
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2023 SCPE. Volume 24, Issues 3, pp. 511–520, DOI 10.12694/scpe.v24i3.2162

# VULNERABILITY DETECTION IN COMPUTER NETWORKS USING VIRTUAL REALITY TECHNOLOGY

SONGLIN LIU*

**Abstract.** This paper challenges the time-related challenges inherent in conventional network security detection methodologies. It is achieved by incorporating virtual reality technology into the domain of computer network security detection. The research methodology employs optimization calculations to extract attributes that characterize network security vulnerabilities. Concurrently, the weighting of diverse vulnerability attributes is adjusted using a web crawler, a comprehensive list of injection points, and meticulous analyses of the attacks' genetic characteristics. This collective approach facilitates the exploration of automated network security vulnerability detection within a virtual reality framework. The study's empirical results demonstrate that the detection method proposed within this investigation exhibits a notably reduced delay of 75.33 milliseconds. The respective delays observed in the two conventional methods stand at 290.11 milliseconds and 337.30 milliseconds. The substantial decrease in detection delay validates the effectiveness and efficiency of the devised automated network vulnerability detection approach grounded in virtual reality technology.

**Key words:** Network security vulnerabilities, Virtual reality technology, Detection methods, Optimization calculations, Automated detection

**1. Introduction.** The abnormal network traffic attacks are evolving into more complex and diversified. This anomalous traffic can disrupt the regular functioning of terminal PC systems at a minor scale, while at a critical level, it can induce server immobility and network breakdowns. Consequently, automation of network vulnerability detection serves a dual purpose: upholding the network's robust operation and aiding network administrators in preemptively identifying vulnerabilities for proactive security measures.

Automated tools for detecting security vulnerabilities prove highly effective in safeguarding network integrity. These tools systematically scan the network for vulnerabilities, quickly identifying and generating reports on potential weaknesses. The utilization of such tools substantively elevates network security and curtails the susceptibility to network breaches. Throughout the automated vulnerability detection process, the system treats each node vulnerability as an independent entity, readjusting the vulnerability's weight attribute differentially through calculation and optimization to derive an attack graph. This graph represents the potential for unauthorized entities to penetrate and attack the network system via vulnerabilities, showcasing their varying weight inequalities. The construction of an attack graph entails the aggregation of vulnerability clusters. With the evolution of network technology, the difficulty of network security challenges is simultaneously escalating, encapsulating various issues like network attacks, data leaks, malware incursions, and phishing exploits [19].

Integrating virtual reality technology into network security management involves various domains such as enterprise network platforms, computer programming, and the global network. This integration has generated an excess of research outcomes. For instance, within enterprise network platforms, authentication virtual reality technology is employed to curtail user system access actions. An illustrative instance involves merging fingerprint identification technology with network access rights management, thus transposing the customer's tangible identity into the virtual domain for multifaceted authentication of their distinct physical identity. When challenging network security issues restricting from malicious network viruses, remedies involve utilizing network encryption virtual reality technology and critical management virtual reality technology to counteract these threats [13]. Among the pivotal strategies for network security management lies security vulnerability detection. The convergence of network security vulnerability detection and virtual reality technology remains relatively

*Design and Art Department, Wuxi Institute of Technology, Wuxi, Jiangsu, 214121, China (songlinLiu6@163.com).

limited in the research scope. Nevertheless, considering the efficacy of applying virtual reality technology to network security management, amalgamating these two realms will emerge as a pivotal avenue for future field advancement.

The paper is structured as follows: Section 2 investigates the literature review, providing an overview of existing approaches, their limitations, and the need for more efficient solutions. Section 3 presents the proposed method, detailing the integration of virtual reality technology, optimization calculations, web crawlers, and simulation of attacks for automatic detection. Section 4 presents the results and discussion, showcasing experimental outcomes that validate the effectiveness of the proposed method compared to traditional approaches, highlighting reductions in detection time delays. Finally, Section 5 summarizes the findings in conclusion, emphasizing the potential of virtual reality technology to reshape the landscape of network security vulnerability detection, improving efficiency and minimizing risks.

