Security Awareness Method of Opportunistic Network Routing Protocol Based on Deep Learning and Knowledge Graph

Yan Zhao(✉) and Xucheng Wan

Ningbo City College of Vocational Technology, Ningbo 315199, China
zhaoyan20210606@163.com

Abstract. In the opportunistic network, the security of the routing protocol operation is low, so we design a security awareness method of the opportunistic network routing protocol based on deep learning and knowledge graph. Design a data acquisition platform to implement the data acquisition of opportunistic network routing protocol. The platform can be divided into four functional modules: data acquisition, data analysis, human-computer interaction interface and system management. The steps of building ontology structure, entity extraction, knowledge reasoning, and knowledge graph storage build routing protocol knowledge graph. An opportunistic network routing protocol intrusion detection method based on DCAEs is designed to realize the security awareness of opportunistic network routing protocols. The test results show that the security perception accuracy of this method is stable at 0.96 after running for a period of time, and the overall security perception accuracy is relatively high.

Keywords: Deep Learning; Knowledge Graph · Opportunistic Networks · D2RQ Tools · Routing Protocol Security Awareness

1 Introduction

The development wave of industrial automation promotes the development of high and new technologies such as information sensing, data communication and data processing., wildlife migration tracking and many other fields, social development has gradually entered the Internet of Things information era.

In order to meet the requirements of ubiquitous interconnection and comprehensive perception of the Internet of Things, intelligent devices need to be interconnected. Therefore, the networking technology between devices has increasingly become the focus of the Internet of Things research. In terms of urban intelligent transportation, vehicles equipped with intelligent devices use the short-range wireless communication function to form an in-vehicle self-organizing network, which realizes the mutual transmission of traffic road condition information between vehicles, improves the efficiency of traffic travel and ensures the safety of urban traffic travel; in terms of marine environment monitoring, by installing sensing equipment with wireless communication capabilities

B. Wang et al. (Eds.): ICMTEL 2023, LNICST 534, pp. 155–168, 2024.
https://doi.org/10.1007/978-3-031-50577-5_12

on ocean buoys, a marine wireless sensor network is formed, and the network is used to collect information on parameters such as seawater temperature, oxygen content, and pH, so as to monitor sea state information and provide marine safety. Jobs provide information assurance. However, in practical applications, ad hoc networks often face problems such as sparse distribution of nodes and drastic changes in network topology, which cannot guarantee network connectivity. Therefore, traditional mobile ad hoc network communication protocols, such as AODV and DSR, are no longer suitable for these complex scenarios. Because the condition for the application of these traditional communication protocols is to ensure that there is no less than one end-to-end link to ensure complete connectivity between any node pair in the network, and this condition is difficult to be satisfied in an actual ad hoc network, it is difficult to ensure that the network transmission performance [1].

In order to solve the above communication problems, in 2006, Pelusi et al. proposed a new communication mode called mobile opportunistic network in view of the dynamic topology and other characteristics faced by self-organizing networks. Based on the original five-layer network architecture, the mobile opportunistic network introduces a bundle layer between the application layer and the transport layer. The bundle layer transforms the original "store-forward" data communication mode of network nodes into a "store-carry-forward" communication mode, transforming the disadvantage of dynamic network topology changes into applicable features, relying on the opportunity contact generated by node movement, The relay node is selected to forward the data packet until the data packet reaches the destination node.

For mobile opportunistic networks, the selection of appropriate relay nodes to carry data packets is critical to the performance of network transmission. At the same time, formulating appropriate data packet management strategies on nodes can also improve network data processing capabilities, both of which are Depends on the performance of the network routing protocol. Therefore, formulating reasonable and effective routing protocols for mobile opportunistic networks according to their network characteristics and node characteristics has become a hot research topic in this field. By further analyzing and extracting network characteristics such as mobility characteristics of mobile opportunistic network nodes, historical interaction information, data packet transmission process, etc., and then formulating routing protocols suitable for the network environment based on these characteristic information, it can ensure the effective transmission of data under extreme conditions., and then improve the network performance of the mobile opportunistic network, and promote the wide application of this communication mode in many fields. In the research of opportunistic network routing protocols, the security awareness of routing protocols is a key research problem. Now we have designed a security awareness method for opportunistic network routing protocols based on deep learning and knowledge graph.The experiment verifies that compared with the two literature methods, the overall security perception accuracy of this design method is higher, so as to ensure the network transmission performance.

