International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18 (2026) pp. 1-15

https://doi.org/10.70917/ijcisim-2026-0367
Article

Research on Cybersecurity Threat Prediction and Prevention Technology for Intelligent Housing Combined with Edge Computing

Guo li [1,*] and Minghua Wang [2]

[1] College of Intelligent Manufacturing and Electrical Engineering, Nanyang Normal University, Nanyang, Henan, 473000, China
[2] Shandong Gete Aviation Technology Co., Ltd., Jinan, Shandong, 250000, China
* Correspondence author: airforce1980@126.com

Abstract: As an important application scenario for the development of Internet of Things (IoT) technology, smart housing is facing unprecedented challenges in its network security. The traditional centralized security protection architecture in the face of large-scale distributed smart devices have problems such as high latency, slow response, etc., and it is difficult to meet the real-time threat detection and protection needs. Aiming at the problems of insufficient real-time threat detection and limited protection effect of smart housing network security, this paper proposes a smart housing network security threat prediction and prevention technology combined with edge computing. First of all, the distributed security monitoring architecture based on edge computing nodes is constructed, the K-means clustering algorithm is used for network traffic anomaly detection, the support vector machine is used to achieve threat intrusion identification, and the dynamic heterogeneous redundancy adaptive defense mechanism is established; then, the multilayer threat detection model is designed, and the behavioral anomaly analysis and threat prediction are combined with machine learning algorithms; finally, the edge infrastructure is secured and trusted through the edge, data security and network security three-layer protection system to build a complete security prevention framework. The experimental results show that the data processing rate of the system reaches 2350 items/s, the mean value of threat detection error is 0.396, the monthly average defense rate reaches 96.262%, and the path interception probability is as low as 0.935.The research results validate the effectiveness of the edge computing technology in the prediction and prevention of network security threats in smart housing, and provide new technical paths for the construction of efficient real-time security protection system of smart housing. The results verify the effectiveness of edge computing technology in the prediction and prevention of smart housing network security threats, and provide a new technical path for building an efficient real-time smart housing security protection system.

Keywords: edge computing; smart housing; network security; threat prediction; prevention technology; anomaly detection

1. Introduction

In recent years, with the rapid rise of technologies such as big data, cloud computing and mobile Internet, the problem of cyber security has become increasingly prominent [1]. Among them, although the emergence of smart housing fits the residential needs of improved customers and provides options for the digital and intelligent needs of improved customers, the massive integration of the Internet of

Things (IoT) and cloud computing technologies has also led to cybersecurity threats to smart housing [2-4]. Cloud computing applied in smart housing design provides powerful computing resources for big data processing and analysis, while IoT enables various types of physical devices to connect and communicate with each other [5-6]. However, behind this technological progress also hides security threats that cannot be ignored, and security problems such as network attacks, data leakage, and malware appear frequently, seriously threatening personal privacy and network security [7-9].

Since cloud computing integrates a large amount of user data, once the attacker breaks through the defense line, it will lead to large-scale data leakage or system paralysis [10-11]. Security threats in the IoT environment are even more complex, as there are a wide variety of IoT devices and most of them do not have strong security protection capabilities, which puts the whole IoT system at risk of being compromised [12-13]. Attackers can utilize the relevant devices to launch malicious attacks, such as spyware and network viruses, which bring great losses to smart residential users [14]. Based on this, it is extremely necessary to design an intelligent cybersecurity threat perception fusion model to complete the analysis and prediction of the cybersecurity posture and formulate precautionary measures to enhance the cybersecurity level of smart housing [15-17].

Currently, the rapid development of IoT technology has promoted the widespread application of smart housing systems, and the number of smart home devices has grown exponentially, forming a complex heterogeneous network environment. These smart devices are usually characterized by limited computing power and weak security protection, making smart housing networks face diverse security threats. Traditional network security protection mainly relies on a centralized security management system, an architecture that often suffers from bandwidth bottlenecks and latency problems when dealing with massive data generated by large-scale distributed devices, making it difficult to achieve real-time response to security threats. At the same time, security threats in the smart housing environment present new features such as diversified means of attack, strong concealment, and fast propagation, including device hijacking, data theft, denial-of-service attacks, and other forms, posing a serious threat to user privacy and property security. Most of the existing security protection technologies are based on static rules or simple anomaly detection algorithms, lacking the ability to analyze the depth of complex attacks and performing poorly in the face of zero-day attacks and advanced persistent threats. In addition, the dynamics and complexity of smart housing networks make it difficult for traditional security assessment methods to accurately assess the overall security posture of the system and lack an effective threat prediction mechanism.

Based on the above problems, this study proposes to introduce edge computing technology into the security protection system of smart housing network, which realizes near detection and fast response to threats by deploying smart computing nodes at the edge of the network. The study first analyzes the characteristics and propagation mechanism of intelligent housing network security threats, and establishes a distributed security monitoring architecture based on edge computing; then it designs multi-level threat detection algorithms, including traffic anomaly detection based on clustering analysis, behavioral pattern recognition based on machine learning, and threat prediction mechanism based on statistical models; then it constructs a dynamic heterogeneous redundant adaptive defense system, and improves the system's performance through diversified protective Then we construct a dynamic heterogeneous redundant adaptive defense system to improve the security protection capability of the system through diversified protection strategies and dynamic scheduling mechanism; finally, we design a complete experimental validation scheme to verify the effectiveness and practicability of the proposed methodology through simulation tests and actual deployment.

2. Forecasting and preventing cybersecurity threats in smart housing

2.1. Edge computing-based cybersecurity threat prediction for smart housing

2.1.1. Edge computing node deployment and data preprocessing mechanisms

Figure 1 shows the intelligent housing security network architecture, where reasonably deployed edge computing (Mec) nodes are able to complete data collection, preliminary processing and anomaly detection in close proximity to the data source, which reduces the latency of data transmission and improves the system response speed and real-time security monitoring [18]. The data preprocessing mechanism is designed to clean, screen and normalize the collected raw data to reduce redundant data and ensure the accuracy and efficiency of subsequent analysis. Through the intelligent processing of the edge nodes, the potential risks and abnormal behaviors in the operation of the intelligent housing network can be effectively excavated, thus providing data support for the subsequent network security protection and emergency response. Edge computing nodes usually adopt distributed processing

architecture to ensure that the security monitoring needs in different regions of intelligent housing are met, and that certain security analysis tasks, such as traffic anomaly detection, intrusion behavior identification, etc., can be accomplished independently in the local region. The performance indicators of the nodes, such as processing capacity, bandwidth requirements, response time, etc., need to be dynamically adjusted and optimized according to the actual needs of intelligent housing to achieve efficient real-time network security monitoring.