**2. Literature Review.** Information security vulnerabilities encompass imperceptible conditions that can compromise or harm information systems' confidentiality, integrity, and availability. Users engaging with information can construct autonomous environments through networking, leveraging high-performance computing systems to execute computational tasks from diverse locations. This approach moderates the burden of system management, boosting utilization efficiency. However, inherent security vulnerabilities exist within the virtualization software domain, posing threats that can severely undermine information services' dependability on virtualized environments.

Systems grounded in virtual technology exhibit complexity, and safeguarding information resources proves to be a complex undertaking. Lapses in caution could culminate in the emergence of information security vulnerabilities. Even when comprehensive measures are undertaken to strengthen and support the entire system's information security to the highest level, there remains the certainty that the level of protection and prevention might fall short, thereby exposing information to potential risks [7, 5].

Network attacks stand out as one of the most prevalent network security challenges. Hackers may exploit vulnerabilities, weak passwords, social engineering, and other techniques to infiltrate systems, appropriate, sensitive information, manipulate data, or disrupt operations. The data leakage issue is equally concerning, which can expose users' details like names, addresses, social security numbers, and credit card information. The malware hazard further intensifies network security, encompassing viruses, trojans, worms, system damage, data theft, or surveillance of user activities. Additionally, phishing represents a deception wherein fraudsters manipulate users into exposing personal information or login credentials through email, SMS, and social media.

The network environment represents high openness, sharing, and interactivity, catering to diverse user operational requirements. However, this expansive range of network functionalities and services simultaneously introduces numerous vulnerabilities, imperiling user information security. As computer network runtime increases, the quantity and intricacy of exposed security vulnerabilities tend to magnify. Among these, link connection vulnerabilities arise during network information interactions. The link serves as a pivotal conduit for transmitting computer network information. If subjected to malicious attacks, it could result in issues like information destruction, loss, and network security incidents.

Typical methods for network vulnerability detection involve constructing a sample set model and deriving network variables for simulation experiments. An example is the application of hidden Markov models to shape the network information sample set, thus facilitating the automated refreshment of network security vulnerability detection. However, this approach's attaining the network flow table is intricate, leading to diminished detection efficiency and presenting an efficiency difficulty [6, 1]. An automatic detection method is formulated by extracting dynamic numeric variables and scrutinizing static attack processes by fusing dynamic and static aspects. Notably, this method's assessment has been confined to simulations on open-source software. Consequently, the outcomes of these experiments are partial, casting uncertainty over their practical significance.

**3. Proposed Methodology.**

**3.1. Automatic detection method of network security vulnerabilities under virtual reality technologygn.** At the core of network security vulnerability, technology analyses the program's corresponding source code and identifies vulnerabilities. Numerous latent vulnerabilities exist within a program's source code, which malicious actors could exploit to breach systems, extract information, or disrupt operations.

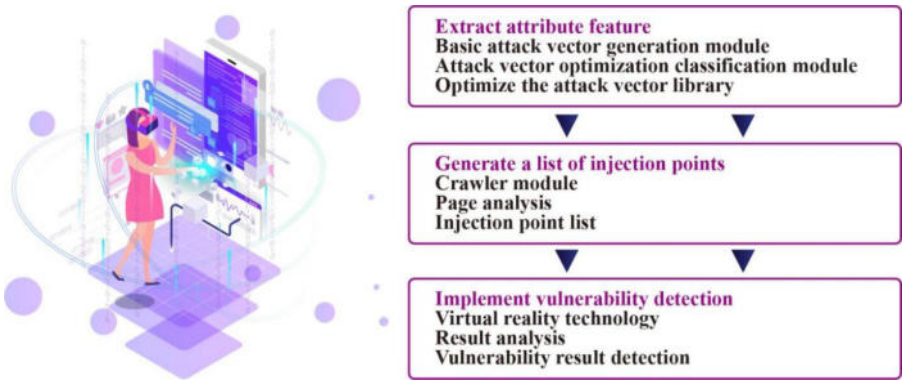


Fig. 3.1: Overall structure of the proposed method

Consequently, the scrutiny of program source code is of paramount importance in the pursuit of detecting network security vulnerabilities. In virtual reality technology, the automated identification of network security vulnerabilities hinges on using virtual reality to simulate and detect malicious attacks. This process automatically generates an attack plan based on the identified findings [17]. The schematic representation of the proposed method is illustrated in Figure 3.1.

