

## Research Article

# The Optimization of Classroom Teaching in Colleges and Universities Based on Network Topology

Yongqiang Wang  and Na Zhang

*Railway Police College, Zhengzhou 450053, China*

Correspondence should be addressed to Yongqiang Wang; wangyongqiang@rpc.edu.cn

Received 6 October 2021; Revised 28 October 2021; Accepted 3 November 2021; Published 22 November 2021

Academic Editor: Le Sun

Copyright © 2021 Yongqiang Wang and Na Zhang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve the teaching effect of programming courses, this article applies network topology to the construction of programming course teaching platform system and constructs a teaching framework combined with intelligent teaching mode. In addition, this article takes the opportunistic social network composed of mobile smart terminal devices as the application background and briefly introduces the concept, characteristics, and application scenarios of the opportunistic social network. In addition, the typical message forwarding algorithm currently studied in this paper is analyzed. Taking into account the social attributes and forwarding willingness of nodes, in order to increase the message transmission rate and reduce message transmission delay and network overhead, two message transmission strategies are proposed to improve the operating effect of the programming course teaching platform. Finally, this paper verifies the effectiveness of the network topology in the teaching of the algorithm and programming of the intelligent teaching mode through experimental research.

## 1. Introduction

In the information age, the widespread application of Internet and other network technologies has caused tremendous changes in the way of teaching and learning. Moreover, various new teaching methods using microclasses, micro-videos, and MOOCs as the media are gradually taking shape, and new learning methods such as personalized learning, mobile learning, and flipped learning are gradually becoming popular. As the main place for imparting knowledge and skills and cultivating quality talents, schools must comply with the requirements of education informatization. At the same time, key changes have taken place in the development of teachers' curriculum, the establishment of education and teaching models, the update and delivery of learning materials, and the way students learn. This is a response to the development trend of the times and an inevitable requirement for the development of education [1].

As a new type of teaching form to promote teaching performance, the flipped classroom mode is the key to efficiently flip educational elements such as teaching concepts,

teacher and student roles, teaching methods, and evaluation methods [2]. That is, before class, students watch the learning videos independently and complete knowledge transfer, and in class teachers and students conduct discussions and exchanges, conduct collaborative exploration and internalize knowledge, and evaluate and reflect after class, and provide timely feedback. It reverses the traditional teaching model, reshapes the role of teachers and students and the relationship between teachers and students, highlights the dominant position of students, improves learning enthusiasm, and improves teaching effect and quality [3].

Object-oriented programming course is a very important language practice course learned by computer majors. This course focuses on cultivating students' ability to solve practical problems independently through algorithm design and is highly professional and practical. With the development of Internet technology, the requirements for students' object-oriented programming have gradually increased, but there are still many problems in the classroom teaching of programming courses. The above-mentioned problems will seriously affect the teaching effect, deviate

from the teaching goal of the programming design course, and cause the quality of training students to decline. Therefore, we reform the programming curriculum and traditional teaching mode by adopting the MOOC-based flipped classroom teaching concept. At the same time, we make full use of MOOC's rich and high-quality online resources while flipping the main role of teachers and students in the classroom and pay more attention to the interactive process and feedback of classroom teaching, to further optimize the teaching quality and effect of programming courses [4].

Since most students ignore the review of the class, the teacher will pay more attention to the teaching of basic grammar knowledge in the classroom teaching of the program design course and lack the training and concern for students to solve problems. Most teachers believe that students can make students master the concept, grammar, and knowledge points and then practice the teaching objectives of the course in accordance with the experimental guidance in the experimental course. But in fact, this kind of teaching does not enable students to really internalize in the early stage of classroom and absorb the knowledge points. In the next year, secondary students can not flexibly use knowledge points to solve practical problems.

The teaching resources of program design courses are still based on textbooks and experimental guidelines, lacking students' way to explore, practice, and solve problems. The final exam of the programming course is due to the paper test, so the form of assessment is mainly based on the theoretical knowledge, ignoring the essence of the programming course. As long as the assault review before the exam, you can get a high score, and practice and practice. The ability is not significantly improved. Therefore, the traditional teaching model of the reform program design curriculum, enriching teaching resources, improving students' learning energy and learning efficiency, has become an important research topic in teaching reform.

This article combines the social cognitive network to optimize the flipped classroom teaching mode of programming courses and combines the information transmission improvement algorithm to improve the transmission efficiency of teaching resources and further enhance the flipped classroom teaching effects of programming courses.

## 2. Related Work

With the development of a large number of online social networks at home and abroad, related research on online social network analysis and mining has attracted great attention from scholars at home and abroad. In today's society, due to the prevalence of social software, social network applications have had a significant impact on people's clothing, housing, and transportation.

People pay more and more attention to large-scale network graph data compression technology. Literature [5] proposed a technique for compressing the original image when it is divided. However, it is still not very good to meet the requirements of fast query. Literature [6] proposed a

dictionary-based compression method. This method is to compress web page data. It sorts the URLs in the web page graph in lexicographical order. Since URLs that are close in sequence often contain similar neighbors, using this property can achieve a better compression effect. However, due to general social network graph data, nodes often do not have dictionary attributes, and the compressed places cannot be found in lexicographic order. Literature [7] proposed an FRS algorithm, which defines a social network data node rearrangement technology to achieve the purpose of node rearrangement and then achieve the purpose of compression. Literature [8] proposed BFS-based technology, which also uses the rearrangement method of nodes and uses the difference between adjacent rows in the adjacency table after sorting to store graph data, so as to achieve the purpose of compression. Literature [9] proposed a multipoint linearization of neighborhood query, which uses some data structures to achieve compression purposes, such as quadrees. Literature [10] showed that the data in the actual graph in reality has the characteristics of power law distribution. Using this feature, literature [11] used the structural equivalence method to transform the points and edges in the original graph into structures with equivalent meanings to achieve the purpose of compression. Literature [12] proposed a method based on attributes, nonoverlapping, and covering node groups to achieve the purpose of compression. The above method considers the problem of thinking from the perspective of how to reduce the size of the original image data and does not consider what content will be lost due to the reduction of the scale and how much impact it will have on the perception of the original image.

Graph aggregation is very important in many fields, but at present, there is relatively little research and analysis work on the basis of graph aggregation. The definition of graph aggregation is to discard some noncritical information in the original graph data and reconstruct a similar simple graph to achieve the purpose of easily analyzing the original graph. In this way, intuitively speaking, it introduces some uncertain factors in the data. For example, when accurate query is required, due to the loss of additional information, there is a certain degree of uncertainty for the query results [13]. Literature [14] developed an algorithm to understand large-scale graph data through graph aggregation, which found key structures in graph data. Although the algorithm can find these key structures, it is found through experiments that the structure found by the algorithm is not completely accurate. For example, in the actual graph data, it should be a clique structure, but the algorithm recognizes multiple star structures. This is because the algorithm does not consider the relationship between the words in the proposed graph vocabulary dictionary, which leads to errors in the final recognition of key structures. At the same time, the algorithm only applies to static large-scale graph data, not to large-scale social network graphs, and has no evolutionary research.

