

## Research Article

# Topology Optimization of Interactive Visual Communication Networks Based on the Non-Line-of-Sight Congestion Control Algorithm

Boya Liu <sup>1</sup> and Xiaobo Zhou<sup>2</sup>

<sup>1</sup>Department of Journalism and Media, Chongqing Normal University, Chongqing 401331, China

<sup>2</sup>Film and Television Animation Department, Sichuan Fine Arts Institute, Chongqing 401331, China

Correspondence should be addressed to Boya Liu; 20132044@cqnu.edu.cn

Received 19 October 2020; Revised 9 December 2020; Accepted 14 December 2020; Published 24 December 2020

Academic Editor: Wei Wang

Copyright © 2020 Boya Liu and Xiaobo Zhou. 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 this paper, an in-depth study of interactive visual communication of network topology through non-line-of-sight congestion control algorithms is conducted to address the real-time routing problem of adapting to dynamic topologies, and a delay-constrained stochastic routing algorithm is proposed to enable packets to reach GB within the delay threshold in the absence of end-to-end delay information while improving network throughput and reducing network resource consumption. The algorithm requires each sending node to select an available relay set based on the location of its neighbor nodes and channel state and computes transfer probabilities for each node in the relay set combining the remaining delay of the packet with the distance from the relay node to GB. Based on the obtained transfer probability and local channel state, the sending node passes the packet to the relay node. The convergence of the algorithm is proved and its performance is verified by simulation. The first part of the algorithm is based on the greedy algorithm to deploy and locate the network flying platform nodes with the goal of efficient coverage of the network flying platform nodes, considering the ground base station services. As the delay on each link varies due to the change of channel state, the source and relay nodes asynchronously update the data generation rate and the pairwise parameters based on the received local information and use the obtained optimal values to pass the packets to GB.

## 1. Introduction

Vision is an important sense for us human beings to distinguish the big and small, light, and dark, as well as movement and stillness and colors of the external world. At least 80% of the information about the external world is obtained and felt through vision and finally processed by the brain and perceived by humans [1]. The process of human beings converting the information obtained through vision into graphic images can be regarded as a simple visualization process [2]. The earliest “visualization technology” is mainly used in computer science and applications and formed a very important branch—scientific computing visualization. But, with the development of time and technology, visualization technology is no longer confined to scientific computing; it has been extended to all aspects of life, such as

the history of Egyptian hieroglyphics, cave paintings, science education, interactive multimedia, engineering, medicine, business, and virtual communities [3]. In recent years, with the growth of the Internet, an increased number of people are spending part of their daily lives on various projects such as social software, online shopping, and outings [4]. Nowadays, our society is already an era of rapid development, with the rapid development of technology, efficient flow of information, more frequent communication, and more convenient lifestyles; all kinds of activities have produced a series of data [5]. Therefore, this era not only creates a lot of big data companies but also creates the “big data” hot word, more importantly the generation of big data to promote the vigorous development of big data industry and technology, including visualization technology. In this era of big data, network topology visualization technology has become an

important branch of the study of visualization technology; network topology visualization is usually the use of points, lines, graphics, and other elements to form a basic image to represent the mobile network, backbone network, or core network topology and information display [6]. Usually, network topology visualization can clearly and intuitively reflect the current operation of the network to assist network managers for network links, nodes, and other aspects of a comprehensive assessment, prediction, analysis, a clear understanding, and understanding of the network internal specific information, laws, and changes [7]. At present, many research institutions related to computer graphics have shifted the focus of their work to the study of network topology visualization tools; visualization technology as a key technology to show network topology and its related products and tools are also increasingly useful [8].

Network topology is the arrangement of various elements (links, nodes, etc.) of a communication network. A network topology is the topology of a network and can be physically or logically described [9]. It is the application of graph theory, where communication devices are modeled as nodes and connections between devices are modeled as links or lines between nodes [10]. The physical topology is the layout of the various components of the network (e.g., device locations and cable installations), while the logical topology describes the way data flows through the network [11]. The distance between nodes, physical interconnections, transmission rates, or signal types may be different in two different networks, but the topology may be the same [12]. The physical topology of the network is of interest to the physical layer of an OSI model. Network topologies can be used to define or describe the layout of various types of telecommunications networks, including command and control radio networks, industrial fieldbuses, and computer networks [13]. Graphical diversity of data can be obtained by working with organizations such as education, research, government, and business partners to obtain different types of big data from various network links and then visual collection tools and data analysis tools are used to better analyze network topology, route addressing, domain security, traffic analysis, performance, policy, and visualization. The goal is to assist in building a scalable, robust Internet data analysis tool and analytics approach. Vizster is used for interesting end-user exploration and navigation of large online social networks. Vizster is based on a familiar node-linked network layout [14]. The system is based on a familiar node-linked network layout, providing custom techniques for exploring connectivity in large graphical structures, supporting visual search and analysis as well as automatic identification and visualization of community structures [15]. The goal of the commission is to focus on the development of self-regulating and self-healing mechanisms that are, on the one hand, decentralized, scalable, and adaptable to changes in their environment [16]. On the other hand, these decentralized mechanisms lead to globally acceptable behavior and avoid nonconformity or instability [17]. The tools developed in this project are mainly applied in the visualization of network topology at the autonomous systems (AS) layer.

The above research and visualization tools aim at a comprehensive, detailed, and accurate presentation of network information and all of them use visual layout algorithms. To represent the information between nodes and links in detail, these systems use more complex icons in rendering topology diagrams and enhance the user's perception of network information through a variety of effects such as animation and interaction. Along with the rapid development of the network, there are numerous frameworks and tools for obtaining network information and displaying the network structure, but they cannot completely display the various requirements in terms of supporting and displaying different topology types. Based on this, this paper builds a real-time interactive network topology visualization system through Web technology to display various data distributions more clearly and effectively, provide interactive operation, and obtain more potential network information. Design and implement a network topology visualization system to display the complex nodes and the connection information between the nodes in a visualized form in the page, and provide users with a variety of dimensional data display methods, so that users can filter, move, and modify attributes and other interactive ways to analyze and monitor the network data in the visualized topology.

