A Fine-grained Dynamic Access Control Method for Power IoT Based on Kformer

Rixuan Qiu, Xue Xue, Mingliang Chen, Jinkun Zheng, Sitong Jing, and Yuancheng Li

Abstract—**The existing static ABAC(Attribute-Based Access Control) model cannot fully meet the increasingly complex, dynamic and scalable demands of the power grid. At the same time, its versatility and flexibility bring high costs. Additionally, the increasing complexity of organizational systems and the need for federated access to their resources make implementing and managing access control more challenging. This paper proposes a fine-grained dynamic access control method based on Kformer to automate authorization management tasks. We use Kformer, which filters and integrates external knowledge through feed-forward layers in Transformer. Then, we use BERT(Bidirectional Encoder Representations from Transformer) to perform feature extraction on the input fused text, extract the implied attribute-authority relationship from the log records and external documents, and finally, perform sequence modeling on the extracted attribute features and input the obtained results. The final authorization result is obtained by classification through the softmax function in the final fully connected layer. The authorization management of the user's request to the object is dynamically completed. Finally, using the access data of the grid information system to evaluate the method proposed by us, the experimental results show that the model can continuously monitor the access behavior of users inside the grid information system, change the access rights of entities and adjust the policy in real-time, and carry out dynamic access authorization. At the same time, the accuracy of the generated access control policy can reach 87.73%.**

Index Terms—**ABAC, Dynamic Authorization, Kformer, Knowledge Injection, Access Control Policy**

I. INTRODUCTION

Access control is one of the essential services responsible for protecting the underlying data of the information system from attack. With the rapid development of computing and information technology, the traditional access control model can no longer meet emerging applications' fine-grained capture and expression security requirements. To improve the power grid informatization and business service capability, the introduced emerging technologies expose some key data of the power grid in the application process, bringing new challenges to the power grid information security [1]. The devices are interconnected and connected to the Internet to realize the effective management of the enterprise. However, there are many security vulnerabilities and threats in data management. The uncertainty may come from internal factors like system failure or external ones like malicious attacks [2]. The Internet of things is dynamic, and there is no clearly defined network boundary. How to solve the access control problem in the Internet of things environment is an important and challenging problem. On the one hand, the security of such protective measures is seriously insufficient, and there is a risk of being broken through. On the other hand, the existing static access control methods and defense strategies are static, solidified, strict, and extensive authority control.

To further standardize the network security system, the International Organization for Standardization (ISO) defines five security services in the network security system standard (ISO7498-2): identity authentication services, access control services, data confidentiality services, data integrity services, and anti-repudiation services. As one of the important components of access control, it was formally proposed in the 1960s and 1970s. In the following decades, many relatively complete and mature access control models appeared successively, among which the widely used models mainly include: Discretional Access Control (DAC) model [3], Mandatory Access Control (MAC) model [4], Role-based access control model (RBAC) [5]. The main problem with these access policies is that they often assign more access rights than the requesting entity needs, exposing system resources to insider attacks [6]. Furthermore, these access policies are manually specified and maintained, assuming the operation is in a closed environment and interaction conditions rarely change. The highly dynamic nature of the IoT environment results in predefined access control policies in the access context that cannot meet policy administrators' security and privacy goals. In an IoT environment, access control policies quickly become outdated due to frequent changes in security and privacy requirements, increasing the risk of insider attacks and making policy management and maintenance tedious and error-prone.

Access control technology will develop in a new direction of fine-grained, flexible, and dynamic. Therefore, an adaptive access control mechanism is needed to respond immediately to changes in the Internet of things environment and refine the access control strategy with minimal or no human intervention at runtime.

The structure of this paper is as follows. Section 2 reviews the existing work on access control policy extraction and application of machine learning. Section 3 designs and implements the main framework of Kformer and describes its use in dynamic access control policy learning. The application and steps of the fourth part introduce the settings and experimental results of the experiment. Section V provides an overview of the results and future work.

Rixuan Qiu, Xue Xue and Yuancheng Li are with School of Control and Computer Engineering, North China Electric Power University (Corresponding author e-mail: ncepua@163.com).

Rixuan Qiu and Jinkun Zheng are with Information & Telecommunication Branch of State Grid Jiangxi Electric Power Supply Co., Ltd.

Mingliang Chen are with State Grid Jiangxi Electric Power Co., Ltd.