Literature [15] found in teaching practice that when students encounter problems in classroom learning, teachers need to give students targeted help or guidance instead of imparting systematic knowledge in the classroom. As a

result, it changed the way of knowledge transfer in the classroom and allowed students to watch the recorded teaching videos before class and answer questions and tests in the classroom and provided students with learning tasks to promote students to complete the internalization of knowledge. Subsequently, based on the practical reflection on flipped teaching, literature [16] proposed new concepts such as flipped classroom, mastered classroom, and flipped learning. Moreover, it emphasized that flipped classroom is only an entry point for teaching innovation, rather than a goal or end point. From flipped classroom to mastery classroom, the ultimate goal is flipped learning. Literature [17] mentioned that the top students have a high degree of participation in classroom activities under the traditional teaching mode, while the diligent middle school students have equal opportunities to participate in classroom activities under the flipped classroom mode. Therefore, flipped classroom teaching can ensure that more students participate in classroom teaching activities. Literature [18] constructed a flipped classroom teaching model and conducted experimental demonstrations.

### 3. Flipped Classroom Network Model of Programming Courses

This article defines the network topology including the connected graph composed of nodes and links, which is  $G=(N, L)$ . Among them,  $N$  represents the set of all nodes in the network, and  $N = \{N_1, N_2, \dots, N_m\} (m \geq 1), \forall N_i \in N$ , and  $N_i$  represents the  $i$ -th node in the opportunistic social network,  $L$  represents the set of communication links between nodes, and  $L = \{L_1, L_2, \dots, L_k\} (m \geq 1), k \in [0, n(n-1)/2], \forall L_i \in L$ , and  $L_i$  represents the  $i$ -th communication link in the network.

Node activity represents the number of times a node has encountered other nodes in the community in a period of time, denoted as  $NA_i$ . Among them,  $C_{ij}$  represents the number of times that the node  $N_i$  and the node  $N_j$  meet in the period  $T$ , and the calculation formula is shown in the following equation [19]:

$$NA_i = \sum_{j=1, j \neq i}^m C_{ij}. \quad (1)$$

The larger the value of  $\sum_{j=1}^k M_{i,j}(t)$ , the higher the activity of the node, and the greater the probability that the node leaves the local community, indicating that the node encounters other community nodes more frequently in the current time period.

The frequency of encounters refers to the ratio of the total number of encounters between node  $i$  and node  $j$  in the time period  $t$  to the total number of encounters between node  $j$  and other nodes from the initial state, namely,

$$P_{(i,j)} = \frac{M_{i,j}(t)}{\sum_{j=1}^k M_{i,j}(t)}. \quad (2)$$

Among them,  $M_{i,j}(t)$  represents the total number of encounters between node  $i$  and node  $j$  during message

forwarding in the time period  $t$  in the network,  $\sum_{j=1}^k M_{i,j}(t)$  represents the total number of encounters between node  $i$  and other nodes in the network, and  $k$  represents the number of nodes in the network.

Encounter probability estimation: Encounter probability estimation refers to the probability of meeting between two nodes in the next time period based on the historical information of the nodes and the network environment without contact.

In order to predict whether two nodes will meet within a specific time  $T_c$ , it is predicted whether the node will meet other nodes within the next time  $T_c$  since the previous time  $T$  they met, that is, whether the current encounter time interval is shorter than  $T_c + T_1$ .

$$P_{cp} = P(I_1 < T_c + T_1). \quad (3)$$

It can be seen from  $T_c + T_1$  that the entire problem has been transformed into an estimate of the current time interval  $I_t$ , and the waiting time is  $T_1$ , so  $I_t \geq T_1$  and

$$P_{cp} = P(I_1 < T_c + T_1 | I_t \geq T_1). \quad (4)$$

Through the conditional probability formula, we can get

$$P(I_1 < T_c + T_1 | I_t \geq T_1) = \frac{P(I_1 < T_c + T_1 \cap I_t \geq T_1)}{P(I_t \geq T_1)}. \quad (5)$$

From formulas (3)–(5), we can get

$$P_{cp} = \frac{f(T_1 < I < T_c + T_1)}{f(I < I)}. \quad (6)$$

In formula (6), the function  $f(x)$  represents the encounter frequency of  $x$ , and the encounter frequency is reflected by historical encounter information. In short, historical information can be used to estimate the probability of encounter.

Bayesian probability: It is an interpretation of the proposition probability proposed by Bayesian theory and the process of estimating the posterior probability through prior historical information.

The Bayesian probability between the node and other nodes is calculated by the following formula [20]:

$$P(i_k | j) = \frac{P(i_k)P(j | i_k)}{\sum_{k=1}^z P(i_k)P(j | i_k)}. \quad (7)$$

Among them,  $i_1, i_2, i_3, \dots, i_n, \dots, i_z$  represent the number of nodes existing in the network, and  $k$  represents the number of historical encounter nodes of node  $i$ .

When the node message is forwarded, the source node determines the prior probabilities  $P(i_k | j) = P(i_k)P(j | i_k) / \sum_{k=1}^z P(i_k)P(j | i_k)$  and  $P(j | i_k)$  through historical encounter data, then calculates  $P(i_k | j)$  using formula (7), and determines the probability of encountering the node with other nodes through comparison. The source node can use a variety of probabilities of encounter to determine, thereby improving accuracy.

Because the mobile nodes in the opportunistic social network are dynamic and social, the smart device nodes that make up the network are divided into different

communities, which are intracommunity and intercommunity methods. The nodes belong to the same community, and the nodes are closely connected, and the number of encounters is high. Between communities, because nodes encounter fewer times when forwarding messages, the probability of encounters is also reduced.

After dividing different communities according to the social attributes of the node, the node represents the individual carrying the mobile device, the edge where the node is located represents the social relationship between the mobile device carriers, and the weight of the edge represents the strength of the social relationship between the mobile device carriers. The nodes in the network are divided according to the above method, and the result is shown in Figure 1 [21].

A single copy mechanism is used in many messaging methods and the forwarding or reception of the message directly when the node and nodes meet. In this case, there is no better encounter opportunity during forwarding. In order to solve the above problem, this paper proposes a method of selecting a Bayesian probability tree forwarding node selection method. This method combines the source node to the probability of the income node and the Bayesian probability between the nodes to select the next forwarding node. In addition, the node carries the node as the root node of the tree, and the node encountered by the node as the second- and third-level nodes of the tree. The Bayesian probability value satisfying these nodes is located on the side where the two nodes are located, and the contents of the tree node indicate that the message is successfully delivered by each node to the target node.