2 Opportunistic Network Routing Protocol Security Awareness

2.1 Opportunistic Network Routing Protocol Data Collection

Design a data acquisition platform to implement the data acquisition of opportunistic network routing protocol. The data acquisition platform can be divided into four functional modules: data acquisition, data analysis, human-computer interaction interface and system management [2]. In the data acquisition module, the data acquisition equipment used is a self-designed and developed PIC controller, with 4 analog input ports and 10 digital input ports, which can realize the collection of analog and digital data of multiple channels, and It has 14 digital output ports. Using these output ports, the upper-level software can control the underlying equipment and realize the real-time data acquisition function under closed loop. The controller transmits the collected data through the network port, and the data communication format adopts a self-defined data format. In order to realize data transmission, the controller uses a self-defined data transmission format during data transmission. This is conducive to ensuring the security of data transmission, and more importantly, it is convenient for later functional expansion. The data in communication are generally transmitted in the form of data packets, which are generally referred to as a frame of data. Protocols that can ensure the correct transmission of data A frame of data is generally composed of frame header, address information, data type, data length, data block and check code. The function of the frame header is to judge whether the data packet is lost during the transmission process. It is required that the frame header should appear the least number of times during a data transmission process, so as to reduce the possibility of data transmission errors caused by judging the data as the frame header as much as possible. Sex [3]. The address information is mainly used in multi-computer communication to realize the distinction of different devices. Here, the multi-channel data acquisition is realized through the controller, and multiple controllers are not involved. Therefore, this part is not involved in the data transmission format design. The three parts of data type, data length, and data block are the main parts of data transmission. The data length is used to identify the number of valid data contained in the data frame, and the data can be easily parsed according to this identification. The function of the check code is to check whether the data is lost and whether the data is correct during data transmission. The controller uses the TCP protocol to transmit data, and the TCP protocol uses mechanisms such as deterministic retransmission to ensure the accuracy of data transmission. Therefore, in order to ensure the efficiency of data transmission, the data format design does not use error checking.

The custom data transmission format of the data acquisition device includes data, port number, data length, identification code and frame header.

The frame header is represented by two bytes, using 0x4111 as the start of each data frame.

Identification code, one byte, used to indicate the type of the port.

Data length, one byte, indicating the number of valid data following.

Port number, one byte, the different ports of the controller are determined by the port number, and the upper software obtains the information of the specified port. There are 4 analog input ports with port numbers 1–4, ten digital ports with port numbers 01–10, and 14 output ports with port numbers 01–14.

Data, two bytes, is the data collected by the controller or the data that needs to be specified by the output port. For decimals, the processing method of its magnification will lose a certain precision, but it can also meet the vast majority of needs. This method can simplify the transmission of decimal data.

The data acquisition module realizes the communication function with the data acquisition equipment, receives the acquisition data uploaded by the data acquisition equipment, parses the data according to the corresponding data transmission protocol, and then displays and stores the acquired data. Therefore, two methods of serial communication based on Modbus-RTU protocol and network communication based on TCP/IP protocol are implemented in the module. At the same time, in order to ensure the increased demand of developers or users for communication methods due to certain requirements in the later stage, the design of The related function interface can easily add the function implementation of other communication methods, which ensures the scalability of the software [4].

The communication part is composed of ServerManager class, including TcpServer sub-module, Udpserver sub-module, Modbus Server sub-module, and self-defined communication mode realization sub-module. The ServerManager class is a communication service management class, which is mainly responsible for managing the implementation of various communication methods, including functions such as adding, registering, and calling. Its main functions include the management and invocation of the implemented communication methods, which receive the user's parameter configuration of the data acquisition equipment using the software, and invoke the relevant communication methods according to the user's choice. Secondly, this class realizes the loading and registration of the communication mode implemented by the user, realizes the unified management and invocation of all communication modes, and can easily realize the function expansion of the user-defined communication mode.