## 2. Interactive Visual Communication Network Topology Optimization Design for Non-Line-of-Sight Distance Congestion Control Algorithm

*2.1. Nonvisual Distance Congestion Control Algorithm.* In this paper, link quality is defined as the probability that a receiver will correctly decode a packet, and at any given moment each link will receive an interference signal from other transmissions in the same network. In this paper, the link quality is defined as the probability that the receiver will correctly decode the packet; at any given moment each link will receive the same network from other transmissions. The cumulative interference value at time  $t$  (known as the instantaneous interference value) does not accurately reflect the current channel state [18]. Also, the fast mobility makes it very difficult for the UAV to obtain the state information of all channels, and the instantaneous interference value can only reflect the current channel state, so it is necessary to find a method that can meet the changes of distance between nodes and accurately reflect the current channel state. The interference prediction method satisfies the requirement well by calculating the average interference value in time  $t$  using the integration idea based on the movement model of the nodes.

$$E[I_j(t|t_0)] = \sum_{i,j} \frac{1}{\Delta t} \int_{t_0}^t g_{ik}(y^i(t)) \cdot f_y(y^j(t)) dt. \quad (1)$$

Using formula (1), the signal-to-interference-plus-noise ratio (SINR) at node  $j$  can be obtained.

$$\kappa_j^i(t|t_0) = \frac{\sum_{i,j} 1/\Delta t \int_{I_0}^I g_{ik}^{\Delta t}(y^i(t)) \cdot f_y(y^j(t)) dt}{N_0 + E[I_j(t|t_0)]}, \quad (2)$$

where  $N$  denotes the variance of ambient noise and it is generally set as a constant. According to formula (2), the average link capacity within  $t$  interval can be obtained.

$$C_{ij} = W \log(20 + \kappa_j^i(t|t_0)), \quad (3)$$

$$P_{ij}^{\text{ON1}} = 1 - e^{(-5^{ij} - 1/s^i)}. \quad (4)$$

The size of each packet is assumed to obey an exponential distribution with a mean value of  $K$  bits. Each node maintains a single queue and passes the packet to the relay node on a first-come-first-serve basis. According to the conclusions obtained in [19, 20], when the arrival process obeys a Poisson distribution and the service process obeys an independent exponential distribution, the average delay of the link  $i, j$  at time  $t$  can be approximated as follows:

$$M_{ij} = \frac{24\kappa}{C_{ij} - r_{ij}}. \quad (5)$$

If the following formulas (6) and (7) are established, the start receiving operation receives the partition information from the main process and initializes the subarea flow field.

$$D_1 = w_{11}x_1 + 2w_{12}x_2 + 4w_{13}x_3, \quad (6)$$

$$D_2 = w_{21}x_1 + 2w_{22}x_2 + 4w_{23}x_3. \quad (7)$$

Continue the unfinished calculation, start the receiving operation flow field, and get the following formula:

$$V(i, j) = \sum_{m,n} I(i+m-2, j+n+1) \cdot K(m-1, n+1). \quad (8)$$

The weighted average of the final aggregation model can be written as

$$G(t+5) = g(t+1) + \lambda g(t+2). \quad (9)$$

The idea behind the non-line-of-sight congestion control algorithm is top-down hierarchical placement. The algorithm points out that the dependency between child nodes and parent nodes is reduced, and the placement of any node is strictly following the grid method of breadth-priority traversal from left to right, from top to bottom, and one by one [21]. This method is easy to implement, highly efficient solution to the crowded situation of network. Because it is a grid, the spacing between each of the nodes is fixed, and there will be no overlapping nodes. This relieves the pressure of space allocation to a certain extent. However, the grid layout algorithm also has shortcomings; using the grid layout algorithm to draw a visual graph, the connection between the parent and child nodes becomes unintuitive, which will seriously affect the user's perception of the network data, as shown in Figure 1.

The term network hierarchy is usually used to refer to the main network and the subnet, where the main network shows the connections between the main nodes and the

subnet shows the connections between the nodes and the nodes within a given range. Because of the huge amount of data in today's networks, it would be very crowded to show all the network nodes in one window. In this case, we only need to click on one of the expert nodes to get a detailed overview of the connections in the network. The network nodes are shown in a total score. The advantage of the hierarchical layout algorithm is that it can roughly visualize large network data in a window, and when the user wants to see a detailed network structure, the window can be shown in detail; because of this advantage, the hierarchical layout algorithm is used in a lot of visual layouts. But the algorithm also has shortcomings; for example, when the number of nodes is too many, how to divide the subnet becomes a severe problem.

It has been praised for its clear display of the correlation between nodes, which is based on the principle that each node will have repulsion and gravity, there is gravity between neighboring nodes, there is repulsion between all nodes, and once the gravity between two nodes is greater than the repulsion, the position of these two nodes will be regulated very close; once the repulsion between two nodes is greater than the gravity, the position of these two nodes will be regulated very far away [22]. In other words, the closeness of the relationship between nodes can be seen by the distance between the nodes; too far away indicates that the two nodes are distantly related, and closer distance indicates that the two nodes are closely related. Also, because of the repulsion, a certain distance is created between the nodes, and the effect of overlap between the nodes is reduced. The force guide mainly contains charge repulsion, gravity, and friction. Topological diagram between the nodes of charge is expressed as mutual repulsion to ensure that nodes do not overlap; gravity is expressed as nodes farther away from the center of gravity; gravity is greater to complete the relative layout of the topological map and to ensure that the nodes are relatively compact. Friction is represented by the energy decay rate; the stronger the frictional force, the faster the stopping speed of the guide map.

*2.2. Interactive Visual Communication Network Topology Optimization Design.* Interactive visual communication network topology optimization is a framework that separates and treats the input, processing, and output modules of an application separately. The logical relationship between the three is as follows: the topology sends a request to the controller, the controller selects the appropriate model for processing, the model feedback information is sent to the controller, and the controller is to select the appropriate topology, thus generating the corresponding view to the user [23]. Model to provide the topology to show the content; at the same time it can realize a model to create a new topology without rewriting the model; as long as the data in the model changes, the topology will receive notification of the model, and the corresponding topology will be rerendered page. The model can be reused, the model topology and controller are independent of each other, and one or more of them can be ported to the new working platform separately, thus greatly