Sitong Jing are with PowerChina Jiangxi Electric Power Engineering Co., Ltd.

DOI: 10.36244/ICJ.2022.4.11

II. RELATED WORK

Strategy mining technology has been proposed in the literature to meet these challenges to help organizations reduce the cost, time, and errors of strategy formulation and management. The policy mining algorithm simplifies the migration to the updated or appropriate authorization model by fully (or partially) automating the process of constructing access control policies. Policy mining technology is first introduced into the development of the RBAC strategy. [7] proposed the term "role mining" to refer to a data mining method that constructs roles from a given permission assignment dataset. This work is followed by various role mining technologies, such as [8], [9], [10]. Although the proposed methods help develop the best character set, they are unsuitable for extracting ABAC strategies. [11] first studied the problem of mining ABAC policies from a given access control matrix or log. Subsequently, [12] proposed a new method for mining ABAC policies, including positive and negative authorization rules, by generating authorization logs as the input of the mining algorithm. In contrast, [13] proposed a rule mining algorithm for creating ABAC policies with rules and a scoring algorithm for evaluating policies from a least-privilege perspective, generating fewer over- and under-privileged than RBAC methods.

However, the current solution assumes that the rules are mined from a static dataset of access rights, and this process only needs to be performed once. Whereas in real life, access policies are dynamic and may change depending on the situation. Using the current method, it is necessary to re-execute the mining algorithm for each update in the permissions or user and object attributes, which will greatly reduce the efficiency. At the same time, the above strategy mining method is difficult to find the error of the original authority distribution relationship, and it is difficult to optimize the strategy.

The state of existing devices changes dynamically, for example, sleep and wake, connect or disconnect, and the context of the device, including location and speed. The above will result in unsuitable policies for the current environment, i.e., low-quality rules. If there are a lot of bad rules in the system, it can lead to delayed access and wrong authorization. In the traditional ABAC access control model, the rules will not change during system operation once the rules are specified. Therefore, these static rules are inefficient and fail to comply with the IoT environment.

Machine learning and deep learning methods are gradually applied to the ABAC policy learning problem and significantly improve traditional policy mining methods. [14] proposed Polisma, which combines data mining, statistics, and machine learning techniques. Polisma learns from logs of historical access requests and their corresponding decisions, leveraging latent contextual information obtained from external sources such as LDAP directories to enhance the learning process. [15] proposed a method to automatically learn ABAC policy rules from system access logs to simplify the policy development process. The method employs an unsupervised learning-based algorithm to detect patterns in access logs and extracts ABAC authorization rules from these patterns. Two policy improvement algorithms are proposed to generate higher-quality mining policies, including rule pruning and policy refinement algorithms. However, this unsupervised clustering algorithm is difficult to find a suitable number of clusters, and it is easy to fall into the minimum optimum.

Subject-object dynamic access control determines the legitimacy of the subject's identity according to a predetermined policy and dynamically authorizes resource access requests from trusted subjects. The network security architecture based on zero trust emphasizes the requirements of dynamic authorization and requires evaluation and authentication for all resource access behaviors of each business. We will study a fine-grained dynamic access control method based on Kformer. Our goal is to dynamically calculate and determine according to the subject attribute, object attribute, and environment attribute to prevent the Internet of things device users from abusing their access privileges or using outdated access control policies to obtain unauthorized access, simplify the management of access control policies and realize the real-time evaluation of the service access behavior of distribution Internet of things terminals Real-time authorization and disposal.

III. A FINE-GRAINED DYNAMIC ACCESS CONTROL METHOD BASED ON KFORMER

This section will provide an overview of Attribute-Based Access Control (ABAC), Kformer Network Design, Power IoT, and its authorization framework.

A. ABAC

In 2013, NIST published the Guidelines for ABAC Definitions and Considerations [16], according to which "ABAC engines can be based on specified attributes of requestors, specified attributes of objects, environmental conditions, and a set of policies specified following these attributes and conditions. Make access control decisions." This paper uses subject attributes, object attributes, and ambient attributes to denote requester attributes, and ambient attributes, respectively. A property is any property of a subject, object, and environment, encoded as name: value pairs. Subjects can be personal or impersonal entities. Objects are system resources, operations are functions performed on objects at the subject's request, and environmental conditions are characteristics of the context in which the access request occurs, independent of subjects and objects.