In the network communication mode of TCP/IP protocol, a communication server based on TCP protocol and UDP protocol is designed and implemented. The communication server designed and implemented can meet the needs of various aspects related to communication, and can meet the flexible and extensible development ideas mentioned above, and can realize the functional requirements of users adding custom data processing processes. For the realization of the communication function, the communication function of the socket is divided into two parts, one is the server class, the main function is to establish the communication with the underlying data acquisition equipment to realize the basic network communication operation, the other is the request processing class, This class mainly realizes how to process the data after uploading the data from the data acquisition device [5]. Through this form of separating communication establishment and request processing, subsequent development only needs to focus on the business logic part, that is, focusing on development and processing of the acquired collected data. This separation is conducive to the scalable development of the entire communication function. In the realization of the server, the idea of object-oriented development is adopted, and different classes are constructed according to different functional requirements, including BaseServer, TcpServer, UdpServer, ThreadTcpServer, and ThreadUdpServer.

The BaseServer class is the base class of all classes. All possible related operations in the network communication process are defined in this class, but there is no specific implementation, which needs to be implemented in subclasses. Both TcpServer and UdpServer inherit from BaseSever, and implement the functions they need in the base class according to the characteristics of the TCP protocol and the UDP protocol. The two classes ThreadTcpServer and ThreadUdpServer are implemented by inheriting TepServer and UdpServer respectively. These two classes realize the function of processing multi-client communication under multi-threading. When using these two classes to establish a communication server, whenever a connection arrives Every time a new thread is established, the data transmission between the software and the data acquisition device is completed in this thread [6]. In the realization of the whole function, BaseServer is the key of the whole design, and the realization of other classes is based on the rewriting of different functions according to the functional characteristics of their respective implementations.

HandleRequest The member of this class is the request processing class object. The server class will receive the request processing class object and call its related functions to realize the data parsing and processing function. This parameter is the parameter that needs to be passed in when the subclass instantiated object of BaseServer is initialized.

address This member is the host IP and port that need to be bound when establishing socket communication. This parameter also needs to be passed in as a parameter when the instance of its subclass is initialized.

init_() initialization function, responsible for the initialization of class instance objects.

get_client() This function receives the socket request and returns the new socket object and client address to communicate with the client.

handle_error() When the processing of data in the request processing class reports abnormal errors, this function is responsible for handling these exceptions.

handle_error() When the processing of data in the request processing class reports abnormal errors, this function is responsible for handling these exceptions.

_handle_request() This function processes a single request. In this function, the get_client() function is called to obtain the socket object, and then the HandleRequest class is instantiated, and its related methods are called to process the data.

run() function, after this function is called, it will start a loop, continue to receive the data transmitted by the client and call the relevant processing functions in the request processing class to parse and process the data. In this function, the IO multiplexing technology is used. In this function, select is used to monitor network IO. When data arrives from the connection, the processing function will be called.

Another important part of the communication module design is the design of the request processing class. The function of the request processing class is how to parse the data uploaded by the data acquisition device. The user may upload the collected data according to the data format defined by himself, so the corresponding data parsing function is also required on the software side. The user can rewrite the request processing class, and then the software loads and runs to realize these functions. When implementing these functions, users need to use the RequestHandle class as the base class and rewrite the relevant class member functions to ensure that they can be successfully called.

The class member functions are designed as follows:

The sock member variable refers to the new socket variable after the server and the client establish a connection.

client_address is the client IP address.

The _init_0 function is the initialization function of the class. In this function, the member variables are assigned values and the execute() function is called. Therefore, when using this class, you only need to create an instance object of the RequestHandle subclass in the server.

init() function This function is called before the execute() function, and mainly implements various initialization work before executing the processing request. When the user implements, if there is other initialization work, this function can be rewritten.

execute() function is a function that must be overridden by subclasses. How to realize the correct parsing of data is implemented in this function.

The function of stop() is to release the occupied related resources after the request is processed.

The serial communication based on the Modbus-RTU protocol realizes the communication connection between the software and the equipment using the Modbus protocol as the communication protocol through the serial port by designing a communication module based on the Modbus protocol.