While creating an attack graph, we emphasize achieving automatic construction. This entails crafting an attack vector repository via optimization calculations. The role of analyzing attack injection points encompasses pinpointing network injection sites, acquiring web pages through web crawlers, and subsequently subjecting the web pages to analysis via these crawler tools. The progression of attack and analysis involves emulating the attack sequence employing virtual reality technology. Subsequent scrutiny of the yielded outcomes facilitates the ultimate determination of vulnerabilities.

**3.2. Extraction of network vulnerability feature attributes.** Throughout the automated network security detection process, the network system perceives vulnerabilities within the nodes as if they were its code. Optimization calculations adjust the weight attributed to distinct vulnerability characteristics, giving rise to an attack diagram. Over time, attackers will exploit the network via diverse vulnerabilities, and the visible weight disparities between these vulnerabilities will become apparent [12]. The composition of an attack graph involves incorporating a vulnerability group technique. At a given point in time, denoted as "$t$", within a cluster featuring "$m$". vulnerabilities, it is postulated that each vulnerability houses "$k$" individual genes. The vulnerability gene studied in this paper adopts the vulnerability gene, exposure gene and repair gene, so $k$ is 3, and the location of a vulnerability can be shown as $q_i = (q_{i1}, q_{i2}, q_{i3})$, where $i = 1, 2, \cdots, m$. In the process of generating the attack map, the starting position of the vulnerability can be regarded as a point in the spatial coordinates, so the location of the vulnerability can be represented by digitizing when the coordinates are iterated. Set the movement speed of the vulnerability $v_i = (v_{i1}, v_{i2}, v_{i3}), v_{i1}, v_{i2}, v_{i3}$ are the movement speed of the vulnerability in the three coordinate axes, and its attribute can be expressed as:

$$V_{i+1} = \omega \vartheta_{ir} + \varepsilon_1 (z_{ir} - p_{ir}) + \varepsilon_2 (z_{ir} - p_{ir}) \tag{3.1}$$

In the above formula, $\varepsilon_1$ is the learning coefficient, $\varepsilon_2$ is the network learning coefficient, $\omega$ is the inertia coefficient, $\vartheta_{ir}$ is the initial speed of the vulnerability, $z_{ir}$ is the optimal location of the vulnerability, and $p_{ir}$ is the probability of the vulnerability being applied in the attack, then $p_{ir}$ meets:

$$\sum_{i=1}^{m} p_{ir} = 1, \quad 0 < p_{ir} < 1 \tag{3.2}$$

When $\omega$ is between 0 and 1, vulnerabilities in motion can converge, so when the value of $\omega$ is determined, the number of malicious attacks can be predicted. When the vulnerability in the network is exposed to a large

Table 3.1: Analysis of characteristics and attributes of network security vulnerabilities

| S.No. | Vulnerability characteristic attribute | Size | Number of source code files | Total lines of source code | Version number |
|---|---|---|---|---|---|
| 1 | Blender | 144.6 MB | 4126 | 1628412 | 2.78c |
| 2 | Clang | 124.6 MB | 2258 | 1324761 | 3.9.1 |
| 3 | Crystal Space | 489.9 MB | 4421 | 944126 | 2.0 |
| 4 | Firefox | 1.65 GB | 21274 | 7321054 | 50.0.1 |
| 5 | MPlayer | 116.3 MB | 3215 | 4521585 | 5.2.2 |
| 6 | Mysql | 482 MB | 5569 | 5445102 | 10.3.6 |
| 7 | PHP | 149.3 MB | 1932 | 4568941 | Va3.8 |
| 8 | Nebula Device 2 | 86.6 MB | 1106 | 2384161 | 4.1.3 |
| 9 | OpenSceneGraph | 166.4 MB | 2601 | 1158523 | 4.1.0 |

extent, the degree of repair will also increase, and the vulnerability will be difficult to be applied. Therefore, the three attributes of vulnerability will increase with time, so $p_{ir}$ will decrease, and the extraction of vulnerability feature attributes is completed.

