred to as Court F) was selected to receive 956 cases of computer software copyright infringement disputes from January 2018 to December 2024, and the number of cases received annually fluctuates as shown in Figure 1.

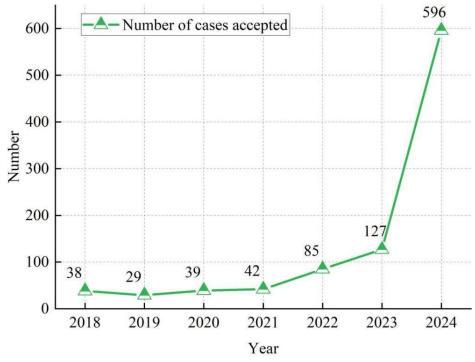

Figure 1: Number of computer software copyright dispute cases accepted

The annual proportion of cases involving right holders is shown in Fig. 2. Such cases present the following characteristics: first, in terms of the subject of dispute litigation, cases in which the plaintiff is a foreign party

accounted for a slightly higher proportion before 2020, but with the increased awareness of IPR protection among Chinese domestic enterprises and the rapid development of the software industry, the number of cases in which Chinese domestic enterprises defended their rights has increased sharply in the past two years. The percentage of cases involving foreign right holders has dropped significantly, and the number of lawsuits between Chinese domestic parties has skyrocketed, reflecting to some extent the booming development of China's domestic software industry.

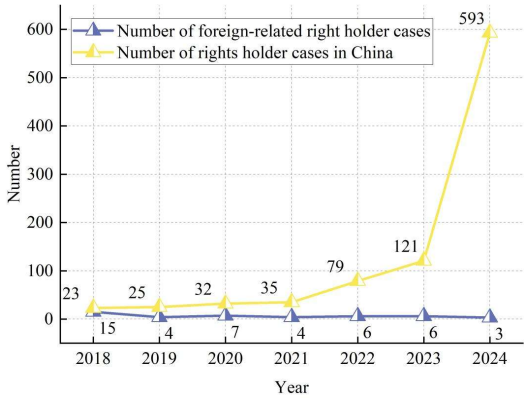

Figure 2: Annual distribution of foreign-related rights holder cases

IV. B. Discretion and compensation for tort damages

IV. B. 1) Main discretionary factors

In order to facilitate a more intuitive understanding of the major infringing discretionary factors, combining past research and the trial of the Court F case, the 15 major discretionary factors in the court's discretionary compensation were distilled as follows: (V1) the nature of the infringing behavior, (V2) the plaintiff's reasonable expenditures, (V3) the duration of the infringing behavior, (V4) the defendant's degree of subjective fault, and (V5) the market price of the software in question, (V6) the circumstances of the infringement, (V7) the quantity of the infringing products (infringing copies), (V8) the type of software involved, (V9) the consequences of the infringement, (V10) the scale of the defendant's business, (V11) the reasonable royalties for the software involved, (V12) the popularity of the software involved, (V13) the manner of the infringement, (V14) the plaintiff's damages, and (V15) the infringing The number and proportion of cases involving the 15 major discretionary factors are shown in Figure 3, of which (V1) the nature of the infringement, (V2) the reasonable expenses of the plaintiff, and (V3) the duration of the infringement accounted for the highest proportion, involving 236, 132 and 116 cases in that order.

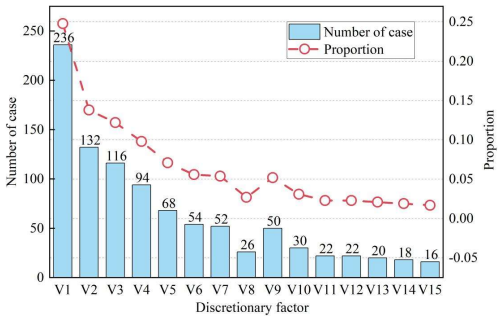

Figure 3: The number and proportion of 15 main discretionary factors involved

IV. B. 2) Differences in damages for different types of software

For the purpose of analysis, the software involved in Court F was combined into the following seven categories in accordance with the principle of proximity: (S1) game software, (S2) industry application software, (S3) system software and office software, (S4) network communication security retrieval software, (S5) enterprise management and financial management software, (S6) auxiliary design and auxiliary manufacturing software, and (S7) other software, on the basis of which the various categories of software were analyzed The average amount of compensation requested, the average amount of actual compensation, the average payout rate and the average unit price of compensation are shown in Table 1.

Table 1: Copyright infringement cases for different categories of software

Software	S1	S2	S3	S4	S5	S6	S7
Number of cases	532	221	58	46	35	27	-919
Proportion(%)	55.65	23.12	6.07	4.81	3.66	2.82	3.87
Average amount of compensation claimed (ten thousand yuan)	106.54	330.34	42.66	111.08	63.32	142.26	38.12
Average actual compensation (ten thousand yuan)	21.48	56.43	25.43	66.51	6.07	36.12	5.326
Average payout ratio(%)	20.16	17.08	59.61	59.88	9.59	25.39	13.97
Average unit price (ten thousand yuan)	19.66	45.51	19.37	24.59	4.96	31.61	5.33

As can be seen from Table 1, there are large differences in the average amount of compensation requested, the average amount of actual compensation, the average payout rate and the average unit price of compensation for each type of software.