This paper mainly contains nodes I, A, B, C, and D in a network, as shown in Figure 2(a) below. Node  $i$  stores the generated message and forwards it to the target node  $J$ , which meets node A at a certain time  $T_0$ ; if it only depends on the probability of transmitting the message and the node, it will not be determined; send it to the node A and consider the next chance. Whether to send a message to the next node  $\alpha$  is determined according to the BFANS algorithm, as shown in Figure 2(b) below. In this case, the transmission probability size of the root node and the value of the Bayesian probability condition satisfying the node are obtained, and finally the probability value is encountered with the node. In case of the node, after encountering the node A, the probability of multiplying the node B by the B node to the target node  $J$  probability is higher than the transmission probability of the node I to A, and the node  $i$  stores the message after the node A is encountered. Node I hopes to meet the node B; repeat the above steps until the target node  $J$  meets and forwards the message.

The application scenario of opportunistic social network is shown in Figure 3. Under the premise that there is no end-to-end transmission path between node S and node D, the source node S wants to send a message to the target node D. In this case, the message needs to be forwarded with the opportunity of encountering other nodes in the network. Therefore, node S will select an intermediate node from a group of neighbor nodes ( $N_q$ ), so that the message is finally transmitted to node D through multihop communication. We assume that S chooses node N, and N forwards the

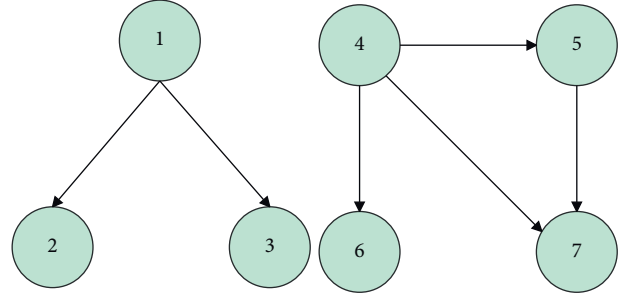


FIGURE 1: Results of community division.

message to N because there is no other neighbor node to forward it. However, node N exhibits selfishness due to resource constraints such as storage and energy and discards the message instead of forwarding it to node D. Due to resource constraints, it is difficult to detect the selfish behavior of nodes in the opportunistic social network, which reduces the message delivery rate and extends the transmission time.

Before analyzing specific problems, we first give the relevant assumptions of the system model:

- (1) The network connectivity graph is  $G = (V, E)$ , where  $V$  represents the set of all nodes in the network, and  $E = \{e = (u, v) | u, v \in V\}$  represents the set of node edges. The network  $G$  can be represented by a matrix  $A$  of size  $|V| \times |V|$ , and the value of  $A$  is as follows [22]:

$$A_{i,j} = \begin{cases} 1, e_{i,j} \in E \\ 0, e_{i,j} \notin E \end{cases} \quad (8)$$

- (2) Nodes in the network are rational and selfish nodes; that is, nodes misrepresent their own information in order to maximize benefits, rather than misrepresenting other information.
- (3) For other nodes in the network except the target node, the node has only two choices: forwarding or discarding.

In opportunistic social networks, location changes often occur according to the mobility of nodes. In order to predict the meeting position of the nodes, the calculation model based on the distance prediction of the nodes first calculates the angle formed by the moving speed of the nodes and the distance between the nodes. Secondly, the moving position of the node at the next moment is predicted and the distance is calculated. Finally, the Merkle hash tree is used to detect the existence of selfish nodes, and the best forwarding node is selected to complete the message forwarding through the node cooperative forwarding mechanism.

Encounter distance calculation: The moving speeds of the carrying node  $C$ , the neighbor node  $N$ , and the target node  $D$  are  $\vec{V}_c$ ,  $\vec{V}_n$ , and  $\vec{V}_d$ , respectively, and the angles formed by the node  $C$ , the neighbor node  $N$ , and the target node  $D$  are  $\omega_c$  and  $\omega_n$ , respectively. The calculation process is shown in formulas (9) and (10) [23]:

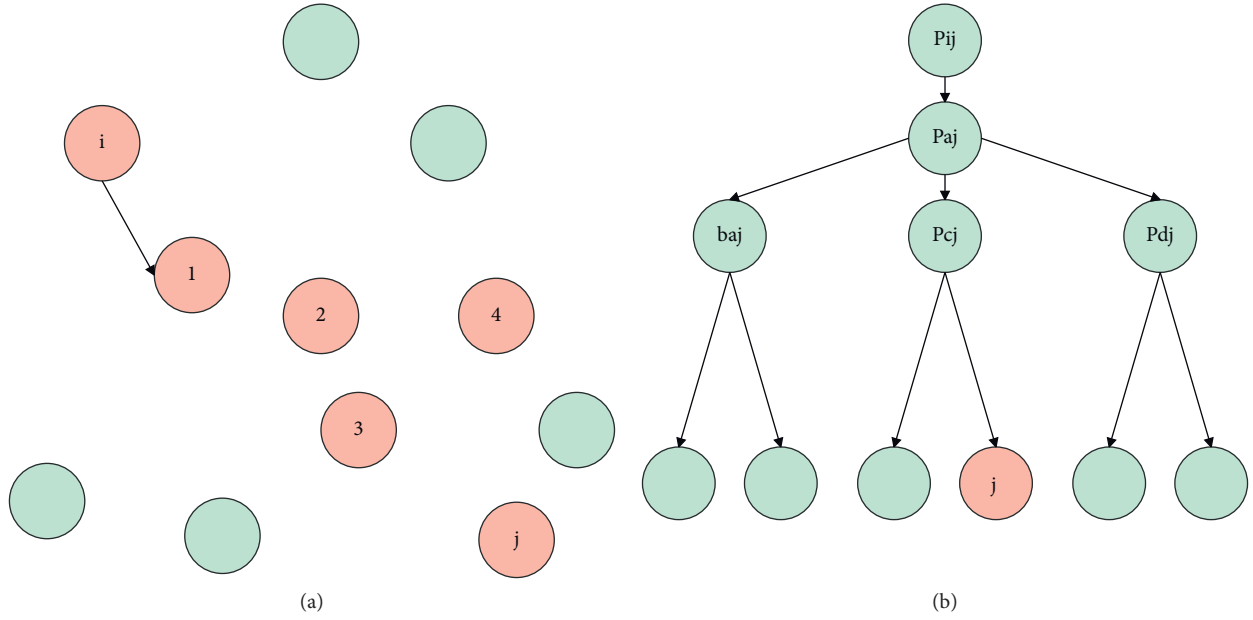


FIGURE 2: The generation process of Bayesian probability tree. (a) Network situation. (b) Bayesian probability tree.