According to the different device interfaces, the Modbus protocol is divided into two forms: Modbus-RTU used in the serial port and Modbus-TCP used in the network port. The designed platform provides specific implementations for these two forms, which can ensure that the data acquisition equipment using the Modbus protocol can be successfully connected to the software regardless of whether the equipment has a serial port or a network port interface. The two forms of realization are Modbus RtuServer and Modbus TcpServer [7].

The main function of the data analysis module is to provide some data analysis, numerical processing and some data visualization functions, enabling users to quickly build their own algorithms, process and analyze the collected data, and visualize them.

Among them, the calling storage function of the data is shown in Table 1.

Table 1. The Calling stored function of the data

Serial number	Function function	Function name
1	Data stored in a table	SaveTable()
2	Read the data in the table	ReadTable()
3	Storage operation of CSV text format data	Save_ csv()
4	Read operation of CSV text format data	Read_ csv()
5	Real time data acquisition	GetRealTimeData()

Communication functions include close(), write(), read(), Connect(), and init().

Matrix functions include init_(), multiply(), dot(), inverse(), transpose(), rank(), eig_value(), eig_vector().

Data preprocessing functions include outlier processing functions, data noise smoothing processing functions, data normalization processing functions, interpolation, fitting and filtering functions.

Data visualization functions include data visualization functions based on Matplotlib, data visualization functions based on PyQtGraph, and 3D drawing based on PyQtDataVisualization.

The human-computer interaction interface module mainly realizes various functions of the platform interaction interface, including the parameter configuration interface during data acquisition and communication with the device, the code editing interface used by the user to edit the program, and the realization of the relevant interface for data visualization display.

The system function module includes other auxiliary functions of the entire platform, including platform function management, loading and registration functions of user-defined functions, and exception handling in the platform.

1.2 Building a knowledge map of routing protocols.

The construction process of routing protocol knowledge graph is building ontology structure, entity extraction, knowledge reasoning, and knowledge graph storage.

Ontology is a way for computers to describe everything in the world. Through ontology, consensus on information structure can be shared among software agents, and domain knowledge can be reused. Knowledge graph describes and defines the knowledge and the scope for the knowledge it describes through ontology.

The ontology construction process of the routing protocol knowledge graph is as follows: first, consider the domain and scope of the ontology involved in the routing protocol, then determine the relevant professional terms, then define the involved classes and inheritance relationships, and finally define the attributes and related relationships.

Knowledge extraction can be divided into three different scenarios according to the form of knowledge data, knowledge extraction for structured data, knowledge extraction for semi-structured data and knowledge extraction for unstructured data. The main consideration in the extraction of routing protocol knowledge is to extract knowledge from relational databases. There are two standards for extracting knowledge from relational databases, Direct Mapping and R2RML. Direct Mapping refers to directly mapping table names to class names and columns to attributes. This method is not flexible enough to map the knowledge data in the relational database to the previously defined ontology structure. Therefore, the R2RML standard is mainly used. The D2RQ tool completes knowledge extraction from relational databases.

R2RML is a description language that maps relational databases to RDF structural knowledge. It can take a table, or view, or SQL query as input, and finally map these logically into triple maps. These triple maps are RDF triple, which finally constitutes the knowledge graph.

Extract knowledge from relational databases through D2RQ software. D2RQ is a system that accesses relational databases as independent RDF graph databases. There is no need to convert relational databases into RDF data, and use SPARQL to query non-RDF databases. The specific process is as follows: First, access the PostgreSQL database through D2RQ to generate a mapping file. The mapping file is the key for D2RQ to access the relational database in the form of RDF, but the automatically generated mapping

does not meet the requirements. Next, the mapping needs to be modified according to the ontology structure defined by the previous Protege. File, and finally use the modified mapping file to generate RDF data through D2RQ.