Therefore, it is defined that U,O,E,OP is the set of subjects, objects, environments, and operations in the system, and user attributes (A_U), object attributes (A_O) and session attributes (A_E) are the subject attributes, objects defined in the NIST guidelines, And mapping of properties and environment properties.

User (U): represents a collection of users. The visitor's attributes, such as age, gender, department, role, etc. the user is the service requester interacting with the computing system, and the access request of the computing system is controlled.

Object (O): represents a collection of objects. The attributes of the accessed object, such as the modification time of a record, the creator, the security level, etc.

Environment (E): represents a collection of environmental information, such as time information, geographic location information, access platform information, etc.

Operation (OP): represents a set of operations on a resource. Actions are operations that can be performed on resources, such as CRUD.

The decision D of the authorization tuple can be either allowed or denied. A tuple with permission decisions means that user U canoperaten OP on object O under context attribute E. An authorization tuple with denying means that the user cannot gain such access

ABAC rules (γ): ABAC rules are a tuple

$$\gamma = \{U, O, E, OP, D\} \tag{1}$$

D represents the decision of the ABAC rule for this combination of attributes and the requested action.

B. Kformer network design

This paper decomposes the problem of access control policy generation into two key tasks: external knowledge injection and statement recognition of access control policy. The statement recognition task of the access control policy is to extract access control-related statements from project-related documents (such as user manuals, requirements analysis documents, instructions for use, etc.). Extractive, the attribute feature of Access control policy, extracts the information of subject attribute, action attribute, object attribute, environment attribute, and the relationship between attributes from the policy statements in natural language. We can directly obtain readable and executable access control policies according to these attribute information.

We use a new model Kformer [17], to inject external knowledge. Kformer consists of three main parts: First, the top N pieces of latent knowledge are retrieved from the knowledge base for each question. Then, knowledge representation is obtained through knowledge embedding. Finally, the retrieved N pieces of knowledge are fused into the pre-trained model through the feed-forward layer in the Transformer.

L layers usually stack transformer encoders. Each layer contains a multi-head self-attention and a feed-forward network (FFN). FFN consists of two linear networks. Assuming that the final attention output of the Transformer layer is $x \in \mathbb{R}^d$. The computation of the feed-forward network can be expressed as (omitting the bias term):

$$FFN = f(x \cdot K^T) \cdot V \tag{2}$$

Among them, K,$V \in \mathbb{R}^{d_m \times d}$, Where K,V is the paaremeter matrices of the two linear networks. The input x is first multiplied by K to generate coefficients, which are activated by f and used to compute the weighted sum of V as the output of the feed-forward layer.

The Apache Lucence-based sparse searcher Elasticsearch is used here, using an inverted index lookup. The attribute description and authorization decision are combined as a query sent to the search engine for each policy. The sentences with the highest scores are then selected as candidate knowledge from the results returned by the search engine. Then a dense representation of each knowledge k is obtained through knowledge embedding, and the input sentence x is calculated through the average embedding of each word. Then, use the question embedding and the knowledge embedding to do the inner product to calculate the score of each candidate's knowledge [18]. Finally, we select the top N candidate knowledge scores as external knowledge dependencies and incorporate them into the model in the knowledge injection part.

Each candidate's knowledge is treated as a sentence, and an embedding layer is used for knowledge representation. Initialize the knowledge embedding matrix to be the same as the embedding matrix in Transformer and update the knowledge embedding layer simultaneously during training. For each knowledge $\mathbf{k}$, the tokens are embedded as $k_1, k_2, \ldots, k_l$ via knowledge embedding. k is expressed as the mean of these token embeddings:

$$Embed(\mathbf{k}) = Aug(k_1, k_2, \ldots, k_l) \tag{3}$$

Here, the candidate knowledge is denoted as $k_1, k_2, \ldots, k_M$, Where M is the number of candidate knowledge.

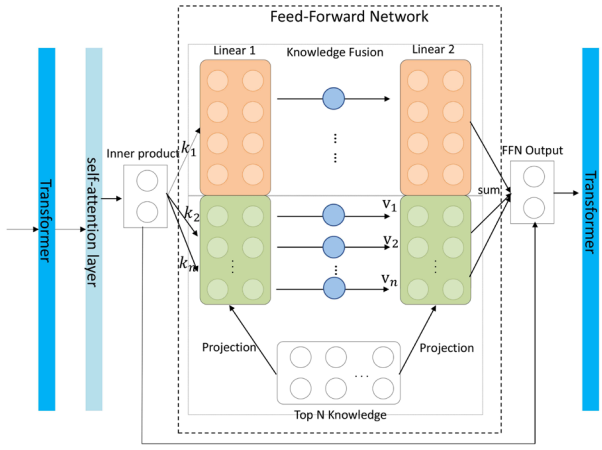


Fig. 1. Knowledge injection process in Kformer.Input vectors are multiplied by each knowledge vector. The final output of the feed-forward network is the sum of the original linear layers' output and the knowledge fusion's output.