IV. C. Infringement Determination Experiment

In order to validate the effectiveness of this paper's method, two groups of experiments are designed, the first group of experiments is to compare the determination accuracy as well as the recall rate of the commonly used method (Y1) FP-growth method, (Y2) Improved Fuzzy Neural Networks method, and (Y3) this paper's method, on the case dataset collated above. The second group is to compare the performance of this paper's method on different categories of infringement determination book retrieval results. Patents with coverage of 0.6 or more are set as suspected infringement determinations. The comparison of the accuracy rate of the three methods is shown in Fig. 4, and the comparison of the recall rate is shown in Fig. 5. Compared with the other two methods, the accuracy rate of the infringement determination method of this paper's method is higher, which is always maintained at 85% and above, and the performance of the (Y1) FP-growth method is more unsatisfactory, with the highest accuracy rate of only 80.68%. In addition, this paper's method also maintains an excellent performance in the recall rate compared to other methods, with a maximum of only 87.60%, while the (Y1)FP-growth method has a minimum recall rate of 90.98%, which is much higher than this paper's method.

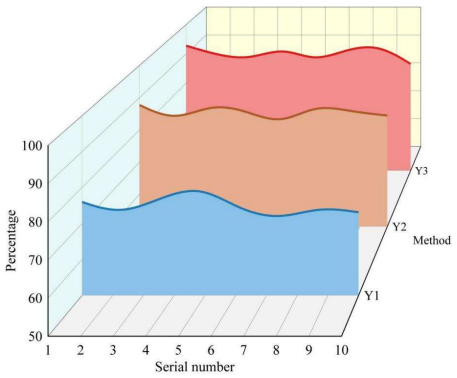

Figure 4: Comparison of accuracy of different retrieval methods

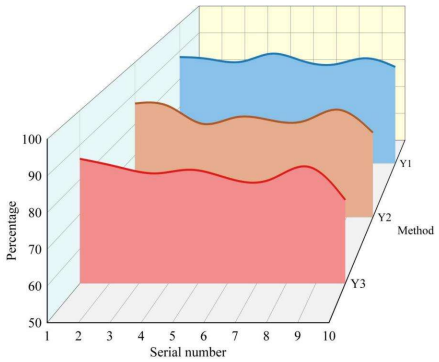

Figure 5: Comparison of recall rates of different retrieval methods

Infringement judgment is usually categorized into (a) fault liability, (b) no-fault liability and (c) fair liability. In order to verify the retrieval effect of this paper's algorithm for different categories of infringement determinations, the experimental data were categorized and re-added some artificial constructed data, and after several experiments, the average accuracy and recall rate were calculated, and the running results of this paper's method were obtained in Fig. 6. It can be found in the Fig. that this paper's proposed algorithm has the best infringement retrieval effect for the claims in the category of fair liability, with the accuracy rate as high as 0.875, with the category of no-fault liability next, and the category of no-fault liability next. The no-fault liability class is the second most effective, while the fault liability class is less effective. Analyzing the reasons, it is due to the fact that there are more nouns and domain terms in the claims of the fair liability class, and the statements are relatively simple, which makes the accuracy of feature extraction higher, and the weight of nouns and terms is higher in the calculation of feature similarity. On the other hand, the statements in the fault liability claims are relatively complex in logic and contain a large number of verbs, prepositions and adverbs, which leads to a poor effect of dependent syntax analysis and a low accuracy rate of feature extraction, resulting in a slightly poorer final retrieval effect.

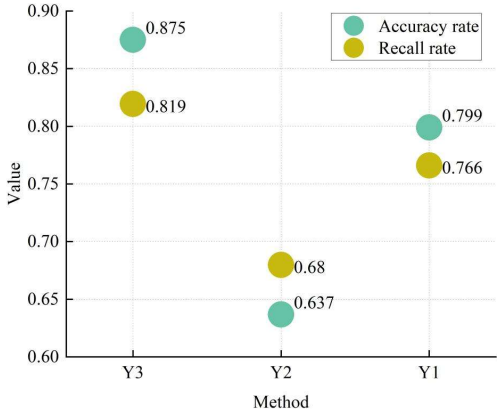

Figure 6: The performance of this method in different categories of judgment books

V. Conclusion

The difficulty of computer software copyright infringement determination lies in the complexity of its internal components and the professional requirements for the determination method. In this paper, based on the theory of substantial similarity plus contact rule, combined with K-Means mean clustering algorithm and data mining algorithm of similarity calculation, we design the computer software copyright infringement determination method.