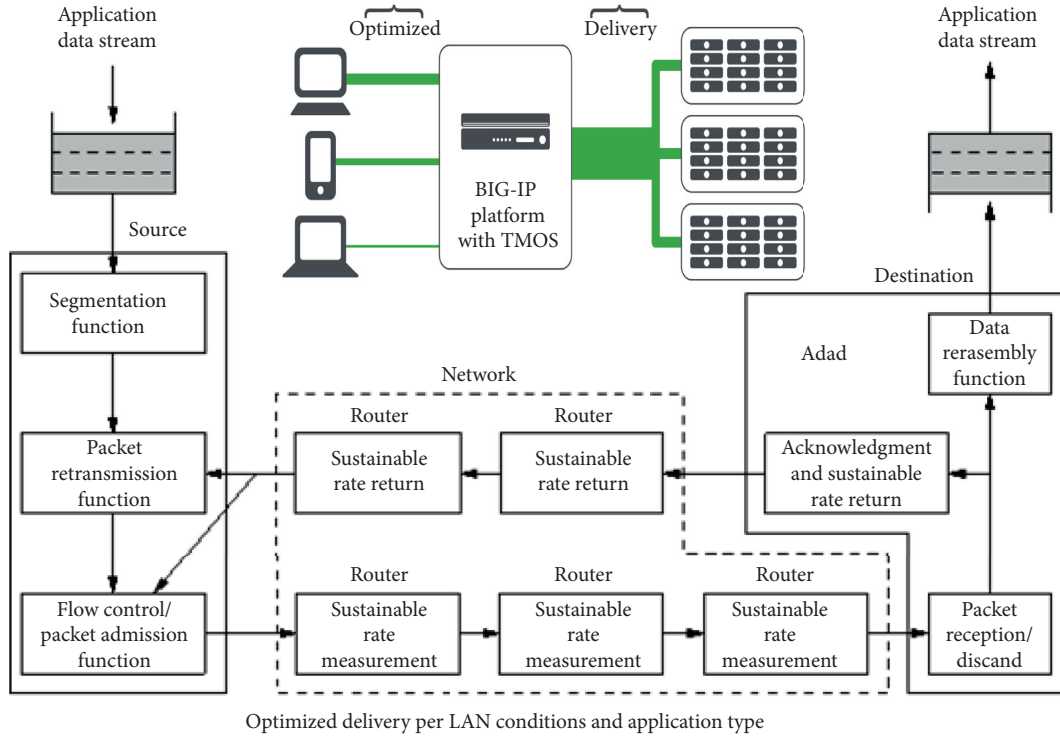


FIGURE 1: Framework of the nonvisual congestion control algorithm.

increasing the efficiency of development. The job of the topology is to think about how to present the data logically in the interface. The model only needs to think about how to maintain the data and handle the business logic. The division of labor between the three greatly increases the efficiency of research and development [24]. In this system, the corresponding model contains node information, business information, protocol information, and relationship data between the nodes and provides the corresponding data addition, deletion, and correction operations. The topology diagram is displayed in the client browser. In this system, the topology map corresponds to the related controls, which are used for data acquisition, component creation, data binding, rendering control, and so forth. The results of these operations are finally presented on the visualization page, as shown in Figure 2.

When the data is modified, for example, deleted or updated, if the data is also in the cache, then the data in the cache is in an invalid state; for example, there is data in the cache, and the front end deletes the data; there is still data in the cache; if the front end requests the data, the data can be read from the cache, and this is not consistent with the facts [25]. Therefore, after the front-end modifies the data, it sends a request to the server, and if the request succeeds, it needs to set the corresponding data in the cache to expire, and when it requests the data from the server again, the server queries the database and returns the found data to the front end and updates the data in the cache. The reason for setting the original data in the cache to invalid if the database is modified successfully is that if the corresponding data in the cache is set to invalid and cannot be queried, a thread will

read the old data without modifying the database in a multithreaded environment, thus overwriting the modified cache data and causing the old data to be stored in the cache, resulting in the existence of dirty data in the cache. This is the reason for modifying the database first and then updating the cache. There are three key points in the rendering process: data initialization, data update, and data merge [26]. Data initialization will render the data existing in the database; when the user opens the topology visualization system page, the system may already be running, and the running state of the database accumulated perceived new data, after opening the page will enter the data update process; in between the data initialization and data update, there is a data merge process.

The business layer extracts data from the data layer and submits the resulting topology map to the UI layer. The business layer is the key module of the network topology visualization system, which is subdivided into a data simulation module, a data management module, a visual layout module, and an interaction control module. The data simulator can generate sample data required by the network topology visualization system to simulate realistic scenes, and the simulated data will be stored in Oracle, which will convert the data into JSON format through schema and then store it in MongoDB, which provides an add-delete-change-check interface to the server and front end. The data management module is to process the data deposited in the data simulator; the module includes topological data preprocessing; that is, the data will be mode conversion, real-time update; take out the data from MongoDB in line with the front-end request, but the data format does not meet the requirements, so you

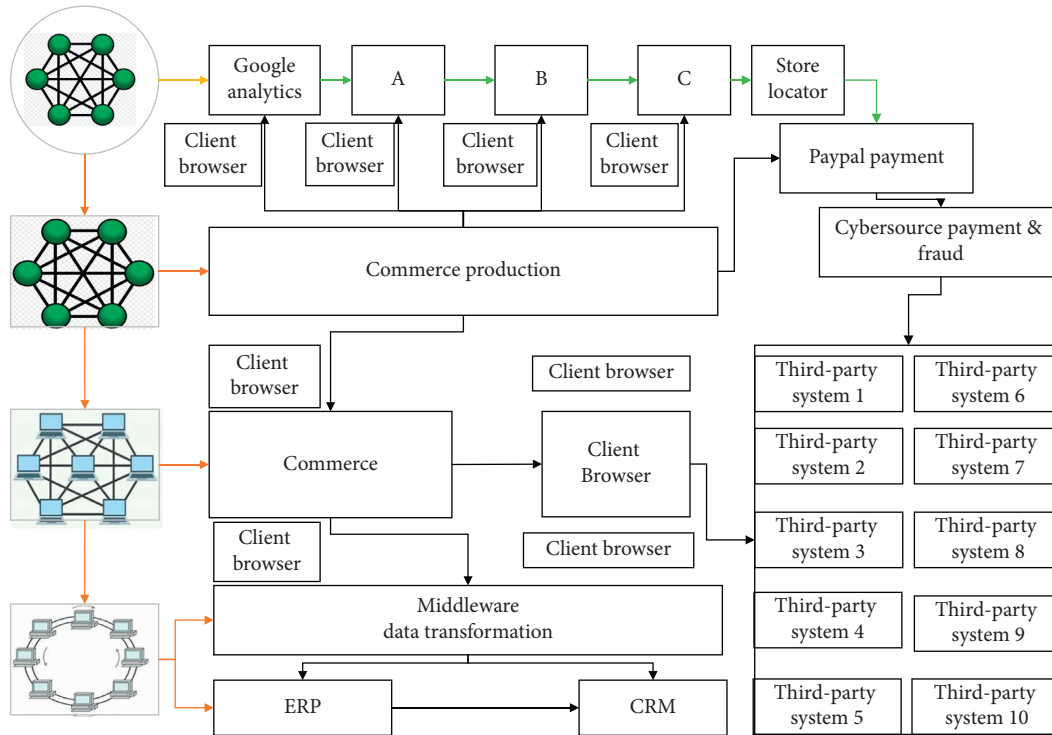


FIGURE 2: Interactive visual communication network topology optimization design.