Knowledge reasoning is a way of complementing the knowledge graph. It can use the explicit knowledge in the knowledge graph to infer the undiscovered tacit knowledge, so as to expand the knowledge graph. Relatively few relational data are extracted from relational databases through D2RQ, and the relational data needs to be expanded by means of knowledge reasoning. There are many methods for knowledge reasoning based on knowledge graphs, including reasoning based on description rules, reasoning based on graph structure and statistical rule mining, reasoning based on knowledge graph representation learning, and methods based on logical probability. Here, the reasoning of knowledge graph is mainly carried out based on the description rules. And the relationship between entity data in the existing knowledge graph is expanded by means of description rules. The ontology structure defined by the previous Protege tool is in OWL format. The OWL ontology language itself provides logical reasoning, but it is relatively simple. It can only perform relatively simple reasoning such as parent class, subclass, and opposite relationship, that is, only supports predefined ones. Inference on ontology axioms, while inference based on description rules can customize rules according to specific scenarios, formulate inference rules by yourself, or combine the ontology reasoning provided by OWL ontology language. The Jena tool is mainly used for knowledge reasoning.

The process of using Jena for knowledge inference is as follows: First, you need to define inference rules according to Jena's grammar and knowledge graph expansion requirements, and save the rules as a file to be called by Jena, then start Jena, initialize Jena's data structure, and initialize Jena's inference engine, and then infer to obtain the expanded relational data, and finally write all the data into a new knowledge graph data file.

After knowledge extraction and knowledge reasoning, the routing protocol knowledge graph is obtained.

Typically, knowledge graphs model and represent knowledge in the form of graphs, and knowledge graphs are usually stored in graph databases or RDF storage systems in the form of graph data. The RDF storage system uses the RDF format as the data format. At present, the common RDF storage systems include Virtuoso, RDF4J, etc. However, compared with the graph database, the RDF storage system has disadvantages such as complex deployment and inactive communities, which hinder the knowledge graph storage system. Therefore, a graph database is used to store the knowledge graph.

The graph database is specially optimized for the data of large-scale graph structure. The basic model of the general graph database is composed of attribute graphs, including nodes, edges and attributes, which can perfectly match the knowledge graph constructed by this system. Common graph databases include Neo4j, JanusGraph, ArangoDB, and TigerGraph. At present, Neo4j is the most popular graph database. The community is active and the ecology is mature. There is a specialized and easy-to-use Cypher query language, and Neo4j is used as the repository of knowledge graphs.

The basic process of knowledge graph storage is as follows: First install Neo4j, since the RDF data needs to be stored in Neo4j, the main operation is to use the sematics

toolkit, and then download the sematics toolkit to the plugins file of Neo4j, then start Neo4j, and then enter the command to perform data import.

2.2 Security Perception

An opportunistic network routing protocol intrusion detection method based on DCAEs is designed to realize the security awareness of opportunistic network routing protocols. First, a new dilated autoencoder model is designed, which realizes a convolutional autoencoder without pooling operation by transposing the convolution, and minimizes the error of input and reconstruction to learn the hidden layer unsupervised. Features, and learn more global features without loss of information through dilated convolutions. Then, the training process of the intrusion detection model based on the dilated convolutional autoencoder is elaborated, which includes data preprocessing, unsupervised pretraining and supervised fine-tuning.

The structure of the designed dilated convolutional autoencoder model is similar to the classical autoencoder. Since the knowledge graph data of the routing protocol used to train the neural network belongs to text information, the pooling operation will lose some information, so the convolutional layer in DCAEs After the pooling layer is not added, the dilated convolution is used instead of the ordinary convolution operation, and a larger receptive field can be obtained without increasing the model parameters, which is more advantageous than the pooling operation. The input of the dilated convolutional autoencoder is mapped into a feature map by the feature function:

$$a^b = g\left(c * Q^b + d^b\right) \tag{1}$$

In formula (1), c refers to a two-dimensional numerical matrix converted from a one-dimensional numerical vector; Q^b refers to the weight matrix of the b th feature map a^b; d^b refers to the b th feature map a^b bias vector; $g(\cdot)$ Refers to the ReLU activation function; $*$ refers to the dilated convolution operation.

The ReLU activation function is a non-saturating function. When the input is less than 0, the output is 0, and the neuron is in a suppressed state. When the input is greater than 0, the output is proportional to the input. The ReLU function has the following advantages: Compared with the sigmoid function and the tanh function, the use of the ReLU function in the stochastic gradient descent algorithm can significantly speed up the convergence speed; the calculation is simple, and complex exponential and reciprocal operations are not required; it alleviates the phenomenon of gradient disappearance and enables training Deeper network; because the ReLU function is suppressed when the input value is less than 0, it can obtain a sparse solution, which strengthens the sparse expression ability of the neural network. The disadvantage of the ReLU function is that when the input value of the neuron has a large gradient value during the training process, the neuron has a gradient of 0 after the parameter update, and the neuron will never activate again. If the learning rate is set too high, the network may 40% of the neurons are no longer activated, so in practical applications, a reasonable learning rate needs to be set.