**3.3. Generation of injection point list.** Upon utilizing the attributes extracted to represent vulnerability features, the subsequent step involves scrutinizing injection points, culminating in creating a roster of injection points. This undertaking necessitates the support of a web crawler [4, 9]. The web crawler essentially functions as a program that autonomously gathers these vulnerability feature attributes. Through analysis of these attributes, it determines the sequencing within the URL queue for capture while also retaining the pertinent attributes.

The web crawler can proficiently retrieve pages from the target system within vulnerability detection. This process involves identifying forms, input fields, and other elements and subsequently dissecting their properties. Based on this attribute information, vulnerability detection programs can identify and append reasonable vulnerability junctures to the injection point list. The collated attributes are categorized into nine groups, and the configuration for vulnerability analysis is outlined in Table 3.1.

Many redundant segments within the URL queue tend to emerge during the crawling process, impacting the crawling pace. In response, we have opted for implementing the BloomFilter algorithm to deduce, identify, adjust, and store vulnerabilities before inclusion in the URL queue procured through crawling. Once this determination is reached, the queues not belonging to the collection are incorporated, thus achieving the desired collection. An "ID" value of "Y" signifies the presence of this queue within the collection. Subsequently, the queue is introduced into the "urlset" to finalize the deduplication process, culminating in compiling the injection point list, outlined in Table 3.2.

The system employs the forms retrieved from the attack injection point for vulnerability analysis and documentation, generating a list of attribute injection points. Subsequently, these analyses and records facilitate an extensive exploration of vulnerabilities, culminating in the automated selection of the most suitable detection approach [15].

**3.4. Implementation of vulnerability detection under virtual reality technology.** In virtual attack testing, guided by the information encapsulated in the generated injection point list, virtual reality technology is harnessed to emulate attack behaviours and engage with the server. In the context of employing virtual reality technology to detect network security vulnerabilities, it proves beneficial for reproducing the network environment. This allows security experts to immerse themselves in the network environment, facilitating a firsthand comprehension of the location, nature, and repercussions of security vulnerabilities.

For instance, within a virtual reality setting, security experts possess the capacity to enact attack scenarios, assess the network environment's resilience against these attacks, and uncover and rectify latent vulnerabilities.

Table 3.2: List of attribute injection points

| URL Queue Name | Source code file name | Starting line number of the duplicate part of the queue | Relative error in duplication (attack input) | Maximum observed relative error (corresponding input) |
|---|---|---|---|---|
| Blender | Key.c | 668 | 9.12E-12 (2.06E12) | 6.12E-11 5.66E14) |
| OpenScene | Shande out set.c | 251 | 6.61E-11 (6.32E02) | 3.54E-61 (1.42E03) |
| Device 2 | Sc allouy.c | 2456 | 6.51E-10 (1.11E12) | 5.61E-03(1.55E06) |
| Blender | key.c | 214 | 5.66E-09 (6.30E11) | 7.31E-08 (1.15E10) |
| MPlayer | note_draw.c | 1032 | 7.54E-12 (1.25E04) | 8.36E-06 (6.58E03) |
| Crystal | audio_concica.c | 825 | 5.32E-11 (8.36E06) | 9.54E-13 (9.32E10) |
| Blender | key.c | 10 | 4.33E-12 (9.34E08) | 7.96E-01(6.47E04) |
| Clang | pixman_gration.c | 251 | 6.48E-04 (6.97E11) | 8.45E-11 (6.32E06) |
| Firefox | opus_Active.c | 483 | 1.61E-09 (8.99E13) | 9.44E-02 (7.35E11) |
| OpenScene | cairo_out.c | 125 | 5.64E-11 (4.66E13) | 6.21E-07 (9.32E12) |
| Firefox | sd_root.c | 2156 | 8.34E-14 (6.32E05) | 3.14E-02 (6.05E11) |
| Graph | cairo_strock.c | 232 | 6.24E-09 (7.98E13) | 4.20E-06 (6.21E01) |
| OpenScene | Shande out | 18 | 1.98E-01 (9.63E07) | 9.34E-05 (6.14E05) |
| PHP | Shortest_path_suppression.c | 231 | 9.61E-81 (5.68E09) | 9.01E-08 (6.77E12) |