need to preprocess the data, and the system has real-time update capabilities, the system is running, the front end via Ajax every 2 seconds to request data from the server; if the background data changes, the merged data will change the data update into the existing page; visual layout module is to calculate the location of the nodes in the data management module to determine the final location of the nodes in the layout and draw out the diagram; the module includes layout algorithm optimization, D3.js library optimization, and D3.js to draw the topology diagram. The topology generated by the visual layout module needs to provide interactive functions to the user. The interaction control module contains six parts: connection filtering, topology scaling control, page synchronization, attribute modification, topology subdiagram, and traffic view.

**2.3. Module Design Analysis.** Interaction means that when a user enters a certain kind of command, the system receives and responds accordingly. Interaction allows the topology diagram to express more content while allowing the user to learn more about the potential information. Commonly used interaction tools are the mouse and keyboard. The interaction control module is built based on the topology map drawn by the visual layout module, and the network topology visualization system provides interaction control function to the user, which makes the user obtain visualization information more intuitively, and at the same time, the content is selectively visible according to different visualization objects, and the information is parsed from all directions to achieve efficient information perception effect.

Connection filtering is one of the interactive functions of a network topology visualization system. The network topology visualization system is a platform to display the network and provides interactive functions to the user so that the user has more freedom in accessing information; the user can choose to view the information they are interested in. The connection filtering function allows the user to choose to see one or more types of information and hide the unselected information. When the user selects one or more connection labels, the user gets the target ID and finds the label that needs to be filtered to return to the filter set that needs to be filtered and hide the information in the filter set based on the filter set that is obtained to achieve the purpose of viewing only one or more types of information. After the filtering function is implemented, when the user clicks on the filter tag again, the filtered links will be re-rendered to the current page, as shown in Figure 3.

In the system operation, when the user wants to know the connection and situation between a specific node and related nodes, just drag and drop the local topology related to the node into the visible area of the page to improve the user interaction. When zooming the topology, the event listener first defines the scale factor range (0.01, 10) and the translation offset (480, 300); once the mouse is placed on the topology, the zoomed function is called via the scroll wheel. One of the features of the network topology visualization system is the centering and highlighting of nodes between multiple pages. The premise is that different pages are in different processes, and different processes are in different memory regions. To achieve page synchronization, it is necessary to push information to the front page through the server. The server to push information to the front page

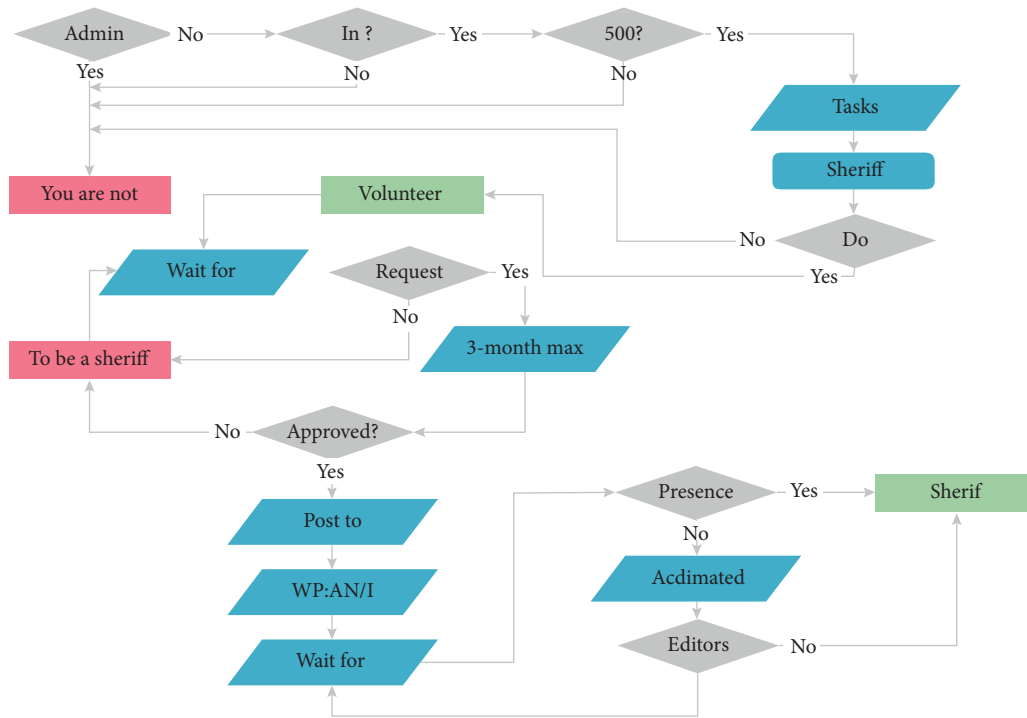


FIGURE 3: Link filter.

needs to pay attention to three aspects; the first push information cannot interfere with irrelevant instructions, for example, when a user sends a centered signal when another user is not able to receive the signal. The second user-related page response needs to be consistent, such as the user to open more than one page; once the user gives the page centered command, all open pages need to be centered at the same time. Third, when the user closes the page, the server automatically clears the state of the page synchronization maintained by the server to reduce memory pressure.