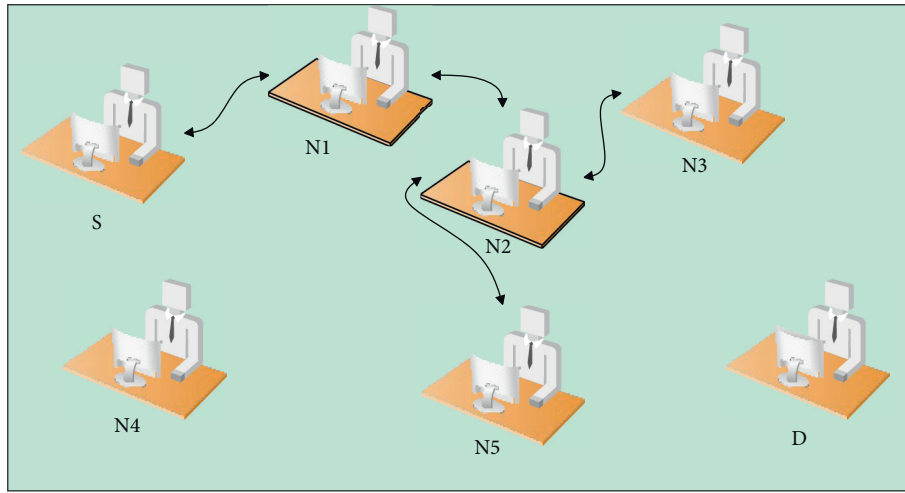


FIGURE 3: Application scenarios of opportunistic social networks.

$$\cos \omega_c = \cos \langle \vec{V}_c, \vec{V}_d \rangle = \frac{\vec{V}_c \cdot \vec{V}_d}{|\vec{V}_c| |\vec{V}_d|}, \quad (9)$$

$$\cos \omega_n = \cos \langle \vec{V}_n, \vec{V}_d \rangle = \frac{\vec{V}_n \cdot \vec{V}_d}{|\vec{V}_n| |\vec{V}_d|}. \quad (10)$$

Among them, the angle calculation of  $\omega_c \equiv \cos^{-1}(\frac{\vec{V}_c \cdot \vec{V}_d}{|\vec{V}_c| |\vec{V}_d|})$  and  $\omega_n = \cos^{-1}(\frac{\vec{V}_n \cdot \vec{V}_d}{|\vec{V}_n| |\vec{V}_d|})$  is shown in Figure 4,  $\omega_c$  represents the angle formed by the moving speed of the node carrying the message and the target node, and  $\omega_{n1}$ ,  $\omega_{n2}$ , and  $\omega_{n3}$  represent the angle formed by the moving speed of the neighbor node to the target node, respectively.

Average speed indicates the ratio of the distance between the previous position of the node and the current position in time  $\Delta T$ . The previous position of node C is represented by coordinates  $(c_{0x}, c_{0y})$ , the current position is  $(c_x, c_y)$ , and the previous position of node N is  $(n_{0x}, n_{0y})$ , and the current position is  $(n_x, n_y)$ . The calculation process of the average speed of node C and node N is shown in formulas (11) and (12):

$$\bar{V}_c = \frac{\sqrt{(c_x - c_{0x})^2 + (c_y - c_{0y})^2}}{\Delta T}, \quad (11)$$

$$\bar{V}_n = \frac{\sqrt{(n_x - n_{0x})^2 + (n_y - n_{0y})^2}}{\Delta T}. \quad (12)$$

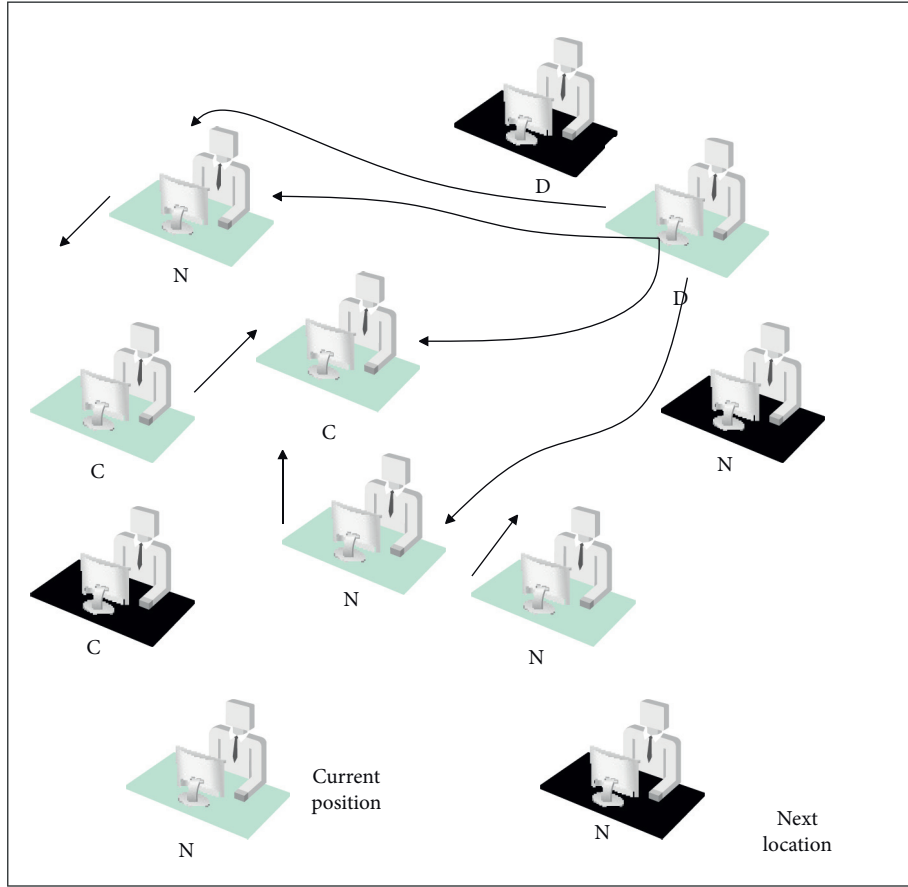


FIGURE 4: Calculation of the included angle of node moving speed.