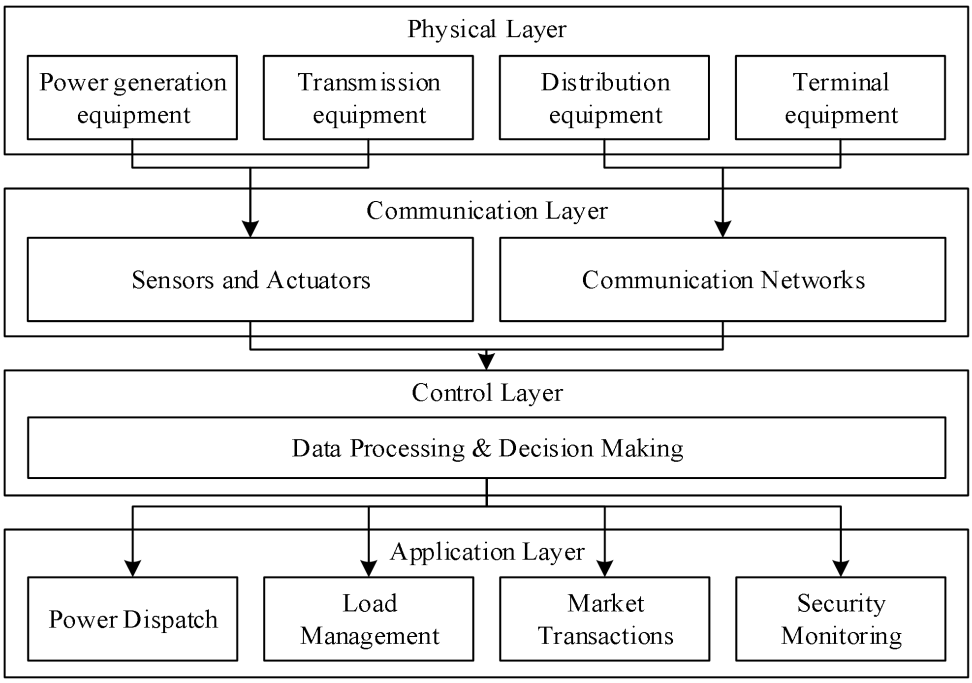

Figure 1. Intelligent housing Security Network Architecture

2.1.2. Network traffic analysis and anomaly detection

Network traffic analysis enables early detection of potential network attacks by characterizing packets and identifying anomalous patterns in traffic. Anomaly detection typically relies on machine learning or statistical methods to identify anomalous behavior that deviates from the model by constructing a behavioral model of normal traffic.

In network traffic analysis, a common anomaly detection method is clustering analysis based on traffic features, of which the K-means clustering algorithm is a widely used technique [19].The goal of the K-means algorithm is to divide the network traffic data into K clusters, such that data points within the clusters are as similar as possible, and data points between the clusters are as different as possible. Assuming a network traffic data set $D = \{x_1, x_2, \cdots, x_n\}$, where each x_n represents a data flow feature vector, the K-means algorithm determines the center of the clusters by minimizing the following objective function:

$$J = \sum_{i=1}^{K} \sum_{x_j \in C_i} \left\| x_j - \mu_i \right\|^2 \tag{1}$$

Where: C_i is the i th cluster, μ_i is the center of the i th cluster, and $\left\| x_j - \mu_i \right\|$ is the Euclidean distance between the data point x_j and the center of the cluster μ_i. By iteratively optimizing this objective function, the K-means algorithm can obtain a set of cluster centers such that each data point is assigned to the cluster closest to it. After completing the clustering, the edge computing node can determine whether a newly arrived data point is an anomalous traffic by detecting whether it belongs to an existing cluster.

2.1.3. Threat Intrusion Detection and Response

The core task of intrusion detection is to extract features from a large amount of network traffic data

and identify anomalous activities by analyzing traffic patterns and behavioral deviations. In order to improve the accuracy and real-time performance of detection, edge computing nodes are able to respond quickly to network intrusion events, reduce data transmission delays, and implement precise defense strategies through local data processing and analysis.

Intrusion detection methods based on Support Vector Machines (SVM) have become an effective technical tool. Suppose the network traffic dataset is $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$, where x_n is the feature vector of the n th data stream, and $y_i \in \{-1, 1\}$ denotes the label of that data stream (where 1 means normal and -1 means abnormal). The goal of the support vector machine is to maximize the classification interval by constructing a decision boundary to classify the data into two classes, thus enabling the detection of abnormal traffic.The optimization problem for SVM can be expressed as:

$$\min_{w,b} \frac{1}{2} w^2 \tag{2}$$

At the same time, the constraints are satisfied:

$$y_i (w \cdot x_i + b) \geq 1, \forall i = 1, 2, \cdots, n \tag{3}$$

Where: w is the normal vector of the hyperplane, b is the bias term, and the goal of the SVM is to make the hyperplane effective in separating normal traffic from abnormal traffic by minimizing the objective function. The support vector machine obtains a classification model by solving this optimization problem and achieves real-time intrusion detection by predicting new data points. When a new data stream x_{new} arrives, the SVM model determines whether it is an abnormal flow by calculating the distance of this data stream from the decision hyperplane:

$$f(x_{new}) = w \cdot x_{new} + b \tag{4}$$

If $f(x_{new}) \geq 1$, the stream is considered normal, if $f(x_{new}) < 1$, the stream is marked as abnormal and the response mechanism is triggered.