The user opens a page in the browser, the page will send a registration request to the server, registration information including the page and the command (because the command can be many kinds, e.g., this system has centered, highlighted, etc.). When the server receives the request, it first checks to see if this type of box has already been registered, and if the box has been registered, it simply places the information into the box; if there is no box, it creates the box first and then places the page and instructions into the box and returns the results to the page. The user issues a command on a page; the server receives the command to find the appropriate box and notifies the other pages in the box to execute the command. Once the user closes the page, the server finds the corresponding box and logs out the corresponding information in it or logs out the box if there are no pages in it. As the data obtained may deviate from the real information, the attribute modification function is introduced to make the attribute information consistent with the real information. If you click the “Confirm” button, Ajax will write the modified information back to the backend, and the coracle will write the data back to Oracle. At the same time, the corresponding information in the cache will be set to expire, and the attribute information will be changed; if

you click the “Cancel” button, the window will be closed automatically, and the attribute information will not be changed. When the attribute is changed, you need to change all the information in Oracle and the cache, because if only the data in Oracle is changed, when the front end requests the information, it will first fetch the data from the cache, and the data that is fetched is the unchanged data, and then the dirty data situation will occur. If only the corresponding data in the cache is set to be invalid, when the incremental update is done, the server Tornado will request data from Oracle because the data stored in Oracle has not been changed, resulting in fetching the dirty data and changing the data in the cache to dirty data, which are not allowed by the program. Therefore, after property modification, you need to change the data in Oracle and cache at the same time.

### 3. Analysis of Results

**3.1. Simulation Results Analysis.** By considering the performance of the algorithm under different network parameters (mainly including the number of nodes, speed, and hello interval), the simulation results are collected and Figure 4 is generated. The delay-constrained random routing algorithm proposed in this chapter is abbreviated as DSRA. To simplify the transmission process, each relay node has only one chance to retransmit the packet. The simulation results are divided into two parts: Figure 4 shows the trend of network parameters for the proposed routing strategies in this section, including DSRA, OR-DSP, P-OLSR, and GRAA, for different travel speeds and hello intervals. OR-DSP mainly uses Dijkstra’s shortest path and the expected locations of intermediate nodes to design the routing algorithm. P-OLSR incorporates the relative speed and link

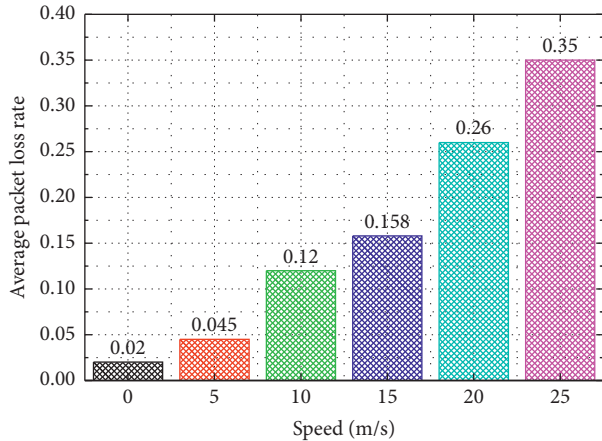


FIGURE 4: Total packet loss rate at different speeds.

state between two nodes into route selection, and the routing process used by GRAA improves the performance of location-based information routing algorithms by using time-based node movement prediction and the idea of designing routes for DTNs. The results of evaluating different network parameters (including timeout rate, packet loss rate, and throughput) at different speeds and the number of nodes are shown in Figure 4. From Figure 4, the total packet loss rate increases with increasing speed. Two main factors determine the total packet loss rate: link quality and latency constraints. The former factor leads to more retransmissions, which not only increases the end-to-end delay but also makes it more likely that the total delay consumed by the packet will exceed the delay threshold. The latter factor increases the packet loss rate because the relay node will drop the packet when the delay consumed by the transmission is greater than the given threshold.

DSRA collects information only from one-hop neighbors and uses this information to make routing decisions. The time interval between each pair of nodes exchanging beacons severely affects the reliability of single-hop transmissions, where the sending node's use of previously saved channel states leads to poorer routing decisions by the nodes and thus degrades the network performance. The channel state computed in equations (3)–(5) is closely related to the distance between the two nodes, and the increase in hello spacing degrades the network performance when the movement speed of the node changes. Therefore, it can be seen from Figure 4 that the channel state information used by a relay node to select a relay node using the previously stored channel state does not reflect the current link quality, which results in the node selecting a route to deliver packets, thus reducing the number of packets correctly received by GB. In practical implementation, a suitable hello interval needs to be set to meet the requirements of a node's single-hop transmission success rate at a given movement speed. Figure 5 shows the relationship between the hello interval and the node's movement speed when the average transmission success rate reaches 80%. At lower movement speeds, the coordinates of each node do not change much and the channel state in the network gradually becomes

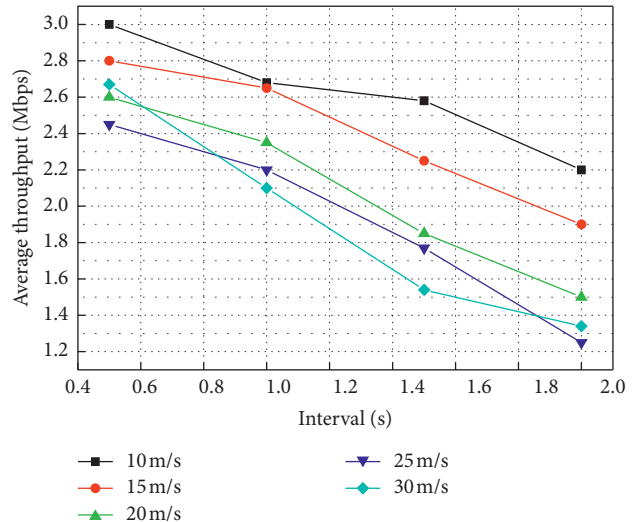


FIGURE 5: Average network throughput with speed for different hello intervals.

more stable. The transmitting node only uses its stored channel state to select a relay node and does not exchange beacons to obtain the current channel state; choosing an appropriate hello interval enables single-hop transmission to achieve network performance like that using the current channel state.

Next, the performance of this chapter's approach with OR-DSP, P-OLSR, and GRAA under different parameters is analyzed. Delay constraints are an important factor considered by DSRA, so the optimization iteration of each relay node considers not only different network metrics (e.g., link quality) but also delay constraints. As with other routing algorithms (P-OLSR and GRAA), although OR-DSP merges the wait time of intermediate nodes into the path selection to reduce end-to-end delay, it does not explicitly consider end-to-end delay constraints, and additionally reducing the one-hop delay does not mean that the total delay can meet a given delay threshold. Therefore, when performing OR-DSP, P-OLSR, and GRAA, the packet timeout rate is greater than DSRA. It can be seen from Figure 6 that, in all four methods, the timeout rate increases as the movement speed increases, which can be explained by the higher mobility of the nodes per transmission leading to poorer link quality. One of the main optimization goals of DSRA is to maximize the effective data transmission rate, taking into account both the data transmission rate and the link quality. Nodes can adjust the transmission data rate to adapt to the changes in the network topology to get a better link quality; thus it can be concluded that the timeout rate of DSRA is lower than that of OR-DSP, P-OLSR, and GRAA at different speeds. The increase in the number of nodes gives more opportunities for each sending node to make better routing decisions to improve transmission quality. As a result, the timeout rate decreases as the number of nodes increases, as shown in Figure 6.