Next, the feature map of the hidden layer is mapped to the reconstruction of the input of the dilated convolutional autoencoder by a transposed convolution operation:

$$\tilde{c} = g\left(\sum_{b\in B} a^b * Q^b + d^b\right) \tag{2}$$

In formula (2), B represents the set of feature maps; $\tilde{c}$ and c have the same shape.

Among them, dilated convolution, also known as hole convolution, introduces a dilation rate hyperparameter in the convolution layer, that is, there is a gap between filter elements. Compared with ordinary convolution operations, dilated convolution provides a larger receptive field at the same computational cost. The expansion convolution diagram is shown in Fig. 1.

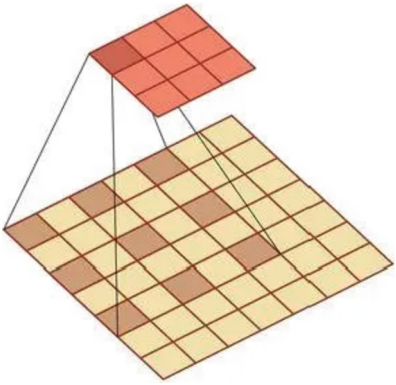

Fig. 1. Expansion convolution diagram

Transposed convolution is also known as microstepped convolution, but calling it deconvolution is actually wrong, the actual mathematical operations of transposed convolution and deconvolution are different, and deconvolution enables convolution operations The mathematical inverse process of, but the transposed convolution is just a conventional convolution operation. In the image task, the image size can be increased to reconstruct the original image spatial resolution. The transposed convolution is a transformation in the opposite direction of the convolution operation. The shape of the output of the convolution operation is converted to the shape of the original convolution input and maintains an effective connection pattern, which can be used in the decoding layer of the convolutional autoencoder or to map the feature map to a high-dimensional space.

The optimization goal of the dilated convolutional autoencoder is to reduce the difference between the input c and the reconstructed $\tilde{c}$, so that the loss function can reach a minimum value, and the loss function is selected as the mean square error function:

$$Y(c, \tilde{c}) = \frac{1}{m}\sum_{j=1}^{m}\left(c_j - \tilde{c}_j\right)^2 \tag{3}$$

In formula (3), mmmmm m represents the number of feature maps; ccccccjjjjjj c_j refers to the j th c matrix; $\tilde{c}_j$ refers to the j th reconstructed $\tilde{c}$.

Deep neural networks can be constructed by stacking multiple dilated convolutional autoencoders, a process similar to stacking autoencoders. Specifically, the input of the next convolutional autoencoder is the output of the hidden layer of the previous convolutional autoencoder, and the process of stacking dilated convolutional autoencoders is an unsupervised layer-by-layer greedy training process.

The training process of the intrusion detection model is mainly divided into three stages: data preprocessing, unsupervised pretraining and supervised fine-tuning. First, in the data preprocessing stage, the raw network traffic in Libpcap file format is converted into numerical vectors by a session-based data preprocessing module, and these numerical vectors are the training samples in the dataset. Second, in the unsupervised pre-training process, the dilated convolutional autoencoder learns important hierarchical feature representations from a large number of unlabeled training samples; finally, supervised fine-tuning is further optimized from unlabeled samples through a backpropagation algorithm and a small number of labeled samples Supervise the features learned during the pre-training process to optimize the model parameters.