2.2. Smart Housing Cybersecurity Threat Detection with Edge Computing

2.2.1. Behavior-based threat detection

Behavior-based threat detection in an intelligent edge computing environment involves monitoring the activities of edge devices, applications, and users to identify suspicious behaviors indicative of security vulnerabilities. The main advantage of behavior-based threat detection is its ability to detect previously unknown threats and circumvent zero-day attacks with traditional signature-based detection methods. By establishing baseline behavioral profiles for edge devices and applications, security systems can detect anomalies such as unusual network traffic patterns, unauthorized access attempts and unusual resource utilization. Behavior-based threat detection relies on advanced analytics, including statistical analysis, machine learning and artificial intelligence, to identify and correlate anomalous behavior across multiple data sources. These techniques enable security systems to distinguish between benign deviations and genuine security threats, thereby minimizing false positives and missed alarms.

2.2.2. Anomaly detection

Anomaly detection algorithms analyze a variety of data sources, including network traffic, system logs, and sensor readings, to identify patterns that significantly deviate from historical norms. The key challenge of anomaly detection in smart edge computing environments is to distinguish between legitimate anomalies and false alarms caused by legitimate changes in system behavior. To address this challenge, anomaly detection systems utilize advanced statistical models, machine learning algorithms, and integration techniques to distinguish benign anomalies from malicious ones. In addition, anomaly detection can be combined with other security controls, such as access control policies, endpoint security measures and intrusion detection systems, to provide a multi-layered defense against security threats. By integrating anomaly detection into the broader security infrastructure, managers can enhance their ability to detect and respond to security events in real time.

2.2.3. Machine Learning for Threat Detection

Machine learning algorithms can be trained on historical data to recognize known attack patterns and anomalies and adapt to changing threats in real time. Machine learning techniques such as supervised learning, unsupervised learning, and reinforcement learning are applied to a variety of security tasks including malware detection, intrusion detection, and anomaly detection in an intelligent edge computing environment. Supervised learning algorithms are trained on labeled datasets containing examples of benign and malicious behavior to learn to correctly classify new instances. These algorithms can be used to develop predictive models for detecting known threats and malware variants based on their characteristics. On the other hand, unsupervised learning algorithms analyze unlabeled data to identify patterns and anomalies without prior knowledge of the expected results. These algorithms are particularly useful for detecting new threats and zero-day attacks that evade signature-based detection methods. Reinforcement learning algorithms learn to make decisions and take action based on environmental feedback, enabling security systems to adapt and improve over time. By continuously learning from experience and adapting behavior to address changing threats, reinforcement learning-based threat detection systems can effectively mitigate emerging security risks.

3. Edge computing-based cybersecurity for smart housing

3.1. Adaptive Security Defense for Edge Computing Endpoints

3.1.1. Security defense modeling

This study combines adaptive defense and dynamic heterogeneous redundancy (DHR) concepts to design a dynamic heterogeneous redundancy defense system as shown in Fig. 2 [20]. By expanding heterogeneous components, adopting dynamic scheduling and setting heterogeneous reserves, the component-level defense diversity is enhanced, and the system security is greatly improved. When attacked, dynamic scheduling can quickly replace the damaged part, destroy the attack conditions, avoid the repetition of similar attacks, while the periodic changes in the system state increase the difficulty of attacker detection, and strengthen the defense effect.

3.1.2. Adaptive defense mechanism establishment

The abstract structure for the security analysis of the dynamic heterogeneous redundant adaptive defense mechanism of the edge computing terminal is as follows: component a represents the attacker, component i represents the input agent module, component logic P represents the set of heterogeneous redundant executables $\left(P_1, P_2, \cdots, P_n\right)$, component o represents the voter, and constitutes the adaptive defense boundary of the system as shown in Figure 3.

The constructed adaptive defense system is based on the following premise: in the security evaluation, no matter what kind of attack is faced, there are always enough diverse execution units to form an adaptive defense structure, and its effectiveness is not affected by the heterogeneity of hardware and software.

Let $T_{dynamic}$ be the update frequency of the adaptive module in the defense system, that is, the period of dynamic replacement of input agents, execution units and decision-makers, T_{attack} represents the time it takes for the attacker to penetrate from one unit to the next unit, reflecting the difficulty of the attack action, P_h is defined as the probability that each execution unit shows different response characteristics when encountering a specific attack, and P_{ij} is the probability that the attacker can successfully penetrate from i units to j units in a non-dynamic and non-heterogeneous static framework.

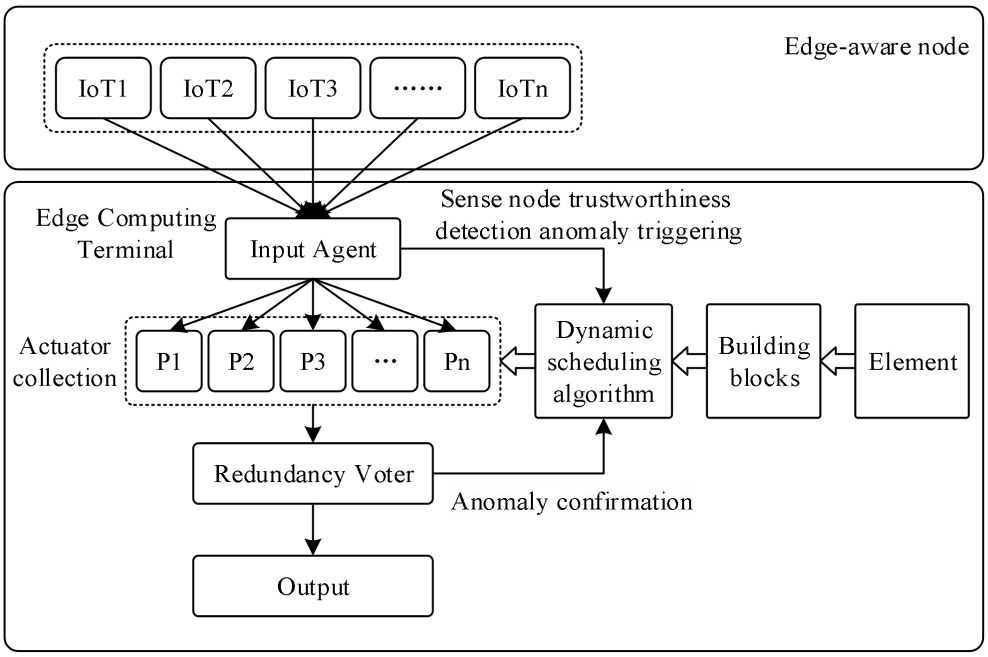

Figure 2. Dynamic heterogeneous redundancy adaptive defense of edge computing terminals