The designed determination method shows superior performance in operation, and its determination accuracy on the actual case dataset is stable at 85.00% and above, and the highest recall rate is only 87.60%. Combined with the characteristics of the computer copyright infringement cases accepted by the research object (Court F) between 2018 and 2024, the study proposes to clarify the scope of copyright protection, improve the registration system of software works, and unify the software infringement determination standard, so as to provide effective references for the determination method and legal response mechanism of the current computer software copyright. The study proposes to clarify the scope of copyright protection, improve the software work registration system, and standardize the software infringement determination standard.

References

[1] Yang, J., Jimenez, C., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., & Press, O. (2024). Swe-agent: Agent-computer interfaces enable automated software engineering. Advances in Neural Information Processing Systems, 37, 50528-50652.

[2] Gonçalves, R., Rocha, T., Martins, J., Branco, F., & Au-Yong-Oliveira, M. (2018). Evaluation of e-commerce websites accessibility and usability: an e-commerce platform analysis with the inclusion of blind users. Universal Access in the Information Society, 17, 567-583.

[3] Vignato, J., Inman, M., Patsais, M., & Conley, V. (2022). Computer-assisted qualitative data analysis software, phenomenology, and colaizzi's method. Western journal of nursing research, 44(12), 1117-1123.

[4] Bessen, J. (2020). Information technology and industry concentration. Journal of Law & Economics, 63(3), 531.

[5] Joseph, K. J. (2014). India's Software Industry in Transition: Lessons for Other Developing Countries and Implications for South-South Cooperation. Productivity, 54(4), 329.

[6] Khan, H. H., Malik, M. N., Alotaibi, Y., Alsufyani, A., & Alghamdi, S. (2021). Crowdsourced Requirements Engineering Challenges and Solutions: A Software Industry Perspective. Computer Systems Science & Engineering, 39(2).

[7] Ebert, C., & Louridas, P. (2023). Generative AI for software practitioners. IEEE Software, 40(4), 30-38.

[8] Kretschmer, M., Meletti, B., & Porangaba, L. H. (2022). Artificial intelligence and intellectual property: copyright and patents—a response by the CREATe Centre to the UK Intellectual Property Office's open consultation. Journal of Intellectual Property Law and Practice, 17(3), 321-326.
[9] Emrich, G. (2021). Cracking the code: How to prevent copyright termination from upending the proprietary and open source software markets. Fordham L. Rev., 90, 1245.
[10] Abdulsala, F. (2022). An Analysis on Copyright Infringement over Internet with Reference to the IT Act and the Copyright. Part 2 Indian J. Integrated Rsch. L., 2, 1.
[11] Toliwal, M. (2022). Copyright Protection for Computer Software: A Critical Analysis. Jus Corpus LJ, 3, 1041.
[12] Menell, P. S. (2017). Rise of the API copyright dead: An updated epitaph for copyright protection of network and functional features of computer software. Harv. JL & Tech., 31, 305.
[13] Arai, Y. (2018). Intellectual property right protection in the software market. Economics of innovation and new technology, 27(1), 1-13.
[14] Abbott, R. (2020). Artificial intelligence, big data and intellectual property: protecting computer generated works in the United Kingdom. In Research handbook on intellectual property and digital technologies (pp. 322-337). Edward Elgar Publishing.
[15] Jaszi, P., Aufderheide, P., Butler, B., & Cox, K. L. (2019). Cracking the copyright dilemma in software preservation: Protecting digital culture through fair use consensus. Journal of Copyright in Education and Librarianship, 3(3).
[16] Atanasova, I. (2019). Copyright infringement in digital environment. Economics & Law, 1(1), 13-22.
[17] Basanagoudar, V., & Srekanth, A. (2023). Copyright Conundrums in Generative AI: Github Copilot's Not-So-Fair Use of Open-Source Licensed Code. J. Intell. Prot. Stud., 7, 58.
[18] Wu, Y., Manabe, Y., Kanda, T., German, D. M., & Inoue, K. (2017). Analysis of license inconsistency in large collections of open source projects. Empirical Software Engineering, 22, 1194-1222.
[19] Oman, R. (2017). Computer Software as Copyrightable Subject Matter: Oracle v. Google, Legislative Intent, and the Scope of Rights in Digital Works. Harv. JL & Tech., 31, 639.
[20] Mulligan, C. (2017). Copyright Without Copying. Cornell JL & Public Policy, 27, 469.
[21] Chen, L. Y. U., & Linjun, J. I. A. N. G. (2023). Open Source Software: A Study on the Copyright Licensing Investigation System of DMCA in the United States. Journal of Library and Information Sciences in Agriculture, 35(8), 78.
[22] Yu, Z., Wu, Y., Zhang, N., Wang, C., Vorobeychik, Y., & Xiao, C. (2023, July). Codeipprompt: intellectual property infringement assessment of code language models. In International conference on machine learning (pp. 40373-40389). PMLR.
[23] Katzy, J., Popescu, R., Van Deursen, A., & Izadi, M. (2024, April). An exploratory investigation into code license infringements in large language model training datasets. In Proceedings of the 2024 IEEE/ACM First International Conference on AI Foundation Models and Software Engineering (pp. 74-85).