Encounter distance indicates the distance between the node's moving position and the current position at the next moment in time  $\Delta T$ . The positions of node C, node N, and node D at the next time are  $(c1x, c1y)$ ,  $(n1x, n1y)$ , and  $(d1x, d1y)$ , respectively. Among them,  $(d1x, d1y)$  represents the sum of the distance from the current position and the distance moved at the next moment, which is calculated by formula  $(c_x + \Delta T \times \overline{V_{0x}}, c_y + \Delta T \times \overline{V_{0y}})$  and formula  $(n_x + \Delta T \times \overline{V_{1x}}, n_y + \Delta T \times \overline{V_{1y}})$ .

$$L_c = \sqrt{(d_{1y} - (c_y + \Delta T \times \overline{V_{0y}}))^2 + (d_{1x} - (c_x + \Delta T \times \overline{V_{0x}}))^2},$$

$$L_n = \sqrt{(d_{1y} - (n_y + \Delta T \times \overline{V_{1y}}))^2 + (d_{1x} - (n_x + \Delta T \times \overline{V_{1x}}))^2}.$$
(13)

The Merkle hash tree is a binary tree that maps a set of nodes to a set of fixed-size strings, as shown in Figure 5. Among them, each leaf node carries a given value, and the value of internal nodes including the root node is obtained by hashing of two child nodes, and a Merkle hash tree is constructed according to the data packets carried by the node and the distance of encounter.

Among them,  $\eta \in (0, \dots, 2^H - 1, H \geq 1)$  represents the index value stored by the leaf node, and  $H \geq 1$  represents the height of the Merkle tree. The hash value stored by the node

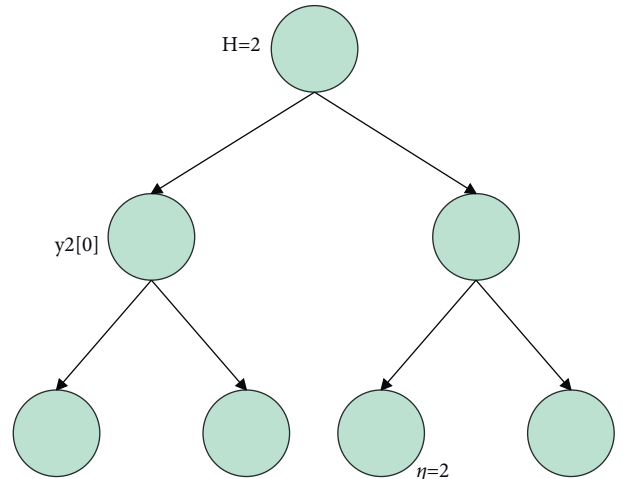


FIGURE 5: Example of Merkle hash tree.

is represented by  $y_h[i]$ , where  $h = 0, \dots, H$  represents the height of the node in the tree (the height of the leaf node is 0, and the height of the root node is  $H$ ), and  $i = 0, \dots, 2^{H-h} - 1$  is the position count of the node from left to right.  $f: (0, 1)^* \rightarrow (0, 1)^n$  is the hash function, and the hash value of the leaf node is calculated by the function  $f$ , where  $\otimes$  represents the connection operation, and the hash value of

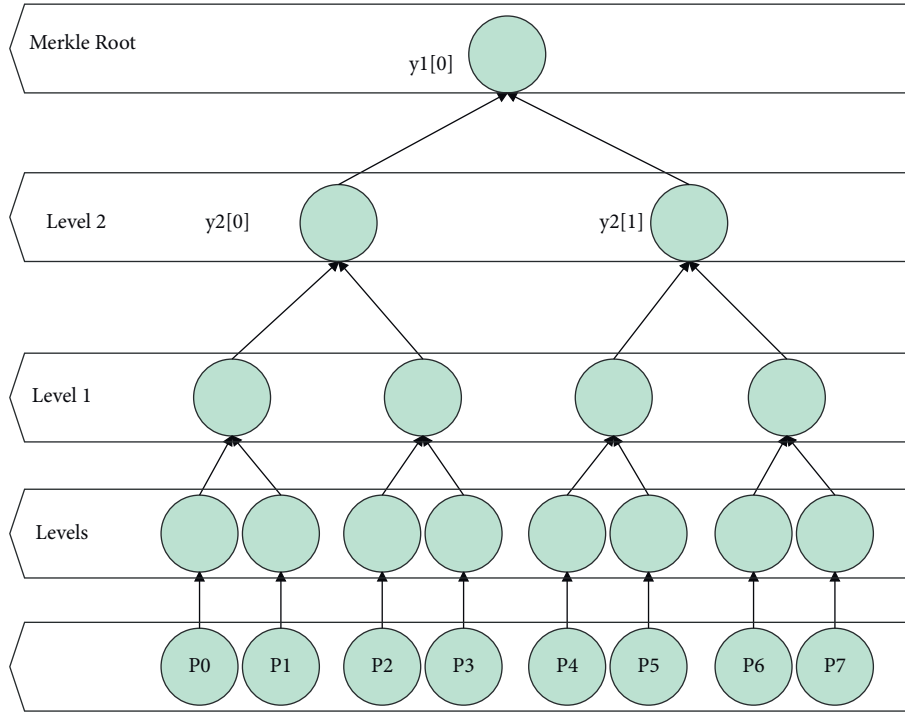


FIGURE 6: A Merkle tree containing 8 messages.

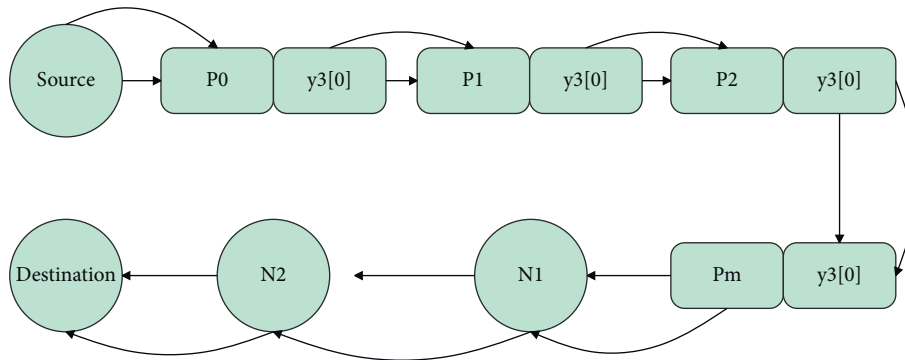


FIGURE 7: The embedding root node to the forwarding node.

the nonleaf node in the Merkle tree is calculated as shown in the following formula:

$$y_{h+1}[i] = f(y_h[2i] \otimes y_h[2i + 1]). \quad (14)$$

We assume that the source node and the target node are legitimate nodes, and the relay node is selfish. The Merkle hash tree creates the first level of the tree by hashing the number of messages, and the value of the nodes hashed in the first level becomes the leaf hash. After that, every two leaf hashes are paired and hashed to create a new hash value, until a hash value is finally created, and this node becomes the Merkle root node ( $y_3[0]$ ), as shown in Figure 6.

In the opportunistic social network, the Merkle hash tree will be used to check whether there is a message loss during the forwarding process between the source node and the

target node. If a message is removed, its parent node will change. Once the parent node changes, it will cause the root hash value of the entire Merkle tree to change. The source node embeds the Merkle root in the head of each message and sends it to the target node through the intermediate node, as shown in Figure 7.