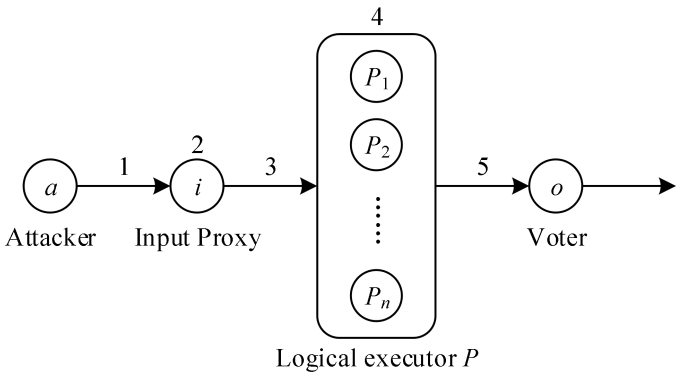

Figure 3. Analysis of Adaptive Defense Security for Edge Computing Terminals

(1) 1 Redundancy Level Adaptive Defense Mechanism Security Analysis

When the adaptive defense architecture is configured with 1 redundancy level, an attacker tries to penetrate the system from a component. Denote P_1, P_2, P_3, P_4, P_5 as the penetration rate from a to agent i, the persistence rate to stay in i, the jump rate from i to logic P, the rate to stay in logic P, and the progression rate from logic P to voter o, in that order.

The probability that an attacker can effectively intrude into component i from component a is expressed as:

$$P_1 = P_{(a,i)} \times \left(1 - P_h\right)^{T_{attack}/T_{dynamic}} \tag{5}$$

Once an attacker has penetrated into component i, he or she is able to launch a number of attacks on entity P equal to the ratio of $T_{dynamic}$ to T_{attack} in each dynamic transition cycle. Thus, the probability of successful intrusion from component i to component P in a single dynamic transition cycle $T_{dynamic}$ can be expressed as:

$$1 - \left(1 - P_{(i,p)}\right)^{T_{dynamic}/T_{attack}} \tag{6}$$

The attacker will remain in component i for the duration of time period T_{attack} under the following two scenarios:

First, if the attacker's intrusion attempt on component P from component i is unsuccessful, and at the same time the dynamic changes experienced by component i do not interfere with the attacker's actions, the probability that the attacker stays on component i in this condition is:

$$\left(1-P_{(i,p)}\right)^{T_{dynamic}/T_{attack}} \times \left(1-P_h\right)^{T_{attack}/T_{dynamic}} \tag{7}$$

Second, even if the attacker's penetration attack from component i to component P is successful, if the dynamic change of component i does not affect the attacker, yet the dynamic change of component P hinders the subsequent effect of the attack, in this scenario, the probability that the attacker still stays on component i is:

$$\left(1-\left(1-P_{(i,p)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(\left(1-P_h\right)^{\frac{T_{attack}}{T_{dynamic}}}\right)\times\left(1-\left(1-P_h\right)^{\frac{T_{attack}}{T_{dynamic}}}\right) \tag{8}$$

Then the probability that the attacker continues to stay in component i is:

$$P_2=\left(1-P_h\right)^{T_{attack}/T_{dynamic}}-\left(1-\left(1-P_{(i,p)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(1-P_h\right)^{\frac{2T_{attack}}{T_{dynamic}}} \tag{9}$$

Provided that the dynamic transformations of component i and component P do not affect the attacker's actions, the probability of successful penetration of the attacker from component i to component P can be expressed as $\left(1-\left(1-P_{(i,p)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)$, then P_3 is:

$$P_3=\left(1-\left(1-P_{(i,p)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(1-P_h\right)^{\frac{2T_{attack}}{T_{dynamic}}} \tag{10}$$

The probability P_p that an attacker successfully compromises component P by component a:

$$P_p=P_1\times\left(P_2^0P_2^1+\cdots+P_2^n\right)\times P_3 \tag{11}$$

The expression for P_4, P_5 is as follows:

$$P_4=\left(1-P_h\right)^{T_{attack}/T_{dynamic}}-\left(1-\left(1-P_{(p,o)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(1-P_h\right)^{\frac{2T_{attack}}{T_{dynamic}}} \tag{12}$$

$$P_5=\left(1-\left(1-P_{(p,o)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(1-P_h\right)^{\frac{2T_{attack}}{T_{dynamic}}} \tag{13}$$

The probability P_a that an attacker successfully compromises component o by component a:

$$\begin{aligned}P_o&=P_1\times\left(P_2^0+P_2^1+\cdots+P_2^n\right)\times P_3\times\left(P_4^0+P_4^1+\cdots+P_4^n\right)\times P_5\\&=\frac{1}{1-P2}\times\frac{1}{1-P4}\times P_{(a,i)}\times\left(1-\left(1-P_{(i,p)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\\&\times\left(1-\left(1-P_{(p,o)}\right)^{\frac{T_{dynamic}}{T_{attack}}}\right)\times\left(1-P_h\right)^{\frac{5T_{attack}}{T_{dynamic}}}\end{aligned} \tag{14}$$

(2) Security analysis of 3-redundancy adaptive defense mechanism

When the adaptive defense mechanism adopts 3 redundancy, the probability that an attacker successfully compromises component o from component a is:

$$
\begin{aligned}
P_o &= P_1 \times \left(P_2^0 P_2^1 + \cdots + P_2^n \right) \times P_3 \times \left(P_4^0 + P_4^1 + \cdots + P_4^n \right) \times P_5 \\
&= \frac{1}{1-P2} \times \frac{1}{1-P4} \times P_{(a,i)} \times \left(1 - \left(1 - P_{(i,p)}^3 \right)^{\frac{T_{dynamic}}{T_{attack}}} \right) \\
&\times \left(1 - \left(1 - P_{(p,o)}^3 \right)^{\frac{T_{dynamic}}{T_{attack}}} \right) \times \left(1 - P_h \right)^{4+\frac{9T_{attack}}{T_{dynamic}}}
\end{aligned} \tag{15}
$$

3.2. Analysis of Edge Computing Security Protection Technology

3.2.1. Secure and trusted edge infrastructure

Edge infrastructure security is the basic guarantee for edge computing, so it is necessary to ensure that the edge infrastructure is secure and trustworthy in the process of startup, operation, operation, etc., and to build a chain of trust for the edge infrastructure, and wherever the chain of trust is connected, security can be protected there.