The Transformer layer is shown in Figure1, where knowledge fusion is performed. The feed-forward network in each Transformer layer consists of two linear transforms with a GeLU activation in the middle. Assuming that the final attention output of layer l is H^l, the outputs of the two linear layers are:

$$FFN(H^l) = f(H^l \cdot K^l) \cdot V^l \tag{4}$$

where K,$V \in \mathbb{R}^{d_m \times d}$ is the parameter matrix of the first and second linear layers, and f represents the non-linear function. d_m is the intermediate layer size of the Transformer, and d is the hidden layer size. After retrieval, Suppose we get the top N knowledge documents $\mathbf{k} \in \mathbb{R}^{\mathbf{d_n} \times \mathbf{d}}$. Through Knowledge Embedding, we get each knowledge as $k_1, k_2, \ldots, k_N \in \mathbb{R}^{d_n}$. To inject knowledge into a specific layer l, we need to map the knowledge to the corresponding vector space. Here, for each layer l, we use two different linear layers for knowledge

mapping. W^K and W^V represent the weights of the two linear layers ($W^K, W^V \in \mathbb{R}^{d\times d}$). The two matrices W^K and W^V are randomly initialized and will be updated during fine-tuning.

$$\phi_k^l = Pr_k\mathbf{k} = W_k^l \cdot \mathbf{k} \tag{5}$$

$$\phi_v^l = Pr_v\mathbf{k} = W_v^l \cdot \mathbf{k} \tag{6}$$

After projection, ϕ_k^l and ϕ_v^l are injected into the corresponding$\mathbf{K}^l$ and $\mathbf{V}^l$, We extend the FFN by concatenating the projection knowledge to the end of the linear layer and obtain the extended $\mathbf{K}_E^l, \mathbf{V}_E^l \in \mathbb{R}^{(d_m+d_n)\times d}$. Therefore, after injection, the calculation of FFN can be described as:

$$FFN(H^l) = f(H^l \cdot \mathbf{K}_E^l) \cdot V^l = f(H^l \cdot [\phi_k^l : \mathbf{K}^l]) \cdot [\phi_v^l : \mathbf{V}^l] \tag{7}$$

The model activates knowledge relevant to the access control policy and injects it through the knowledge fusion part. Next, the collected information will be processed and aggregated by the following Transformer layers.

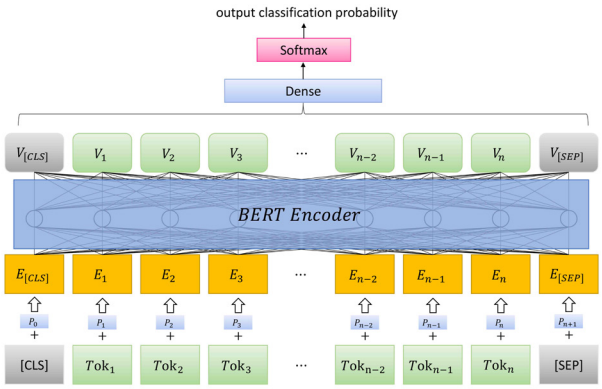


Fig. 2. Bert model architecture diagram. The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

We use Bert [19] for subsequent processing. The model architecture is shown in Figure2. A Bert model is used at the word level, and the word flattening tokens in the sentence are fed into the model. Words are [CLS] preprocessed by the preprocessing module, and the same token vocabulary is used in Bert to obtain word-patched tokens. We keep two special tokens [CLS] and [SEP] at the beginning and end of sentences, respectively, like the Bert word-level module. The first token of each sentence is [CLS], and its corresponding hidden state is considered to represent the aggregate of the entire sentence. [SEP] is at the end of a sentence and is important to distinguish sentences. We omit segment embeddings and keep positional encodings. Therefore, for a given token i, the input embedding E_i is constructed by concatenating the token embedding Tok_i and the position encoding vector P_i. We simply apply a dense layer with a $softmax$ function to output the classification probabilities for the final labels.