Throughout this process, due to the attacker's virtual identity, the attribute of network security vulnerabilities can be construed as the fundamental genetic element for constructing a virtual attack. The established virtual reality environment and virtual attacks are tools for extracting information embedded within network security vulnerabilities. Subsequently, the security vulnerability classification is executed by categorizing vulnerability, exposure, and repair genes in alignment with the classification of attack information [3]. This delineates the functional framework within the realm of vulnerability detection for virtual attacks.

In the detection process, after orchestrating artificial virtual vulnerability attacks through gene selection and the cross-assembly of virtual attacks, the results of these attacks are inputted into the calculation of a fitness function. This yields vulnerability information and other pertinent data, ultimately detecting network security vulnerabilities. The research on automatic detection methodologies for network security vulnerabilities under the purview of virtual reality technology has been concluded.

**4. Results and Discussion.** A simulation experiment using network security vulnerabilities is constructed within the virtual reality technology to validate the credibility of the automated detection approach presented in this paper. This analysis aims to compare the detection outcomes with those yielded by two conventional vulnerability methodologies, thus substantiating the merits of the proposed approach. The illustrative network structure for the experimentation is depicted in Figure 4.1. The figure illustrates the presence of five nodes: A, B, C, D, and E. Node A represents the external network intrusion host. Within the firewall, the four nodes have distinct roles: furnishing network services, managing database services, safeguarding and overseeing servers, and preserving critical files. It's worth noting that intranet external network access and intranet-to-intranet communication do not necessitate firewalls. Nevertheless, when an extranet node seeks access to an intranet node, passage through the firewall is mandatory, granting exclusively network server access [18, 11].

It is important to acknowledge that specific actual vulnerabilities within the intranet can potentially target nodes within the intranet. The vulnerability configurations for each intranet node are listed in Table 4.1.

Upon establishing the experimental environment, three distinct methods are employed for the detection: the conventional modelling detection method, the numerical detection method, and the network security vulnerability automatic detection method developed within the framework of virtual reality technology as outlined in