Edge infrastructure security covers device security, hardware security, virtualization security and operating system security in the entire process from startup to operation. For example, the first step is to check and verify the integrity of the systems and applications in the edge infrastructure, safeguard the integrity of the systems and applications, ensure that the edge nodes are running in the expected state, be able to effectively identify, differentiate and identify each edge node, realize edge node management, task allocation and security policy differentiation management, allow specific devices to access the network and deny access to illegal devices according to the security policy It can effectively avoid sinkhole attack and witch attack, which are attack modes initiated by introducing forged devices into the network.

3.2.2. Edge data security

Data security is the basis for creating a secure edge computing environment, the fundamental purpose of which is to ensure the confidentiality and integrity of data. In view of the characteristics of the edge computing environment, such as the separation of data ownership and control, and the randomization of storage, it is necessary to focus on solving the problems of data loss, data leakage, and illegal data manipulation.

Data security focuses on technologies such as data confidentiality and secure sharing, integrity auditing, and searchable encryption. Existing data confidentiality and secure data sharing solutions are usually implemented using encryption technology, and the regular process is to encrypt the outsourced data in advance by the data owner for processing and uploading operations, and decrypted by the data user when needed, and lightweight cryptographic solutions generated by customization or tailoring need to be provided for resource-constrained edge devices.

3.2.3. Edge network security

The huge number of nodes and complex network topology in the edge computing network lead to an increase in the attack surface, and attackers can easily find a breakthrough to send malicious packets to the edge computing nodes and launch denial-of-service attacks. Strengthening edge network security protection should establish a deep defense system, from security protocols, network domain isolation, network monitoring, network protection, etc., from the inside to the outside to ensure edge network security.

4. Mec in smart housing cybersecurity threats and prevention

4.1. System overview

4.1.1. Design Ideas

At the point of interest in the smart housing, multiple edge computing units are installed to form an edge computing system. Physical location close to each other edge computing read-write connected to the same data concentration unit constitutes the edge computing unit, edge computing unit in the read-write senses the data and sends the data through the network to the convergence center, the data management center through the analysis of the data, the point of interest in the building at this point in time to assess the security situation, the need to intervene in the state of affairs to respond in a timely

and effective manner. The whole system has the functions of label issuance, sensing labels and data management. The overall system structure uses modular design ideas.

4.1.2. Core technologies

In the case that the computer system of the data concentrator in the edge computing unit cannot be connected to the network or there is no NTP service, a computational model for time synchronization of the edge computing unit is established, and a time synchronization method for the edge computing unit based on BP neural network is designed.

4.2. System structure and functionality

4.2.1. System architecture

The system adopts a modular structure with the following main modules: (1) time synchronization module, (2) read/write setup module, (3) tag issuance module, and (4) data processing module.

(1) Time synchronization module

In the case that the computer system of the data concentrator in the edge computing unit has no NTP service, BP neural network is used to realize the time synchronization of each edge computing unit within the edge computing system of multiple edge computing units.

(2) Read/write setting module

The parameters that need to be configured for the operation of the edge computing system of multiple edge computing units include the communication mode, the read-write working parameters and the electronic label working parameters. The read-write setting module provides the configuration means of the above parameters in the form of a dialog box respectively.

(3) Tag issuance module

Tag release module in the form of a dialog box to achieve the user requirements to write data to the electronic tag, and dialog box to achieve the release of the state of the query.

(4) Data processing module

The data processing module realizes the data sensing of each edge computing unit in the system and transmits the data to the data management center, which processes and displays the data.

4.2.2. Main functions

The main functions realized by the system include: time synchronization, read/write settings, data sensing, tag issuance, data storage and data management. The main functions are as follows:

(1) To realize time synchronization of edge computing units without NTP service conditions.

(2) Setting the communication mode and the working status of read-write and electronic tags in the form of dialog box.

(3) Sensing of electronic tags by readers in each edge computing unit within the system.

(4) Issue operation of electronic tags by means of a dialog box.

(5) Storage and management of data perceived by each edge computing unit in the system in the data management system.

5. Experiments and analysis of results

5.1. Experimental environment

In this paper, the training and testing process based on edge computing technology are carried out on Ubuntu system and the algorithms are written using Python language. The hardware environment for the experiments is Intel Core i7-7700 HQ processor, 8GB RAM, GTX1050 graphics card and 16GB RAM.

In addition, in order to validate the effectiveness of the cyber threat assessment methodology in this paper, the Snort 3.0 intrusion detection tool was deployed on the PC server so that the threat information could be recorded and the type of threat could be determined when the cyber threat test was conducted.

5.2. Comparison of Network Security Threat Test Results

5.2.1. Processing rate

In order to prove the effectiveness of the model in this paper, four models, AE, VAE, GAN and

Edge Computing, are utilized to form a network ensemble for model training in the model training phase. Under the same sample training set, with the increase of the number of layers in the network collection, the comparison of the data processing rate of the four models is shown in Fig. 4.

Under the same sample training set, when the number of network collection layers is 14 and the model training reaches Nash equilibrium, the data processing rate of the network collection composed of edge computing reaches about 2350 entries/s, while the data processing rate of the AE-based network collection reaches nearly 3250 entries/s when the number of network collection layers is 20. Although the data processing rate of the latter is higher than that of the former, the ability of AE to reconstruct the raw data is weaker than that of the edge computing network, a conclusion that can be verified by the following comparative experiments.