C. A Fine-grained dynamic access control method based on Kformer

The terminal is the subject of the permission request, the accessed network resource is the object, and the owner of the resource is the user. Access rules can be set to restrict the user's access to the resource. The terminal makes a resource access request, and the system judges whether to grant the current user access rights according to the access control rules. When the terminal environment changes, it is necessary to evaluate it to give the corresponding permissions dynamically.

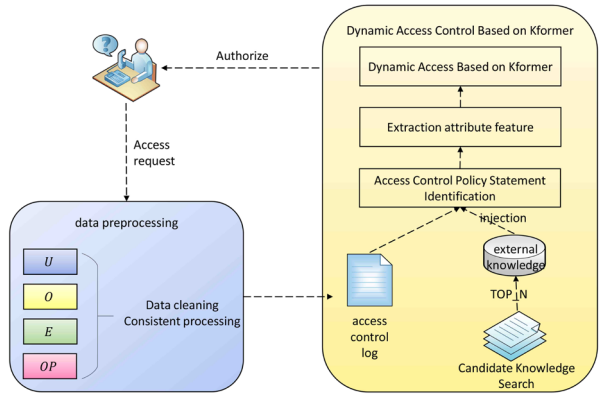


Fig. 3. Kformer-based fine-grained dynamic access control.

After the Kformer network structure is trained according to the above description, it can be used for dynamic and fine-grained access control authorization.As shown in Figure3 The main steps of the fine-grained dynamic access control authorization method based on Kformer are as follows:

Step 1: Data preprocessing: Each access control log record contains information such as the subject user, object resource, action, and operation execution result corresponding to the record. First, perform consistent processing on duplicate and conflicting log data in the log. Remove redundant duplicate log records that exist in the log. Consistently process log records with inconsistent operations according to the time dimension, delete other conflicting log records, and build a globally consistent set of log records.

Step 2: Using the sparse searcher Elasticsearch, we combine attributes and authorization decisions as a query sent to the search engine for each policy. We select the sentence with the highest score as candidate knowledge from the results returned by the search engine.

Step 3: Each Transformer block contains two important modules: multi-head self-attention and feed-forward layers. Multi-head self-attention is an important part that plays the role of message passing between tokens [18]. In the self-attention module, input tags interact and determine what they should pay more attention to. In this part, we inject knowledge into the self-attention module, and the fusion calculation is as follows:

$$Attention^l = softmax(\frac{Q^l[\phi_k^l : \mathbf{K}^l]^T}{\sqrt{d}})[\phi_v^l : \mathbf{V}^l] \tag{8}$$

Knowledge is connected in the K and V parts of self-attention.

Step 4: BERT uses Transformers as the feature extractor, mainly focusing on the part of the Encoder. First, the irregular text is transformed to improve the performance and robustness of the model. The feature extraction is performed on the input fused text, and finally, the extracted text is extracted. The obtained attribute features are sequenced, and the features extracted in the previous step can be reshaped into sequences (Batch size, Seq length, Embedding size).

Step 5: Input the obtained result into the final fully connected layer, and classify it through the $softmax$ function to obtain the final authorization result.

IV. EXPERIMENTS

We have implemented a prototype of the method proposed in Section 3. In this section, we present our experimental evaluation.

A. Experimental Setting

To verify the effectiveness of this method, simulation experiments are performed based on derived from the information system of a provincial power grid company of State Grid Co., Ltd.. The experimental environment is as follows: the operating system is Win10 64 bit, the CPU is Intel(R) Core(TM) i5-6300HQ@ 2.5 GHz, and the GPU is GeForce GTX 850 M, the memory size is 16 GB, and the Python version is 3.6. The metrics used to address our research question are then given, first defining the following variables:

1) True class TP (True Positive) indicates that the policy that allows authorization is identified as the number of policies allowed.

2) The false-positive class FP (False Positive) indicates that the policy that refuses authorization is identified as the number of allowed policies.

3) The False Negative class FN (False Negative) indicates the number of policies that allow authorities to be identified as the number of denied policies.

4) The true negative class TN (true Negative) indicates the number of policies that recognize the policies that refuse authorization as rejected.

We use accuracy to measure the performance of the Kformer model. In addition to $Accuracy$, it also includes $Precision$, $Recall$ and $F1$. Therefore, the definitions of the three indicators are described as follows.