The algorithm in this paper is compared with the traditional Epidemic and Prophet algorithms in terms of message delivery rate, average delay, and routing cost ratio. The definitions of the three indicators are introduced below.

**3.1. Message Delivery Rate.** The message delivery rate refers to the proportion of the total number of messages that can successfully reach the target node to the total number of messages sent by the source node. The calculation formula is

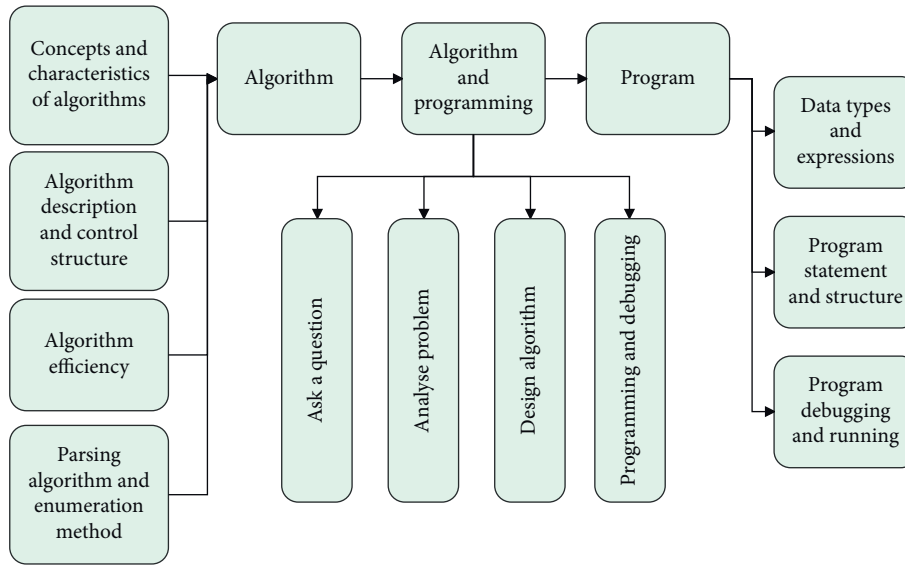


FIGURE 8: The relationship between the core concepts and key capabilities of algorithms and programs.

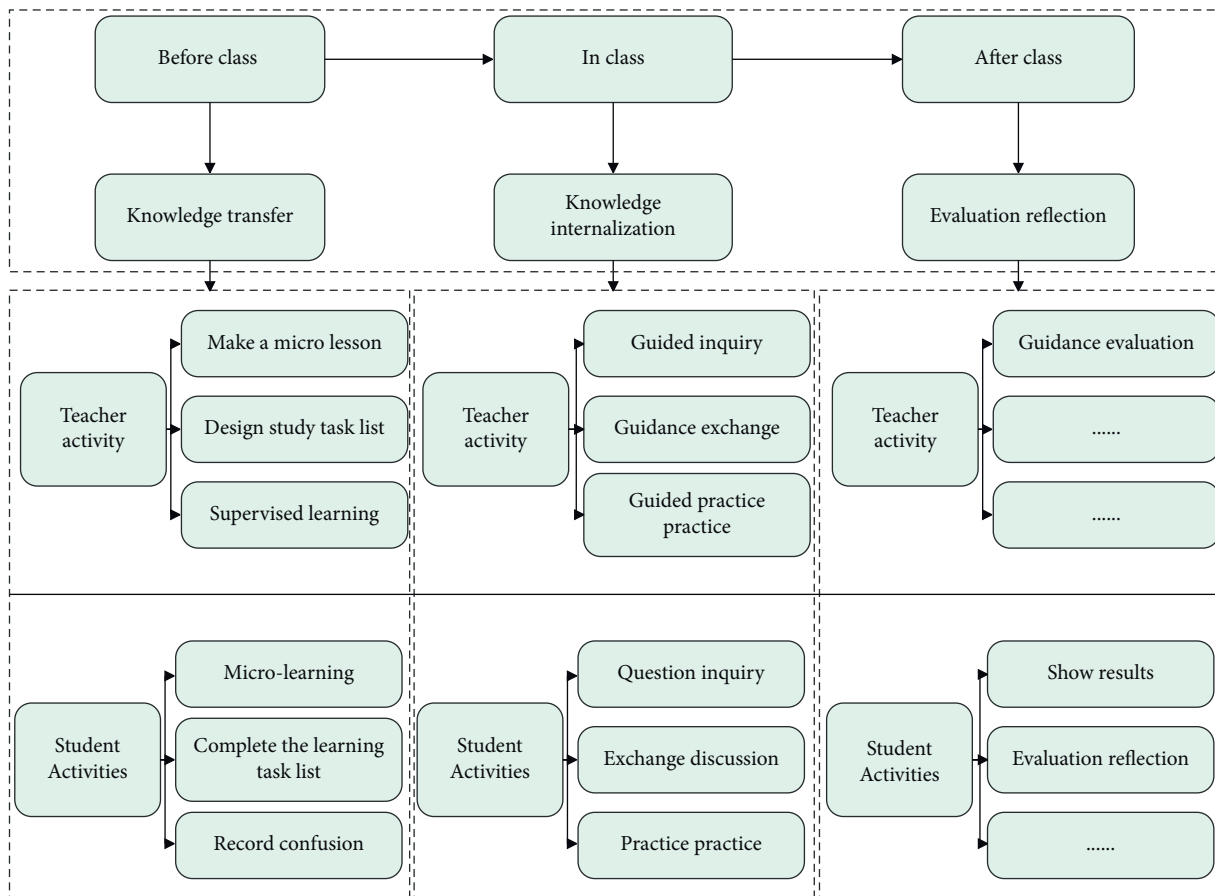


FIGURE 9: Flipped classroom teaching process design.