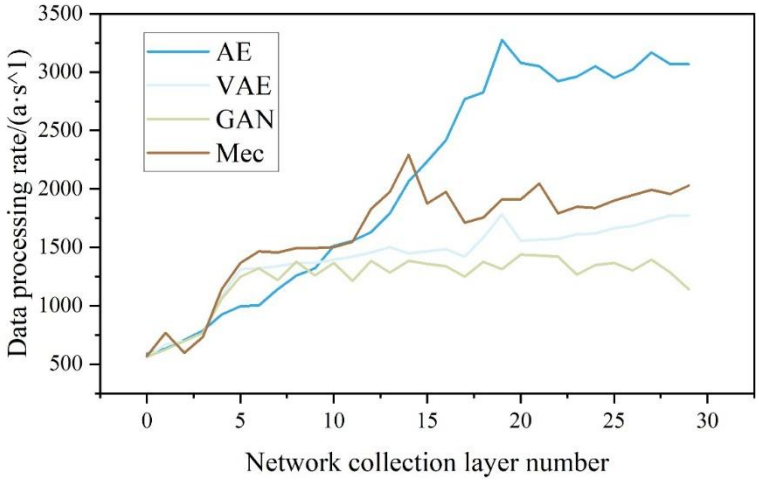

Figure 4. The number of network collection layers and data processing rate

5.2.2. Threat test results

In the threat testing phase, this paper conducts 200 groups of threat testing experiments, and each group of tests is randomly selected from the test dataset with the same number of test samples for testing. Threat tests were conducted using four models, namely, AE, VAE, GAN, and Edge Computing, respectively, and the number of layers in the network collection was 15. The normalized test error values obtained from 10 sets of these threat test experiments ω are shown in Figure 5.

Using the same sample test set, when the number of network collection layers reaches 15, the edge computing-based network threat test model outputs the largest test error value ω compared to the other three models, with the error value interval between [0.26018,0.67964], and the average value of the error for the 10 groups is 0.396, which indicates that it is more prominent in its ability to detect network threats.

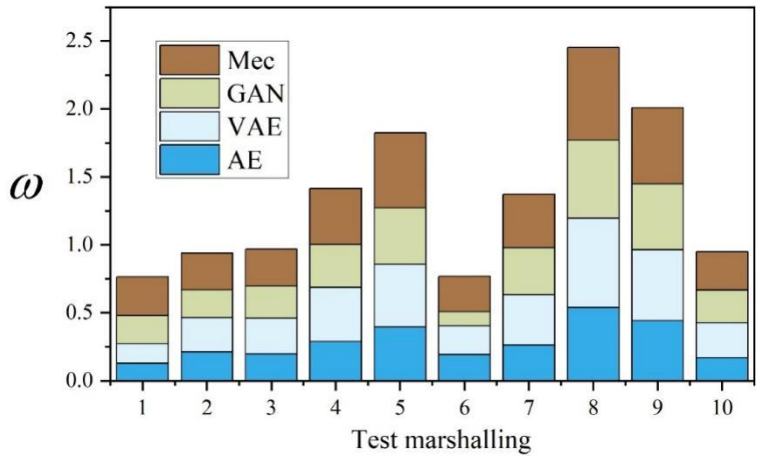

Figure 5. The threat test results of four models

5.2.3. Forecasted security threat posture

In order to assess the overall cyber threat posture, a quantitative assessment of the 2 threat factors that determine the cyber security posture is required. First, the error value ω for each group of tests is obtained through the threat testing process, which is normalized in the [0, 1] interval, and then the threat severity and threat impact levels are determined. The results of the threat severity and threat impact assessments for 10 groups of cybersecurity postures are shown in Table 1.The security threat prediction of the smart housing network using edge computing shows that the threat severity of all numbered groups is in the safe, low-risk, and medium-risk status, the corresponding threat impact degree is no impact and low impact, it is worth noting that the predicted probability of threat occurrence for number 8 is 0.642, and the threat impact degree is high impact.

Table 1. Assessment results of threat severity and threat impact degree

Number	*Threat probability*	*Threat severity*	*Threat influence*
1	0.185	Safety	No effect
2	0.245	Low risk	No effect
3	0.269	Low risk	No effect
4	0.495	Medium hazard	Low effect
5	0.515	Medium hazard	Low effect
6	0.248	Low risk	No effect
7	0.348	Low risk	No effect
8	0.642	High risk	High effect
9	0.569	Medium hazard	Low effect
10	0.348	Low risk	No effect

5.3. Smart Housing Cybersecurity Defense Results

5.3.1. Successful defense rate

In practice, smart housing security teams often need to record and analyze security events to evaluate the actual effectiveness of protection technologies. Table 2 shows the monthly statistics of smart housing security events. For example, Table 2 shows the monthly security event statistics of a smart housing over the past year, including the number of attack detections, the number of false positives, and the percentage of attacks that were successfully blocked. The successful defense rates for smart housing security threat prevention using edge computing methods are all above 90%, with a monthly average defense rate of 96.262%, and the successful defense rate in November reached 100%. By analyzing this data, Smart Housing can assess the overall performance of its security facilities, identify trends and potential risk points, and thus continuously optimize its security strategy.

Through this regular data evaluation, Smart Housing can not only monitor the short-term effectiveness of its protection technologies, but also adjust its security strategy based on trends in the long term to ensure a continuous fight against emerging threats.

Table 2. The monthly statistics of intelligent housing safety events

Month	**Number of attack detection**	**A successful attack**	**False number**	**Successful defense rate**
January	125	120	5	96.000%
February	120	118	2	98.333%
March	155	150	5	96.774%
April	165	155	10	93.939%
May	160	150	10	93.750%
June	170	160	10	94.118%
July	160	150	10	93.750%
August	165	160	5	96.970%
September	175	170	5	97.143%
October	180	175	5	97.222%
November	160	160	0	100.000%
December	175	170	5	97.143%

5.3.2. Route interception probability

The defense interception is implemented for specific network paths, and the number of successful interception and the number of failures on each route are counted to get the interception probability of

each route, and the results of the interception probability calculation for each route are shown in Table 3.