1) The calculation expression of the accuracy rate is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{9}$$

It represents the ratio of all correctly judged samples to the entirety. The higher the accuracy, the better the algorithm effect.

2) The recall calculation expression is:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

It indicates how many positive examples in the original sample are correctly predicted. The higher the recall rate, the better the algorithm effect.

3) The F1 calculation expression is:

$$F1 = \frac{2Precision \times Recall}{Precision + Recall} \tag{11}$$

F1 is an indicator used in statistics to measure the accuracy of a binary classification model. It takes into account the accuracy and recall of the classification model simultaneously. It is a weighted average of the accuracy and recall of the model. The maximum value is 1, and the minimum value is 0. The larger the value, the better the model.

B. Dataset

The data of this experiment are mainly derived from the information system of a provincial power grid company of State Grid Co., Ltd. First of all, the current assets of the power grid are analyzed, and there are currently 30 kinds of Internet of Things business systems and 131 kinds of equipment terminals in the company's management information region, totaling 6.6983 million units. The company/service is mainly the traditional information service of the State Grid Co., Ltd., mainly involving vehicle management, infrastructure projects, storage materials, electric vehicles, power payment, online monitoring of transmission and distribution and transformation lines, online monitoring of power quality, electricity information collection, power supply voltage collection, power inspection/emergency repair, distribution automation, etc. The data source used in the experiment is the network traffic data collected by the information system from 2019.03 to 2019.05, with about 97,000 messages, which mainly come from two ports, one is the upstream message of business requests sent by the terminal to the master station (port 5200). The other type is the downstream message of business requests sent by the master to the terminal (port 5100). The corresponding business types of data messages include power information collection, remote intelligent payment, remote fee control power outage, and power load control.

C. Experimental results

Each access control log record contains information such as the subject user, object resource, action, and operation execution result corresponding to the record. First, we cleaned and consistently processed the duplicate and conflicting log data in the log and then used the sparse searcher Elasticsearch to select the sentence with the highest score from the results returned by the search engine as candidate knowledge and performed it in the feed-forward layer of the Transformer. Knowledge fusion injects knowledge into the self-attention module; uses BERT as a feature extractor, converts the irregular text, performs feature extraction on the input fused text, and performs feature extraction on the extracted attributes. Sequence modeling, input the obtained results into the final fully connected layer and classify through the softmax function to obtain the final authorization result.

We adjusted the number of candidates' knowledge and listed the results in Figure4. The extracted information has both positive and negative effects on the model. If we retrieve less information, the model will not be able to solve the problem

TABLE I
MODEL COMPARISON EXPERIMENTAL RESULTS

Metrics	Transformer	BERT	RoBERTa	Kformer
Accuracy	79.22%	82.23%	84.65%	87.73%
Recall	90.73%	91.35%	93.13%	92.27%
F1-score	0.846	0.866	0.887	0.899
Running Time (s)	90.3	88.58	80.6	86.4

due to a lack of sufficient evidence. However, if we retrieve too much information, the model will suffer from knowledge noise and make wrong predictions. In the figure, when the retrieval information exceeds the first 10 sentences, the performance on our datesets drops. Figure4 shows the impact of different TOP_Ns on model accuracy in the knowledge fusion stage on our datasets.

Fig. 4. Influence of different TOP_N on model accuracy.

We also conducted comparative experiments with the other three models to evaluate the model's performance. The models are as follows:

Transformer [20]: mainly divided into Encoder and decoder. The basic principle is to input a sequence, process it through the Encoder, consider the correlation of the information before and after the whole sequence with the self-attention mechanism, and then input it into the decoder. The decoder will process the input, then classify it through a softmax, and output the final result.

Bert [19]: it is an algorithm proposed by some researchers of Google. Pre-training is carried out in a large-scale corpus, and then fine-tuning is carried out in downstream tasks. The basic model of pre-training is still the transformer model. Bert uses three ways of embedding accumulation in the embedding layer and proposes a two-way language model.

Roberta [21]: it is a method jointly proposed by Facebook and the University of Washington. By adjusting some parameters and step length of Bert, the SOTA effect is achieved in multiple tasks. Roberta's main experimental tuning aspects are as follows: Increase the training data set, batch and epoch. Replace static masked LM with dynamic masked LM. Remove the NSP. Replace characters with bytes.

We compared the four dimensions of Accuracy, Recall, F1-score, and running time, respectively. The experimental results are shown in Table I. The variance of these means is between 0.001 and 0.1.