The problem is formalized as an optimization problem with maximizing the minimum transfer rate as an objective

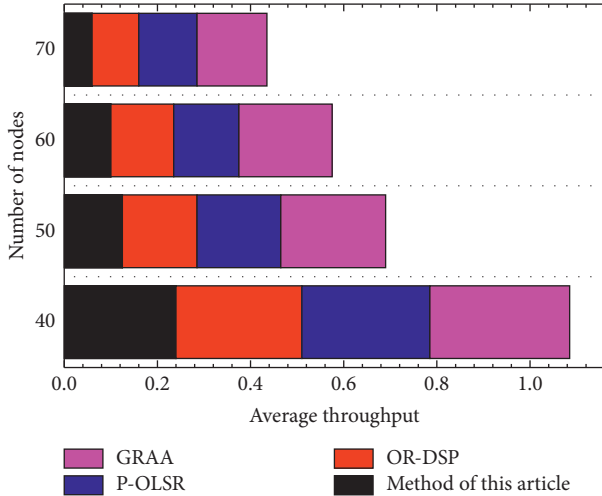


FIGURE 6: Packet loss rate for four routing methods with different number of nodes.

function and the total delay consumption of end-to-end transfers as a constraint, and local approximate solutions are obtained using first-order derivatives and projection methods. To avoid routing into a local optimum, each node selects a set of available relays based on one-hop neighbor information and computes transfer probabilities for nodes in each relay set considering both the remaining latency of the packet and the distance from the relay node to GB. Each sending node passes the packet to the relay node using the transfer probability and local channel state information. Finally, it is demonstrated that the proposed method can converge to an optimal solution when certain conditions are satisfied. From the simulation results, the proposed algorithm in this chapter has good performance in terms of network throughput, packet loss rate, and packet-timeout rate.

**3.2. Analysis of System Performance Test Results.** Trajectory data is a kind of multidimensional data, which contains not only the position information (latitude and longitude) of trajectory points but also a variety of other-dimensional information, such as time, speed, and direction. For the multidimensional information in the track data, this system has developed an attribute visualization module, which aims to synthesize different attributes of the track data and dig deep into the hidden space-time laws. The attribute visualization module contains several submodules: yaw analysis, velocity visualization, direct visualization, and correlation analysis, each of which is applied to the analysis of different track attributes. In this paper, the chosen visualization method is a line graph to visually depict the yaw situation. The period of the selected data source is abstracted to the  $[0, 1]$  interval, and then the deviation distance of each data recording point on the interval concerning the main course is calculated, and to facilitate the visual presentation, they are also coded and normalized to the  $[0, 1]$  interval to show the degree of deviation of the track points. As shown in Figure 7,

the yaw analysis of the selected four-ship trajectory data is presented.

The direction is another important property in trajectory data, and exploring the distribution of movement directions can both find certain movement patterns and patterns and may also reveal some unusual movement patterns. For example, drifting may occur when there is a large deviation between the direction of the ship's route and the ship's course, which is considered a dangerous behavior in the maritime field. Therefore, visual analysis for the directional information of the trajectory data is also very necessary. In this paper, two headings rose diagrams are designed to visually analyze the directional information in the trajectory data for exploring and analyzing the traffic flow direction. As shown in Figure 8, the distribution of traffic flow in different directions is visualized, and the distribution of traffic flow velocity (average velocity in the direction taken) in different directions is also visualized. The percentage of stationary points (trajectory points with almost zero velocity) in the trajectory data is shown in the middle of the rose plot. Figure 8 shows the directional distribution of vessel trajectory data for a week at a port, with outbound in the northwest and homeward in the southeast. The outbound traffic for this week is much more than the homing traffic, and because the ships leave the port right in front of the ocean, the outbound traffic is also relatively fast. In Figure 8, we can see that there is an abnormal traffic flow in the direction far away from the main traffic direction, and the average speed is relatively fast, which may be a detour from the main route due to special circumstances encountered in a certain area.

This test compares the force guided layout algorithm with the OPTFR algorithm. After several simulations, set the number of iterations to 250,  $m = 100$ , and depth  $l$  is 3. Set the number of nodes generated by the data simulator each time to 50, 100, 150, 200, 400, and 800, and the time unit for calculating the latitude and longitude of the nodes is  $s$ . The test results are shown in Figure 9.

The results show that as the number of nodes increases, the OPTFR algorithm computes node positions faster than the force guided layout algorithm; in particular, with more nodes, this contrast is more pronounced. The value of set in the OPTFR algorithm can seriously affect the performance of the algorithm itself; when  $m$  is consistent with the number of nodes in the graph, the algorithm degrades to a force guided layout algorithm; when the value of  $m$  is relatively small, the algorithm's  $m$  value is also related to the number of depth layers  $l$  between adjacent nodes, and it is found that when  $l$  is increasingly small, the computation of repulsion decreases, resulting in distant nodes that are a little closer than before; how to take the value of  $l$ , according to the specific business-specific analysis. The server used by the system is Tornado, which provides services through asynchronous IO single thread, so the CPU resources are not fully utilized; because the service is not high on the CPU requirements, the single-core processor is not fully utilized. Tornado uses an IO callback function, which increases the memory usage as the number of requests increases. Also,