The simulation results show that the five interception routes with the smallest interception probability are L40, L39, L11, L9, and L30, with the interception probability of 0.935, 0.94, 0.951, 0.953, and 0.955, respectively, reflecting the insufficient protection capability of these routes, which is consistent with the results of the top 5 vulnerability values of the library corresponding to the failure of the protection routes derived from the evaluation method in this paper. At the same time, compared with the method of assessing system vulnerability based on intrusion paths used in the simulation, the method proposed in this paper has the advantage of being able to simultaneously derive the weak points of protection, the vulnerability of protection paths and the vulnerability of the system, and it can be improved based on the results of the assessment.

The simulation can also obtain the generality of this paper's assessment method in various scenarios, i.e., for an alternative scenario, it can also be scored and assessed by the indicators of its network structural characteristics, which replaces the modeling complexity and single applicability of using the path-based method.

Table 3. Simulation results of the route

Path	Success number	Failure number	Intercept probability	Path	Success number	Failure number	Intercept probability
L1	15	985	0.985	L22	15	985	0.985
L2	4	996	0.996	L23	10	990	0.99
L3	17	983	0.983	L24	17	983	0.983
L4	18	982	0.982	L25	28	972	0.972
L5	30	970	0.97	L26	35	965	0.965
L6	5	995	0.995	L27	3	997	0.997
L7	7	993	0.993	L28	30	970	0.97
L8	36	964	0.964	L29	20	980	0.98
L9	47	953	0.953	L30	45	955	0.955
L10	35	965	0.965	L31	30	970	0.97
L11	49	951	0.951	L32	24	976	0.976
L12	17	983	0.983	L33	17	983	0.983
L13	15	985	0.985	L34	7	993	0.993
L14	17	983	0.983	L35	5	995	0.995
L15	5	995	0.995	L36	22	978	0.978
L16	16	984	0.984	L37	4	996	0.996
L17	9	991	0.991	L38	5	995	0.995
L18	35	965	0.965	L39	60	940	0.94
L19	2	998	0.998	L40	65	935	0.935
L20	25	975	0.975	L41	30	970	0.97
L21	17	983	0.983				

5.3.3. Effective overhead under the security program

Fig. 6 shows the average task failure rate curves for different security schemes, set in this paper scheme of edge computing, DHR mechanism, restart policy with L=10 and system without security mechanism. The security architecture with edge computing has a lower and stable average task failure rate with a value of about 0.025 compared to the system with restart policy and no security mechanism.The average task failure rate of the system with restart policy and no security mechanism increases at the same rate for the first 10 time slots, while the task failure rate of the restart policy returns to the lowest level after every 10 time slots.

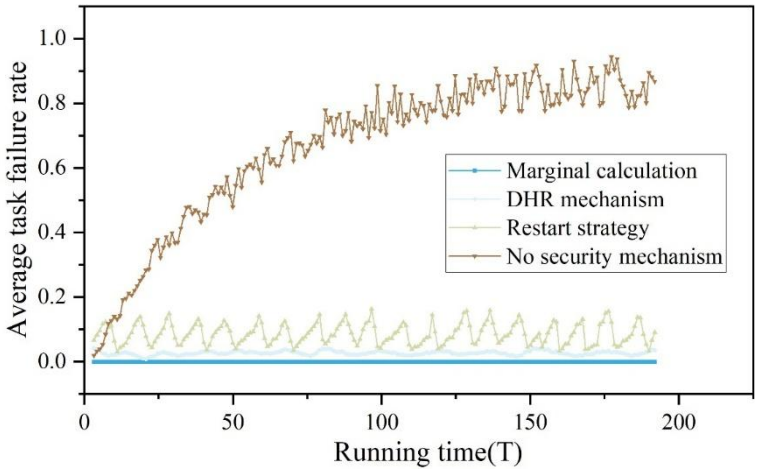

Figure 6. The average task failure rate under different security schemes

Figure 7 shows the average effective overhead curve for different security schemes. The effective overhead of the system is calculated by dividing the detection and maintenance cost by the number of successful tasks. The results show that the edge computing based security mechanism has the lowest average effective overhead, which is around 2 to 2.5. The L=10 restart policy also has a low acceptable average effective overhead.The DHR system performs the same task through multiple heterogeneous ME apps at each time slot, which ensures the minimum number of compromised ME apps and also results in the minimum number of successful tasks, and therefore has a high average effective overhead. The average effective overhead of the system without security mechanism grows rapidly after the system is running, which is due to the lack of security mechanism, the number of compromised ME apps keeps increasing, which leads to a continuous decrease in the success rate of the tasks. The system without security mechanism is breached 10 times in 500 time slots, which shows the discontinuity of the function curve, in the simulation in this paper, the breached system is automatically restored to its initial state.

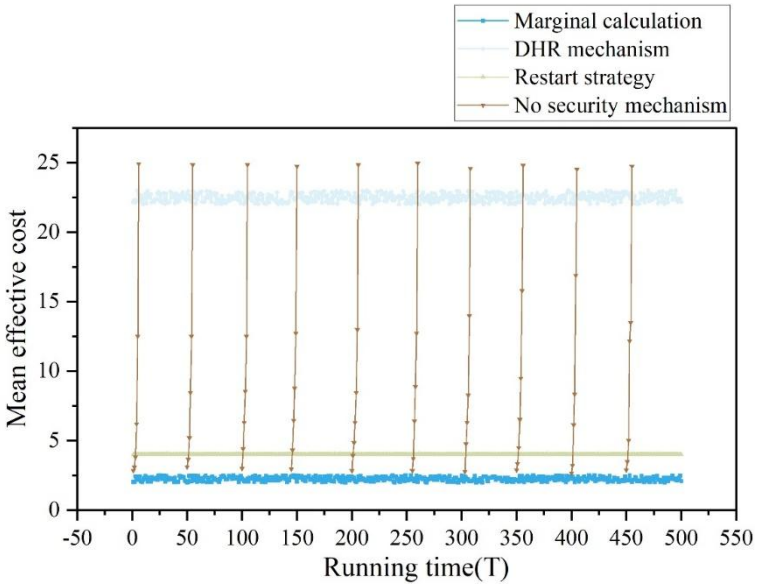

Figure 7. Average effective cost under different security scenarios

6. Conclusion

The effectiveness and practicality of the proposed technical solution is verified through an in-depth study of edge computing in the prediction and prevention of cybersecurity threats in smart housing. The experimental results show that the threat detection system based on edge computing exhibits excellent performance in processing network traffic data, and when the number of network collection layers reaches 15, the system outputs test error values in the range of 0.26018-0.67964, and the error mean value of 10 groups of tests is 0.396, which is significantly better than the traditional models such as AE,