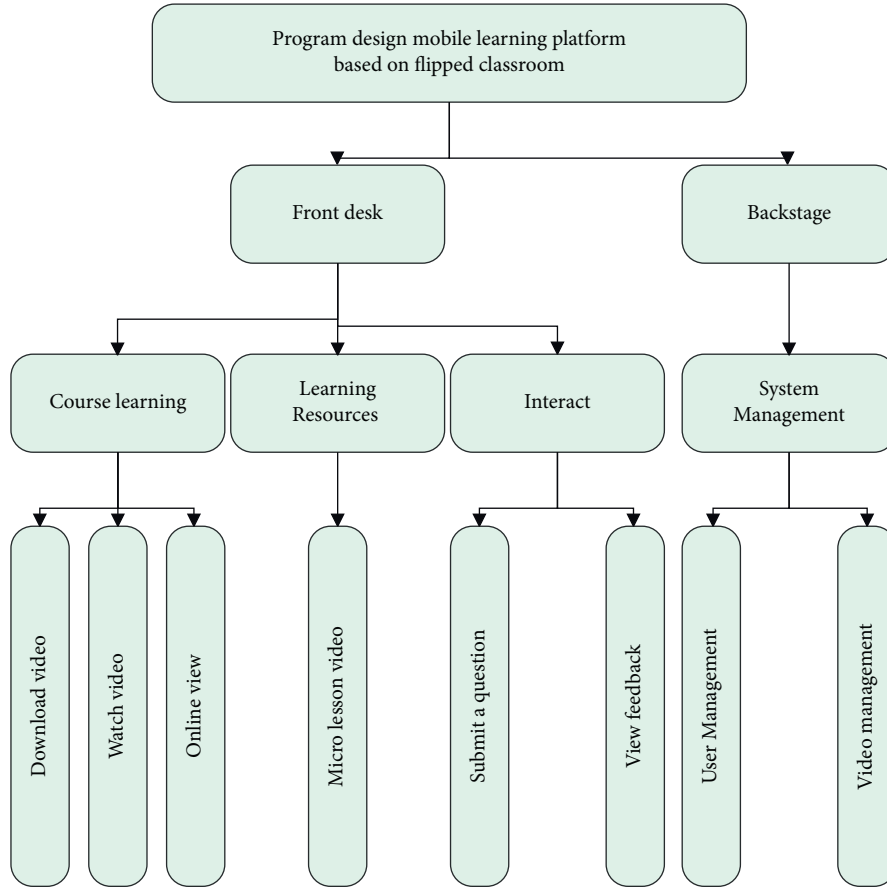


FIGURE 10: The overall structure design of the system.

$$D_{ratio} = \frac{R}{S}. \quad (15)$$

In this formula,  $R$  represents the total number of messages successfully sent to the target node, and  $S$  represents the number of messages that the source node has sent to the target node.

**3.2. Average Delay.** The average delay is the ratio of the total time of messages sent by the node to the total number of successfully reaching the target node. The calculation formula is

$$\bar{T} = \sum_{i=1}^n \frac{T_i}{R}. \quad (16)$$

Among them,  $T_i$  represents the message transmission delay for the  $i$ -th node to send a message to the target node within time  $T$ , and  $R$  is the total number of messages successfully reaching the target node.

**3.3. Routing Cost Ratio.** It represents the ratio of the total number of messages that the node fails to reach the destination to the total number of messages that successfully reach the destination under the conditions of the experiment. The calculation formula is

$$R_{ratio} = \frac{(\sum_{j=1}^n S_j - R)}{R}. \quad (17)$$

Among them,  $R_{ratio}$  represents the routing cost ratio of the message forwarding process, and  $\sum_{j=1}^n S_j$  represents the total number of messages sent by the source node to the destination node.

#### 4. Optimization of the Flipped Classroom Teaching Mode of Programming Courses Based on Social Cognitive Network

In this paper, the optimization analysis of the flipped classroom teaching mode of programming courses is carried out by transferring part of the cognitive network. In the teaching process, students should be guided to understand the relationship between the core concepts and key capabilities of this chapter. The relationship between the core concepts and key capabilities of “Algorithm and Program” is shown in Figure 8.

On the basis of sorting out and analyzing the research of the flipped classroom model, according to the particularity of the information technology curriculum and the specific situation of the school, this paper designs a three-stage flipped classroom model process including “before class, knowledge transfer,” “in class, knowledge

TABLE 1: Statistical table of evaluation of the teaching effect of flipped classroom of programming courses based on social cognitive network.

NO	Teaching evaluation
1	86.31
2	91.19
3	87.96
4	81.39
5	90.09
6	86.07
7	87.86
8	92.90
9	91.19
10	86.07
11	89.99
12	86.46
13	89.15
14	92.34
15	86.93
16	84.47
17	91.50
18	90.81
19	85.67
20	91.59
21	92.50
22	80.55
23	91.43
24	85.89
25	91.44
26	84.80
27	91.08
28	92.48
29	83.42
30	92.16
31	91.93
32	79.54
33	88.35
34	90.97
35	80.82
36	84.79
37	82.76
38	90.38
39	84.02
40	81.30
41	83.36
42	83.41
43	85.95
44	84.66
45	91.41
46	84.44
47	83.43
48	82.01
49	86.77
50	92.19
51	87.66
52	84.42
53	81.96
54	86.71
55	92.62
56	81.19
57	86.95
58	81.30

TABLE 1: Continued.

NO	Teaching evaluation
59	84.46
60	91.36
61	80.06
62	79.51
63	86.89
64	82.73
65	92.24
66	87.49
67	90.44
68	85.17
69	86.57
70	82.21
71	83.70
72	84.11
73	85.45
74	89.65
75	92.97
76	82.16
77	85.11
78	81.76
79	82.71
80	89.65

internalization,” and “after class, evaluation and reflection.” The tasks in the preclass stage include the following: teachers make microclasses, design self-learning task lists, and supervise learning, and students watch microclasses to self-learn, complete self-learning task lists, and record confusion. The tasks of the in-class stage are as follows: teachers guide inquiry, guide communication, guide practice, and practice, and students carry out problem exploration, exchange and discuss, and implement practice. The tasks in the after-school phase are as follows: teachers perform guidance and evaluation, and students show results and perform evaluation and reflection. The design of the flipped classroom teaching process is shown in Figure 9.

The mobile learning platform is developed based on the mobile terminal platform and is mainly used in mobile devices, such as smart phones and tablet computers, including an operating system, middleware, and some key platform applications. Moreover, it can be ported to different hardware platforms. The system consists of two parts: the foreground and the background. The front desk functions include user login, user registration, interactive communication module, course learning module, and learning resource module. The background is the system management module, which is mainly operated by the administrator, including student management and resource management. Figure 10 shows the overall structure of the mobile learning platform.

After constructing the above system, the performance of this system is verified through experimental teaching methods, and the teaching effect is counted. The results obtained are shown in Table 1 and Figure 11 below.

From the above research, we can see that the flipped classroom teaching system of programming courses based

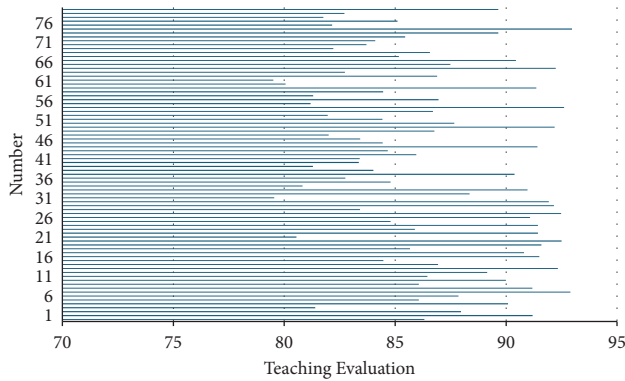


FIGURE 11: Statistical diagram of evaluation of the teaching effect of flipped classroom of programming courses based on social cognitive network.

on social cognitive network constructed in this paper has good teaching effects and can effectively improve the teaching quality of programming courses.