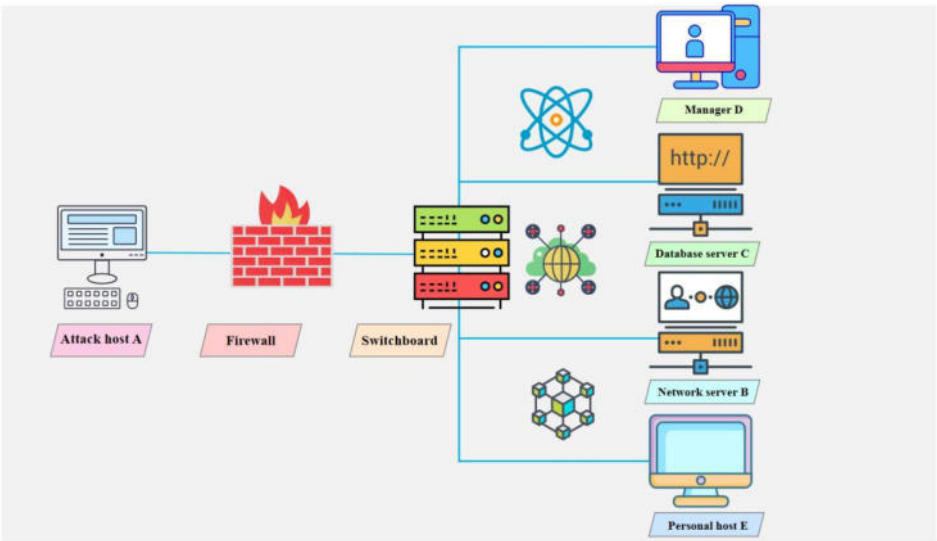


Fig. 4.1: Construction of network topology

Table 4.1: Intranet node vulnerability settings

| S.No. | Node | Vulnerability number | Server or software | Privilege escalation | Ease of use (%) |
|---|---|---|---|---|---|
| 1 | Network server B | Mc10-062 | www | U-R | 70 |
| 2 | Network server B | Mc11-002 | www | U-A | 100 |
| 3 | Database server C | Mc11-012 | Oracle | O-A | 90 |
| 4 | Database server C | Mc10-065 | Oracle | O-U | 85 |
| 5 | Manager D | Mc10-045 | Windows | O-A | 100 |
| 6 | Personal host E | Mc10-006 | Office | O-A | 70 |

this paper. The Figure 4.2 illustrates a structured listing of vulnerabilities linked to diverse nodes, servers, and software components within the network environment. Each entry is distinguished by a unique vulnerability number, denoting its specific characteristics. The corresponding server or software connected to each vulnerability is specified, alongside an assessment of the level of privilege escalation that the vulnerability permits.

Moreover, the exploitability of each vulnerability is represented as a percentage, shedding light on the relative complexity of exploiting each instance. The vulnerabilities span across a range of nodes, encompassing network servers, database servers, manager nodes, and personal hosts. The figure furnishes an in-depth understanding of the vulnerabilities infusing the network environment, explaining their relevant attributes and potential implications for security.

**4.1. Detection of interactive frequency.** To advance the detection process, a key measure is to elevate the frequency of network interactions. To enhance the precision of vulnerability detection outcomes, 70 detection samples are used to monitor the variations in interaction frequency among the three methods throughout the detection procedure. The time consumption for each method is illustrated in Figure 4.3.

The depicted figure highlights a discernible trend: as the quantity of test samples grows, the method introduced in this paper consistently exhibits an escalation in response requirements. During the initial 20-second detection window, the modeling detection method gains a notable edge, owing to its coverage of a portion

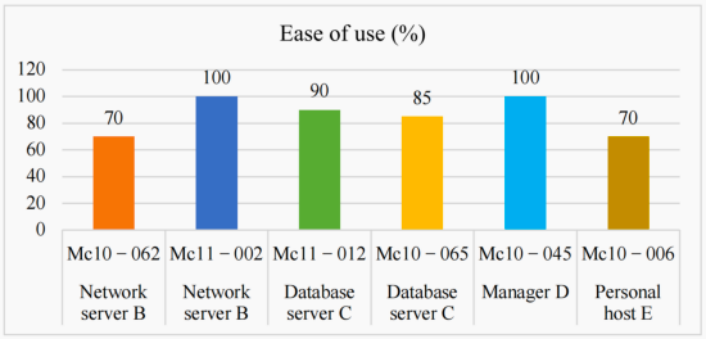


Fig. 4.2: Overview of vulnerabilities in the network environment

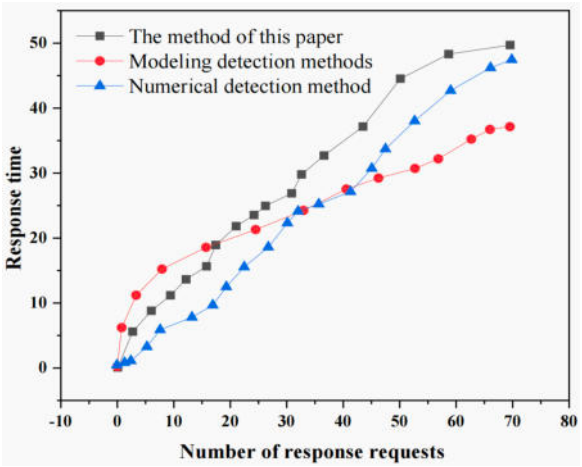


Fig. 4.3: Comparison of interaction of response time

of the sample data. As the testing duration reaches the 20-second mark, compared to the conventional two methods, the method outlined in this paper surpasses them regarding response requirements within the same time span. This interaction frequency results in a more incredible count of detected vulnerabilities, showcasing distinct advantages [10, 2].