The structure of the dilated convolutional auto-encoder of the fine-tuning process is the same as the network structure of the final test process. The structure of the dilated convolutional auto-encoder of the fine-tuning process is a convolutional neural network without a pooling layer and replacing the ordinary convolution operation with dilated convolution. Network. In order to facilitate the dilated convolution operation, the original training samples are transformed into the shape of the image. The convolutional autoencoder has only one convolutional layer, and the fine-tuning process uses an early stopping mechanism to avoid overfitting. The fully connected layer before the Softmax classifier is the last learned abstract feature, and the Softmax classifier uses the output of the fully connected layer as the input of the classifier to perform the classification task. The network intrusion detection model based on DCAEs can handle different kinds of raw network traffic, and unsupervised pre-training does not require a large amount of labeled data, so it has good adaptability and flexibility.

Dilated convolutional autoencoders combine the concepts of autonomous learning and representation learning. Autonomous learning and unsupervised feature learning are similar. The difference between the two is that the distribution of unlabeled data used for autonomous learning is not necessarily the same as that of labeled data. For example For image classification tasks, the types of unlabeled image data may be much more than those with labels. After training the neural network with these unlabeled data, the labeled data can still be classified correctly. Representation learning, also known as feature learning, can automatically learn useful feature representations from raw data, replacing the traditional machine learning method of artificially constructing features based on feature engineering. The advantages of dilated convolutional autoencoders are as follows: dilated convolutional autoencoders enable dilated convolutional autoencoders to have a larger receptive field to learn more global information without increasing the computational cost. The pooling operation is used to free the input data from information loss; the unsupervised pre-training process only requires a large amount of unlabeled data. Due to the scarcity of labeled data, the dilated convolutional autoencoder is more suitable

for practical applications; dilated convolutional autoencoders The encoder has fewer training parameters than the fully connected neural network, which is more efficient and time-saving than other fully connected unsupervised deep learning methods; since the activation function of the dilated convolutional autoencoder is the ReLU function, there is no gradient disappearance problem, so Dilated convolutional autoencoders do not have the same depth restrictions as fully connected neural networks. Generally, the deeper the neural network, the stronger the expressiveness or learning ability, which enables the dilated convolutional autoencoders to process high-dimensional data and build more Deep deep learning architectures.

3 Experimental Tests

3.1 Experimental Method Design

For the designed security perception method of opportunistic network routing protocol based on deep learning and knowledge graph, its perception performance is tested through experiments. Set the number of experiment iterations as 20.The opportunistic network is first simulated.

The chosen opportunistic network is a mobile social network. Mobile social networking is to using everyone's mobile phone and the surrounding WLAN hotspots to organize into an opportunistic network. Access to the Internet to achieve global communication. When a communicator is far away from the WLAN access point and cannot directly communicate with it, the information can be forwarded by other mobile phones in the network, and the information can be forwarded to the WLAN access point in a hop-by-hop manner. After the information is obtained on the Internet, the information is returned to the correspondent in the same way. This makes full use of the storage, computing and bandwidth resources of each mobile phone, reducing the burden on the 5G network to a certain extent.

The simulation parameters are set as shown in Table 2.

Table 2. The setting of opportunistic network simulation parameters

Serial number	Simulation parameters	Company	Parameter setting
1	Packet size	MB	5
2	Packet lifetime	Min	80
3	Packet generation interval	s	30
4	Node transmission rate	Mbps	6
5	Communication radius	m	20
6	Node cache	MB	50
7	Number of nodes	piece	120
8	Area of simulation area	m2	3200×5300
9	Simulation time	h	30

The routing protocol is simulated through the opportunistic network simulation platform ONE1.60. The selected routing protocol is Epidemic routing protocol.

The basic idea of the Epidemic routing protocol is that every message in the network can generate a copy of the message without limit. When a node moves in the network and encounters other nodes, it sends a copy of the message to the one it encounters that has not yet encountered this. The neighbor node of the message copy, the neighbor node that receives the message copy continues to generate the message copy and send it to the other nodes it encounters that have no copy of the message, and so on, until a copy of the message is delivered to the target node. The Epidemic routing protocol algorithm is essentially a flooding algorithm. Using this type of routing protocol, in theory, every non-isolated node in the network has the opportunity to receive all copies of the information, which can maximize packet transmission. It can improve the success rate of the network, reduce the transmission delay, and effectively improve the quality of network communication when the traffic in the network is small. However, with the increase of the number of nodes or the amount of network information, this transmission method will consume a lot of network communication resources, causing network congestion and greatly reducing the success rate of message delivery to the destination.