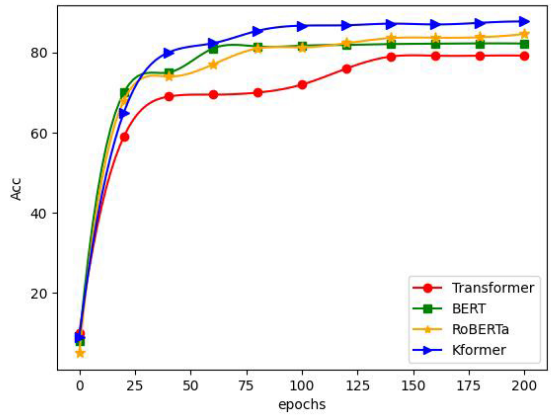


Fig. 5. Accuracy of different models.

As can be seen from TableI, Kformer has the highest accuracy of the two data sets, and the average accuracy in large real data sets can reach 87.73%. Considering the knowledge fusion in Transformer feed-forward layer, Kformer runs slowly. As can be seen from Figure5, these four models all perform well on this dataset. However, due to the injection of external knowledge, KFomer's performance is more stable than other models and has higher accuracy.

V. CONCLUSION

This paper proposes a Kformer-based fine-grained dynamic access control method to automate authorization management tasks. We use Kformer to filter and integrate external knowledge through the feed-forward layer in Transformer. Then, we use BERT to perform feature extraction on the input fused text and extract the implied attribute-authority relationship from log records and external documents. Finally, we perform sequence modeling on the extracted attribute features and input the obtained results. The final authorization result is obtained by classification through the softmax function in the final fully connected layer. The authorization management of the subject's access to the object is dynamically completed. Finally, we conducted experiments on the power information system. The experimental results show that KFormer is stable and has high accuracy. At the same time, the model can continuously monitor the user's access behavior and change it in real-time—entity access rights, real-time policy adjustment, and dynamic access authorization. For future work, We plan to extend our framework with an anomaly detection component to detect changes in normal access behaviors (i.e.,unseen behaviors) to maintain accurate and up-to-date access control policies.

ACKNOWLEDGMENT

This work was supported in part by the State Grid Jiangxi Information & Telecommunication Company Project "Research on de boundary security protection technology based on zero trust framework" under Grant 52183520007V.

REFERENCES

[1] C. Wang, J. Chen, Y. Yang, X. Ma, and J. Liu, "Poisoning attacks and countermeasures in intelligent networks: Status quo and prospects," *Digital Communications and Networks*, vol. 8, no. 2, pp. 225–234, 2022, DOI: 10.1016/j.dcan.2021.07.009.

[2] S. Maksuti, M. Zsilak, M. Tauber, and J. Delsing, "Security and autonomic management in system of systems," *Infocommunications Journal: A Publication Of The Scientific Association For Infocommunications (HTE)*, vol. 13, no. 3, pp. 66–75, 2021, DOI: 10.36244/ICJ.2021.3.7.

[3] M. A. Harrison, W. L. Ruzzo, and J. D. Ullman, "Protection in operating systems," *Communications of the ACM*, vol. 19, no. 8, pp. 461–471, 1976, DOI: 10.1145/360303.360333.

[4] R. S. Sandhu, "Lattice-based access control models,"*Computer*, vol. 26, no. 11, pp. 9–19, 1993, https://doi.org/10.1109/2.241422.

[5] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996, DOI: 10.1109/2.485845.

[6] J. K. Lee, J. Park, S. Gregor, and V. Yoon, "Axiomatic theories and improving the relevance of information systems research," *Information Systems Research*, vol. 32, no. 1, pp. 147–171, 2021, DOI: 10.1287/isre.2020.0958.

[7] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role mining-revealing business roles for security administration using data mining technology," in *Proceedings of the eighth ACM symposium on Access control models and technologies*, 2003, pp. 179–186, DOI: 10.1145/775412.775435.

[8] Q. Guo and M. Tripunitara, "The secrecy resilience of access control policies and its application to role mining," in *Proceedings of the 27th ACM on Symposium on Access Control Models and Technologies,* 2022, pp. 115–126, DOI: 10.1145/3532105.3535030.