**4.2. Detection of vulnerabilities.** As established in the preceding article, the simulation network vulnerability detection experiment is conducted for the four designated target nodes - B, C, D, and E. During this experiment, the time delay for vulnerability detection is logged across the three methods and subsequent detection outcomes are compared. This information is outlined in Table 4.2.

The data presented in the table above clearly indicates that the detection delay associated with the method detailed in this paper stands at 75.33 milliseconds. This represents a significant reduction of 290.11 milliseconds and 337.30 milliseconds compared to the respective delay times of the two traditional methods. This achievement not only translates to minimized detection delays but also substantiates the efficacy of the designed automated detection approach for network security vulnerabilities within the context of virtual reality technology [16, 8]. Virtual reality technology systems are commonly employed in addressing network security challenges. Constructed upon virtual reality technology, a novel scheme for automated network security vulnerability detection emerges, effectively improving historical issues of continued delays and detection efficiency in vulnerability assessment. Furthermore, this scheme strengthens the network security vulnerability detection capacity.

Table 4.2: Comparison of test results of three methods

| Method | Target node | Number of detected vulnerabilities | Detection delay (ms) |
|---|---|---|---|
| The method of this paper | B | Mc10-062<br>Mc11-002 | 75.33 |
| | C | Mc11-012<br>Mc10-065 | |
| | D | Mc10-045 | |
| | E | Mc10-006 | |
| Modeling detection method | B | Mc10-062 | 365.44 |
| | C | Mc10-065 | |
| | D | Mc10-045 | |
| | E | - | |
| Numerical detection method | B | Mc10-062<br>Mc11-002 | 412.63 |
| | C | - | |
| | D | Mc10-045 | |
| | E | Mc10-006 | |

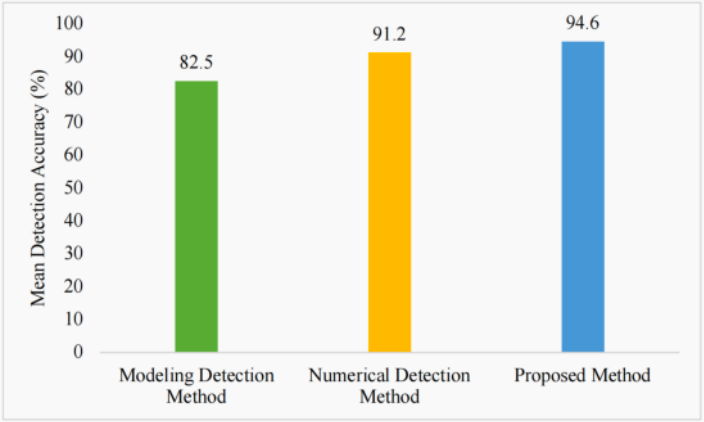


Fig. 4.4: Comparison of mean accuracy detection

A comparison of the three detection methods varying mean detection accuracy in vulnerability identification is shown in Figure 4.4. The modeling detection method achieved 82.5% accuracy, displaying proficiency in recognizing pattern-based vulnerabilities, yet showing potential limitations against anomalies and zero-day vulnerabilities. Outperforming this, the numerical detection method reached 91.2%, employing advanced algorithms for heightened precision in detecting known vulnerabilities and refined deviations. However, the proposed method excelled significantly with 94.6% accuracy, capitalizing on virtual reality and optimization to demonstrate exceptional adaptability, versatility, and effectiveness in identifying vulnerabilities, particularly within complex scenarios and emerging threats.