## 5. Conclusion

The emergence of the flipped classroom has provided a new direction and injected fresh vitality for the reform of education and teaching. Based on the relevant theories and practices of microclasses and flipped classrooms, this paper creatively combines microclasses, flipped classrooms, and high school algorithms with programming teaching based on the actual teaching work. Moreover, this paper combines social cognitive network and wireless information dissemination technology to design a three-stage model of flipped classroom teaching based on microclasses and apply it to the teaching practice of algorithm and programming. In addition, this paper analyzes the optimization of the flipped classroom teaching mode of programming courses through the transfer of part of the cognitive network. Finally, this paper designs an experiment to verify the performance of the system constructed in this paper. The research results show the effectiveness of flipped classroom based on social cognitive network in algorithm and programming teaching.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

The research was supported by Railway Police College university-level curriculum ideological and political construction project (Communication and Expression SZ202005) and the Railway Police College university-level first-class undergraduate curriculum construction project in 2020 (Communication and Expression YLKC2011).

## References

- [1] L. Zhou, H. Li, and K. Sun, "Teaching performance evaluation by means of a hierarchical multifactorial evaluation model based on type-2 fuzzy sets," *Applied Intelligence*, vol. 46, no. 1, pp. 1–11, 2016.
- [2] J. Porozovs, L. Liepniece, and D. Voita, "Evaluation of the teaching methods used in secondary school biology lessons," *Journal of Pedagogy and Psychology "Signum Temporis"*, vol. 7, no. 1, pp. 60–66, 2015.
- [3] M. A. Oliveros, A. García, and B. Valdez, "Evaluation of a teaching sequence regarding science, technology and society values in higher education," *Creative Education*, vol. 6, no. 16, pp. 1768–1775, 2015.
- [4] M. Sánchez Cerón and F. M. D. S. Corte Cruz, "The evaluation of teaching: some consequences for Latin America," *Revista Mexicana de Investigación Educativa*, vol. 20, no. 67, pp. 1233–1253, 2015.
- [5] A. Li, K. Li, K. Li, and Z. Ge, "Application of data mining in the colleges' in-class teaching quality evaluation system," *Journal of Computers*, vol. 10, no. 3, pp. 166–175, 2015.
- [6] K. Angell and E. Tewell, "Teaching and un-teaching source evaluation: questioning authority in information literacy instruction," *CommInfoLit*, vol. 11, no. 1, pp. 95–121, 2017.
- [7] M. Brkovic and P. Chiles, "'Spector - the sustainability inspector': participatory teaching, learning and evaluation game for architects, architecture students and pupils," *Facta Universitatis - Series: Architecture and Civil Engineering*, vol. 14, no. 1, pp. 1–20, 2016.
- [8] T. H. Reisenwitz, "Student evaluation of teaching: an investigation of nonresponse bias in an online context," *Journal of Marketing Education*, vol. 38, no. 4, pp. 139–144, 2015.
- [9] Y. Jiang and Y. Wang, "Evaluation of teaching quality of public physical education in colleges based on the fuzzy evaluation theory," *Journal of Computational and Theoretical Nanoscience*, vol. 13, no. 12, pp. 9848–9851, 2016.
- [10] K. Royal, "A guide for making valid interpretations of student evaluation of teaching (SET) results," *Journal of Veterinary Medical Education*, vol. 44, no. 2, pp. 316–322, 2016.
- [11] F. Garofalo, P. Mota-Moya, A. Munday, and S. Romy, "Total extraperitoneal hernia repair: residency teaching program and outcome evaluation[J]," *World Journal of Surgery*, vol. 41, no. 1, pp. 1–6, 2017.
- [12] G. Gong and S. Liu, "Consideration of evaluation of teaching at colleges," *Open Journal of Social Sciences*, vol. 4, no. 7, pp. 82–84, 2016.
- [13] U. Klusmann, D. Richter, and O. Lüdtke, "Teachers' emotional exhaustion is negatively related to students' achievement: evidence from a large-scale assessment study," *Journal of Educational Psychology*, vol. 108, no. 8, pp. 1193–1203, 2016.
- [14] H. W. Marsh, A. S. Abduljabbar, P. D. Parker et al., "The internal/external frame of reference model of self-concept and achievement relations," *American Educational Research Journal*, vol. 52, no. 1, pp. 168–202, 2015.
- [15] M. Pinxten, C. Van Soom, C. Peeters, T. De Laet, and G. Langie, "At-risk at the gate: prediction of study success of first-year science and engineering students in an open-admission university in Flanders-any incremental validity of study strategies?" *European Journal of Psychology of Education*, vol. 34, no. 1, pp. 45–66, 2019.
- [16] M. Brehm, S. A. Imberman, and M. F. Lovenheim, "Achievement effects of individual performance incentives in a teacher merit pay tournament," *Labour Economics*, vol. 44, no. 5, pp. 133–150, 2017.

- [17] R. J. McGill and A. R. Spurgin, "Assessing the incremental value OF kabc-II luria model scores IN predicting achievement: what do they tell US beyond the mpi?" *Psychology in the Schools*, vol. 53, no. 7, pp. 677–689, 2016.
- [18] T. Mohoric and V. Taksic, "Emotional understanding as a predictor of socio-emotional functioning and school achievement in adolescence," *Psihologija*, vol. 49, no. 4, pp. 357–374, 2016.
- [19] D. L. Rabiner, J. Godwin, and K. A. Dodge, "Predicting academic achievement and attainment: the contribution of early academic skills, attention difficulties, and social competence," *School Psychology Review*, vol. 45, no. 2, pp. 250–267, 2016.
- [20] C. W. Coley, R. Barzilay, T. S. Jaakkola, W. H. Green, and K. F. Jensen, "Prediction of organic reaction outcomes using machine learning," *ACS Central Science*, vol. 3, no. 5, pp. 434–443, 2017.
- [21] A. Chowdhury, E. Kautz, B. Yener, and D. Lewis, "Image driven machine learning methods for microstructure recognition," *Computational Materials Science*, vol. 123, no. 8, pp. 176–187, 2016.
- [22] S. Basith, B. Manavalan, T. H. Shin, and G. Lee, "SDM6A: a web-based integrative machine-learning framework for predicting 6mA sites in the rice genome," *Molecular Therapy - Nucleic Acids*, vol. 18, no. 6, pp. 131–141, 2019.
- [23] C. Voyant, G. Notton, S. Kalogirou et al., "Machine learning methods for solar radiation forecasting: a review," *Renewable Energy*, vol. 105, no. 2, pp. 569–582, 2017.