VAE, GAN, etc., and proves that the edge computing has an threat detection, proving the outstanding ability of edge computing in network threat detection. In the practical application of intelligent housing security protection, the constructed defense system shows good protection effect, and through the statistical analysis of 12 months' security events, it is found that the successful defense rate of the system stays above 90%, of which the perfect defense rate of 100% is reached in November, and the overall monthly average defense rate is 96.262%. The evaluation results of path protection capability show that among the 41 tested paths, the lowest interception probability is 0.935, and the interception probability of most paths exceeds 0.95, which effectively guarantees network security. The edge computing security mechanism also performs well in system overhead control, with the average effective overhead maintained between 2-2.5, significantly lower than other security schemes. These data fully prove the superiority of edge computing technology in smart housing network security protection, and provide reliable technical support and practical guidance for the construction of future smart housing security system.

Funding
This research was supported by the Nanyang Normal University Foundation of China (Grant 2024PY011).

References

1. Mallaboyev, N. M., Sharifjanovna, Q. M., Muxammadjon, Q., & Shukurullo, C. (2022, May). Information security issues. In Conference Zone (pp. 241-245).
2. Lata, M., & Kumar, V. (2022). IoT network security in smart homes. Cybersecurity in smart homes: architectures, solutions and technologies, 155-176.
3. Singh, S., Sharma, P. K., & Park, J. H. (2017). SH-SecNet: an enhanced secure network architecture for the diagnosis of security threats in a smart home. Sustainability, 9(4), 513.
4. Farooq, M., & Hassan, M. (2021). IoT smart homes security challenges and solution. International Journal of Security and Networks, 16(4), 235-243.
5. Yassine, A., Singh, S., Hossain, M. S., & Muhammad, G. (2019). IoT big data analytics for smart homes with fog and cloud computing. Future Generation Computer Systems, 91, 563-573.
6. Ray, A. K., & Bagwari, A. (2020, April). IoT based Smart home: Security Aspects and security architecture. In 2020 IEEE 9th international conference on communication systems and network technologies (CSNT) (pp. 218-222). IEEE.
7. Sadeeq, M. M., Abdulkareem, N. M., Zeebaree, S. R., Ahmed, D. M., Sami, A. S., & Zebari, R. R. (2021). IoT and Cloud computing issues, challenges and opportunities: A review. Qubahan Academic Journal, 1(2), 1-7.
8. Edu, J. S., Such, J. M., & Suarez-Tangil, G. (2020). Smart home personal assistants: a security and privacy review. ACM Computing Surveys (CSUR), 53(6), 1-36.
9. Yao, Y., Basdeo, J. R., Mcdonough, O. R., & Wang, Y. (2019). Privacy perceptions and designs of bystanders in smart homes. Proceedings of the ACM on Human-Computer Interaction, 3(CSCW), 1-24.
10. Hammi, B., Zeadally, S., Khatoun, R., & Nebhen, J. (2022). Survey on smart homes: Vulnerabilities, risks, and countermeasures. Computers & Security, 117, 102677.
11. He, J., Ota, K., Dong, M., Yang, L. T., Fan, M., Wang, G., & Yau, S. S. (2017). Customized network security for cloud service. IEEE Transactions on Services Computing, 13(5), 801-814.
12. Geneiatakis, D., Kounelis, I., Neisse, R., Nai-Fovino, I., Steri, G., & Baldini, G. (2017, May). Security and privacy issues for an IoT based smart home. In 2017 40th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1292-1297). IEEE.
13. Abdulla, A. I., Abdulraheem, A. S., Salih, A. A., Sadeeq, M. A., Ahmed, A. J., Ferzor, B. M., ... & Mohammed, S. I. (2020). Internet of things and smart home security. Technol. Rep. Kansai Univ, 62(5), 2465-2476.

14. Shin, D., Sharma, V., Kim, J., Kwon, S., & You, I. (2017). Secure and efficient protocol for route optimization in PMIPv6-based smart home IoT networks. IEEE Access, 5, 11100-11117.
15. Tanwar, S., Patel, P., Patel, K., Tyagi, S., Kumar, N., & Obaidat, M. S. (2017, July). An advanced internet of thing based security alert system for smart home. In 2017 international conference on computer, information and telecommunication systems (CITS) (pp. 25-29). IEEE.
16. Taiwo, O., Ezugwu, A. E., Oyelade, O. N., & Almutairi, M. S. (2022). Enhanced intelligent smart home control and security system based on deep learning model. Wireless communications and mobile computing, 2022(1), 9307961.
17. Umer, M., Sadiq, S., Alhebshi, R. M., Sabir, M. F., Alsubai, S., Al Hejaili, A., ... & Mohamed, A. (2023). IoT based smart home automation using blockchain and deep learning models. PeerJ Computer Science, 9, e1332.
18. Rathore Himmat & Singla Priyanka. (2025). Cyber security in smart home Internet of Things devices: Threat detection and prevention using artificial intelligence. Journal of AI, Robotics & Workplace Automation,3(4),339-349.
19. Yu* Lin,Bai Yujie & Shankar Achyut. (2024). Design of network security monitoring system based on K-means clustering algorithm. Intelligent Decision Technologies,18(4),3105-3118.
20. Shandilya Shishir Kumar,Upadhyay Saket,Kumar Ajit & Nagar Atulya K.. (2022). AI-assisted Computer Network Operations testbed for Nature-Inspired Cyber Security based adaptive defense simulation and analysis. Future Generation Computer Systems,127,297-308.