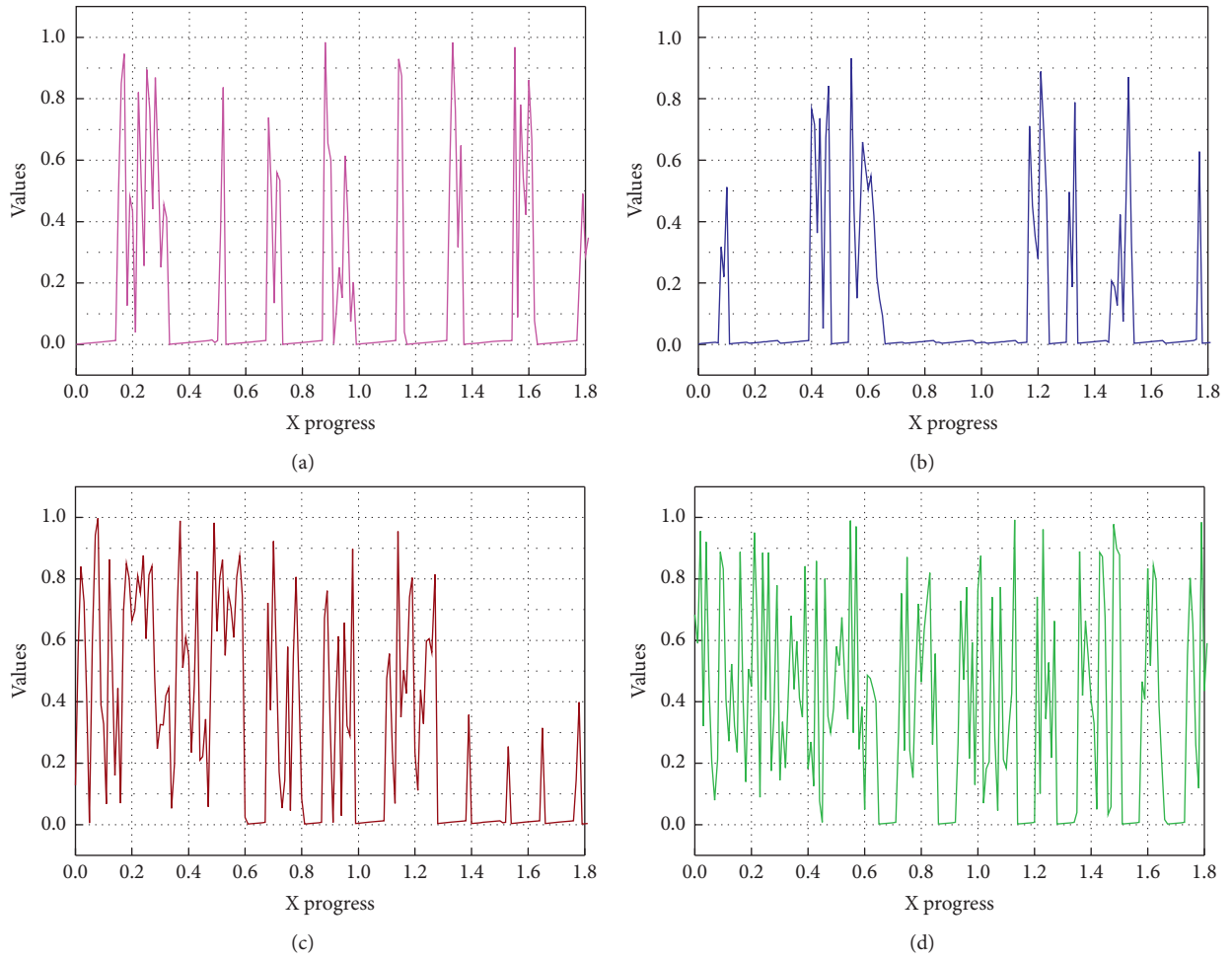


FIGURE 7: Trajectory yaw analysis.

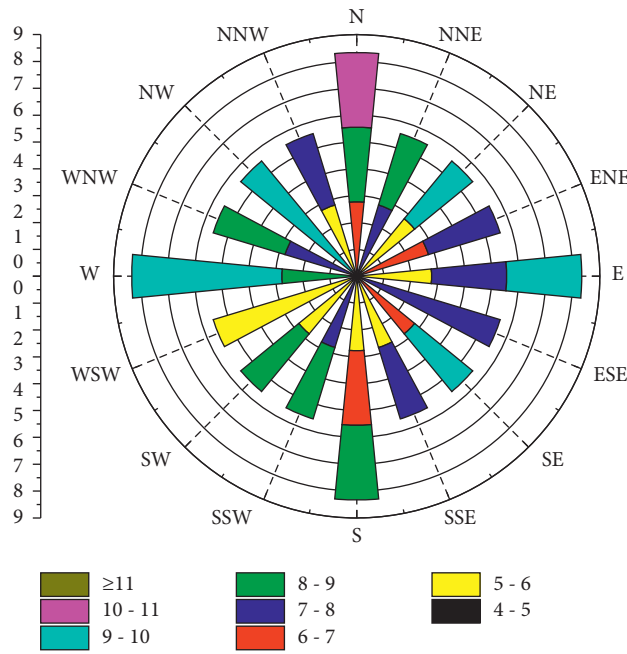


FIGURE 8: Rose diagram analysis.

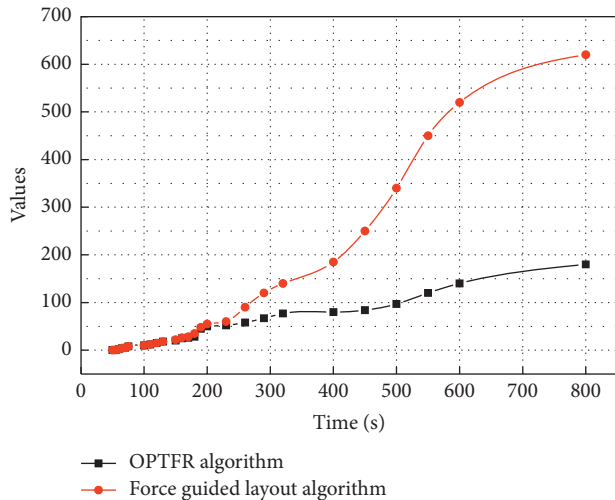


FIGURE 9: Time comparison chart.

Tornado introduces a caching mechanism, which also increases memory consumption.

#### 4. Conclusion

Design and implement a network topology visualization system to visualize complex nodes and internode connectivity information in a page and provide users with a variety of dimensional data presentation methods, enabling users to analyze and monitor network data in the topology by filtering, moving, and modifying attributes and other interactive methods. The interference control problem for satisfying end-to-end delay constraints is formalized as an optimization problem targeting data transmission rate and transmit power. The link-layer constraints are eliminated using a pairwise decomposition method and the end-to-end delay constraint is transformed into a single-hop delay constraint based on a horizontal decomposition method. To maximize the objective function and reduce the local delay in the distributed model, the single-hop delay estimated for each node in the network is used as a local delay constraint. Based on the local neighbor information nodes performing parameter update operations independently, the convergence of the algorithm using a subgradient method to update the logarithmic parameters and first-order derivative method to obtain the solution of the original problem is considered. A distributed interference control algorithm is also proposed, which does not require nodes to collect global information and each relay node performs the data transmission task based on local channel information only. Finally, the convergence of the optimization method is proved and its performance is verified by the results of simulation experiments. In contrast, the non-line-of-sight blocking control algorithm proposed in this paper assumes the existence of at least one available relay node per transmitter node at any moment, and the empty region problem is not considered. Although the method proposed by GPSR can solve this problem well, it does not consider the delay constraints, and delivering packets using the right-hand rule will consume more delay. Therefore, in future work, there is

a need to devise a routing method that not only solves the empty region problem but also satisfies the latency constraint.