The security perception of the routing protocol is carried out by the designed method, and the security perception accuracy of the designed method is tested.

3.2 Safety Perception Accuracy Test

The method of literature [1] and the method of literature [2] are used as comparison methods to test the safety perception accuracy of the three methods, and the test results are shown in Fig. 2.

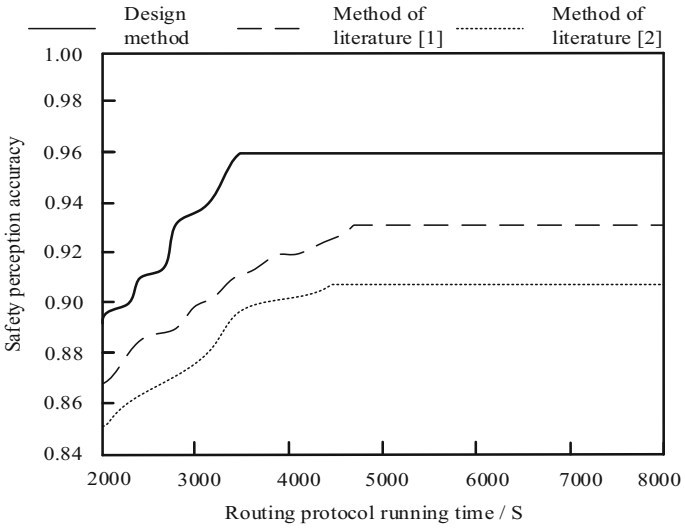

Fig. 2. Safety perception accuracy test results

According to the test results in Fig. 2, the security perception accuracy of the design method is low, which is unstable in the initial operation of the routing protocol and stable at 0.96 after running for a period of time, while the security perception accuracy of the two literature methods is low. Compared with the two literature methods, the overall safety perception accuracy of this design method is higher. This is because the design method builds the data acquisition platform, builds the knowledge mapping of the routing protocol, and designs the new expansion autoencoder model.

4 Conclusion

As a new type of wireless network, opportunistic network has remarkable characteristics such as intermittent link connectivity, high information transmission delay and limited node resources. The security of its routing protocol is also low, so it must be aware of the security of its routing protocol. In this paper, we design a security perception method of opportunistic network routing protocol based on deep learning and knowledge graph. By using four functional modules of data acquisition, data analysis, human-computer interaction interface and system management, we construct a data acquisition platform, construct knowledge mapping of routing protocol, and design a new expansion automatic encoder model to achieve relatively accurate security perception. In the future development, the security perception of opportunistic network routing protocol will be deeply explored to provide a strong theoretical support for the intrusion detection of opportunistic network routing protocol.

References

1. Nuruzzaman, M.T., Feng, H.W.: Beaconless geographical routing protocol for a heterogeneous MSN. IEEE Trans. Mobile Comput. **21**, 2332–2343 (2020)
2. Song, H., Liu, L., Pudlewski, S.M., et al.: Random network coding enabled routing protocol in unmanned aerial vehicle networks. IEEE Trans. Wirel. Commun. **19**, 8382–8395 (2020)
3. Velusamy, D., Pugalendhi, G., Ramasamy, K.: A cross-layer trust evaluation protocol for secured routing in communication network of smart grid. IEEE J. Sel. Areas Commun. **38**(1), 193–204 (2020)
4. Fatemidokht, H., Rafsanjani, M.K., Gupta, B.B., Hsu, C.-H.: Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad Hoc networks in intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. **22**(7), 4757–4769 (2021). https://doi.org/10.1109/TITS.2020.3041746
5. Yue-bo, L., Wei-jie, Z.: Implementation of dynamic clustering scheduling algorithm for social network data. Comput. Simul. **38**(1), 269–272 (2021)
6. Chen, C., Liu, L., Qiu, T., et al.: Routing with traffic awareness and link preference in internet of vehicles. IEEE Trans. Intell. Transp. Syst. **23**, 200–214 (2020)
7. Boudouaia, M.A., Ali-Pacha, A., Abouaissa, A., et al.: Security against rank attack in RPL protocol. IEEE Network **34**(4), 133–139 (2020)