[9] C. Blundo, S. Cimato, and L. Siniscalchi, "Role mining heuristics for permission-role-usage cardinality constraints," *The Computer Journal*, vol. 65, no. 6, pp. 1386–1411, 2022, DOI: 10.1093/comjnl/bxaa186.

[10] M. Abolfathi, Z. Raghebi, H. Jafarian, and F. Banaei-Kashani, "A scalable role mining approach for large organizations," in *Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics*, 2021, pp. 45–54, DOI: 10.1145/3445970.3451154.

[11] Z. Xuand S. D. Stoller, "Mining attribute-based access control policies," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 533–545, 2014, DOI: 10.1109/TDSC.2014.2369048.

[12] P. Iyer and A. Masoumzadeh, "Mining positive and negative attribute-based access control policy rules," in *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*, 2018, pp. 161–172, DOI: 10.1145/3205977.3205988.

[13] M. W. Sanders and C. Yue, "Mining least privilege attribute based access control policies," in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 404–416, DOI: 10.1145/3359789.3359805.

[14] A. Abu Jabal, E. Bertino, J. Lobo, M. Law, A. Russo, S. Calo, and D. Verma, "Polisma-a framework for learning attribute-based access control policies," in *European Symposium on Research in Computer Security*. Springer, 2020, pp. 523–544, DOI: 10.1007/978-3-030-58951-6_26.

[15] L. Karimi, M. Aldairi, J. Joshi, and M. Abdelhakim, "An automatic attribute based access control policy extraction from access logs," *IEEE Transactions on Dependable and Secure Computing*, 2021, DOI: 10.1109/TDSC.2021.3054331.

[16] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone et al., "Guide to attribute based access control (abac) definition and considerations (draft)," *NIST special publication*, vol. 800, no. 162, pp. 1–54, 2013.

[17] Y. Yao, S. Huang, N. Zhang, L. Dong, F. Wei, and H. Chen, "Kformer: Knowledge injection in transformer feed-forward layers," arXiv preprint *arXiv:2201.05742*, 2022.

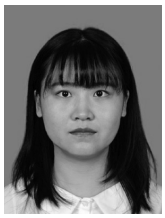
[18] Y. Hao, L. Dong, F. Wei, and K. Xu, "Self-attention attribution: Interpreting information interactions inside transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, 2021, pp. 12 963–12 971, DOI: 10.1609/aaai.v35i14.17533.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding." Minneapolis, Minnesota: Association for Computational Linguistics, jun 2019, pp. 4171–4186, DOI: 10.18653/v1/N19-1423. [Online]. Available: https://aclanthology.org/N19-1423

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint *arXiv:1907.11692*, 2019.

Rixuan Qiu was born in 1994, graduated from North China Electric Power University majoring in computer application technology, his research direction is new power system network and information security.



Xue Xue was born in Langfang, Hebei Province, China, in 1997. She received the B.S. degree in software engineering from Shijiangzhuang University, in 2019. She is currently pursuing the M.S. degree with North China Electric Power University. She has published one articles in Chinese Core Journals. Her research interests include zero trust networks and access control.

Mingliang Chen was born in Ganzhou, Jiangxi Province, China, in 1989. He received the B.Eng. degree and the M.Eng. degree from Nanchang University, Nanchang, China, in 2011 and 2017. He is currently working towards the Ph.D. degree at Xi'an Jiaotong University, Xi'an, China, and a deputy director of Power Dispatch Control Center of State Grid Jiangxi Electric Power Co., Ltd, Nanchang, China. His research interests include power system network security.

Jinkun Zheng was born in Nanchang, Jiangxi Province, China, in 1992. He received the master's degree in engineering from Harbin Institute of Technology in 2016, and has been engaged in digital development and implementation in State Grid Jiangxi Electric Power Co., Ltd. since 2016, his research areas are mainly data mining and analysis in the energy industry.

Sitong Jing is Female, Engineer. She received the master's degree from North China Electric Power University. Research Direction relay protection and scheduling automation of power systems.

Yuancheng Li received the Ph.D. degree from University of Science and Technology of China, Hefei, China, in 2003. From 2004 to 2005, he was a postdoctoral research fellow in the Digital Media Lab, Beihang University, Beijing, China. Since 2005, he has been with the North China Electric Power University, where he is a professor and the Dean of the Institute of Smart Grid and Information Security. From 2009 to 2010, he was a postdoctoral research fellow in the Cyber Security Lab, college of information science and technology of Pennsylvania State University, Pennsylvania, USA.