#### 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 known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### References

- [1] T. Gernat, V. D. Rao, M. Middendorf, H. Dankowicz, N. Goldenfeld, and G. E. Robinson, "Automated monitoring of behavior reveals bursty interaction patterns and rapid spreading dynamics in honeybee social networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 7, pp. 1433–1438, 2018.
- [2] T. Fujiwara, J. K. Li, M. Mubarak et al., "A visual analytics system for optimizing the performance of large-scale networks in supercomputing systems," *Visual Informatics*, vol. 2, no. 1, pp. 98–110, 2018.
- [3] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 26–32, 2018.
- [4] X. Fan, C. Li, X. Yuan, X. Dong, and J. Liang, "An interactive visual analytics approach for network anomaly detection through smart labeling," *Journal of Visualization*, vol. 22, no. 5, pp. 955–971, 2019.
- [5] R. A. Rossi, N. K. Ahmed, R. Zhou, and H. Eldardiry, "Interactive visual graph mining and learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, pp. 1–25, 2018.
- [6] J. Nayak and B. Naik, "A novel honey-bees mating optimization approach with higher order neural network for classification," *Journal of Classification*, vol. 35, no. 3, pp. 511–548, 2018.
- [7] B. Chakrabarty, V. Naganathan, K. Garg, Y. Agarwal, and N. Parekh, "NAPS update: network analysis of molecular dynamics data and protein-nucleic acid complexes," *Nucleic Acids Research*, vol. 47, no. W1, pp. W462–W470, 2019.
- [8] H. Senaratne, M. Mueller, M. Behrisch et al., "Urban mobility analysis with mobile network data: a visual analytics approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 5, pp. 1537–1546, 2018.
- [9] N. S. Frolov, V. A. Maksimenko, M. V. Khranova, A. N. Pisarchik, and A. E. Hramov, "Dynamics of functional connectivity in multilayer cortical brain network during sensory information processing," *The European Physical Journal Special Topics*, vol. 228, no. 11, pp. 2381–2389, 2019.
- [10] D. Piovesan, G. Minervini, and S. C. E. Tosatto, "The RING 2.0 web server for high quality residue interaction networks," *Nucleic Acids Research*, vol. 44, no. W1, pp. W367–W374, 2016.
- [11] B. Rieck, U. Fugacci, J. Lukaszczuk et al., "Cliques community persistence: a topological visual analysis approach for

- complex networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 822–831, 2018.
- [12] F. Shahzad, T. R. Sheltami, and E. M. Shakshuki, “Multi-objective optimization for a reliable localization scheme in wireless sensor networks,” *Journal of Communications and Networks*, vol. 18, no. 5, pp. 796–805, 2016.
- [13] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, “Designing neural networks through neuroevolution,” *Nature Machine Intelligence*, vol. 1, no. 1, pp. 24–35, 2019.
- [14] N. C. Luong, D. T. Hoang, S. Gong et al., “Applications of deep reinforcement learning in communications and networking: a survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.
- [15] F. Shahzad, T. R. Sheltami, and E. M. Shakshuki, “DV-maxHop: a fast and accurate range-free localization algorithm for anisotropic wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2494–2505, 2017.
- [16] K. Abboud, H. A. Omar, and W. Zhuang, “Interworking of DSRC and cellular network technologies for V2X communications: a survey,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9457–9470, 2016.
- [17] Y. Xu, T. Yu, and B. Yang, “Reliability assessment of distribution networks through graph theory, topology similarity and statistical analysis,” *IET Generation, Transmission & Distribution*, vol. 13, no. 1, pp. 37–45, 2019.
- [18] R. Mueller-Bady, M. Kappes, I. Medina-Bulo, and F. Palomo-Lozano, “An evolutionary hybrid search heuristic for monitor placement in communication networks,” *Journal of Heuristics*, vol. 25, no. 6, pp. 861–899, 2019.
- [19] C. Nobre, M. Meyer, M. Streit, and A. Lex, “The state of the art in visualizing multivariate networks,” *Computer Graphics Forum*, vol. 38, no. 3, pp. 807–832, 2019.
- [20] B. P. Busby, E. Niktab, C. A. Roberts et al., “Genetic interaction networks mediate individual statin drug response in *Saccharomyces cerevisiae*,” *NPJ Systems Biology and Applications*, vol. 5, no. 1, pp. 1–13, 2019.
- [21] Q.-T. Bui, “Metaheuristic algorithms in optimizing neural network: a comparative study for forest fire susceptibility mapping in Dak Nong, Vietnam,” *Geomatics, Natural Hazards and Risk*, vol. 10, no. 1, pp. 136–150, 2019.
- [22] A. E. Sizemore, J. E. Phillips-Cremens, R. Ghrist, and D. S. Bassett, “The importance of the whole: topological data analysis for the network neuroscientist,” *Network Neuroscience*, vol. 3, no. 3, pp. 656–673, 2019.
- [23] T. Ozcan and A. Basturk, “Transfer learning-based convolutional neural networks with heuristic optimization for hand gesture recognition,” *Neural Computing and Applications*, vol. 31, no. 12, pp. 8955–8970, 2019.
- [24] A. Martin, J. Egaña, J. Flórez et al., “Network resource allocation system for QoE-aware delivery of media services in 5G networks,” *IEEE Transactions on Broadcasting*, vol. 64, no. 2, pp. 561–574, 2018.
- [25] H. Shukla, R. Shukla, A. Sonkar, and T. Tripathi, “Alterations in conformational topology and interaction dynamics caused by L418A mutation leads to activity loss of Mycobacterium tuberculosis isocitrate lyase,” *Biochemical and Biophysical Research Communications*, vol. 490, no. 2, pp. 276–282, 2017.
- [26] S. Murugesan, K. Bouchard, J. A. Brown et al., “Brain modulyzer: interactive visual analysis of functional brain connectivity,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 4, pp. 805–818, 2016.