Virtual reality technology plays a dual role in this context. Firstly, it replicates authentic network environments encompassing network topology, servers, and network devices. This gives security experts deeper insights into network operations and potential vulnerability locations. Through the lens of virtual reality technology, security experts can more intuitively identify network vulnerabilities, thereby enhancing the evaluation of risks

and threats in the network. Secondly, virtual reality technology enables the simulation of attack scenarios and vulnerability remediation processes [14, 20]. This enhances the skillsets and competencies of security experts. As we progress, we must intensify research into the fusion of virtual reality technology and network security management. This will further enhance virtual reality technology's advantages and guarantee the network environment's stable and secure operation.

**5. Conclusion.** This research introduces the integration of virtual reality technology for computer network security vulnerability detection, addressing issues related to extended detection delays in traditional systems. Through optimization calculations, distinct vulnerability attributes are extracted and analyzed using web crawlers to compile an injection point list. Virtual reality technology simulates attacks, capturing genetic attributes through virtual vulnerabilities. Experimental results exhibit significant reductions in detection time delays, affirming the efficiency of the proposed automatic detection method compared to traditional approaches. The findings reveal a substantial reduction in detection time delays, with respective reductions of 290.11 milliseconds and 337.30 milliseconds compared to the two traditional methods. Virtual reality technology offers substantial application potential in computer network security vulnerability detection. Its ability to simulate real network environments empowers experts to identify and remediate vulnerabilities virtually, enhancing detection precision and efficiency while minimizing risks inherent in real-world assessments. This approach can shape the future of vulnerability detection, accompanying in a new era of effective and efficient security measures within the evolving landscape of network technology.

REFERENCES


[1] A. S. Adegoke, T. T. Oladokun, T. O. Ayodele, S. E. Agbato, A. D. Jinadu, and S. O. Olaleye, *Analysing the criteria for measuring the determinants of virtual reality technology adoption in real estate agency practice in lagos: a dematel method*, Property Management, 40 (2022), pp. 285–301.

[2] F. Bellalouna, *Digitization of industrial engineering processes using the augmented reality technology: Industrial case studies*, Procedia CIRP, 100 (2021), pp. 554–559.

[3] M. Brugger, *Limitations of the hyperplane separation technique for bounding the extension complexity of polytopes*, Operations Research Letters, 49 (2021), pp. 896–901.

[4] Y. Du, C. Duan, and T. Wu, *Lubricating oil deterioration modeling and remaining useful life prediction based on hidden semi-markov modeling*, Journal of Engineering Tribology, 236 (2022), pp. 916–923.

[5] E. Ferretti, J. R. Schoenherr, A. Mattiola, and T. Daboval, *Vulnerabilities in clinician–parent exchanges and the cascade of communication traps: a review*, Archives of Disease in Childhood, 108 (2023), pp. 86–90.

[6] K. P. Gupta and P. Bhaskar, *Teachers' intention to adopt virtual reality technology in management education*, International Journal of Learning and Change, 15 (2023), pp. 28–50.

[7] Y. Li and X. Li, *Research on multi-target network security assessment with attack graph expert system model*, Scientific Programming, 2021 (2021), pp. 1–11.

[8] Z. Lin, Y. Lai, T. Pan, W. Zhang, J. Zheng, X. Ge, and Y. Liu, *A new method for automatic detection of defects in selective laser melting based on machine vision*, Materials, 14 (2021), p. 4175.

[9] X. Liu, K. Shi, Z. Wang, and J. Chen, *Exploit camera raw data for video super-resolution via hidden markov model inference*, IEEE Transactions on Image Processing, 30 (2021), pp. 2127–2140.

[10] T.-W. Lui and L. Goel, *Learning effectiveness of 3d virtual reality in hospitality training: a situated cognitive perspective*, Journal of Hospitality and Tourism Technology, 13 (2022), pp. 441–460.

[11] J. McFadden, K. Jung, B. Robinson, and T. R. Tretter, *Teacher-developed multi-dimensional science assessments supporting elementary teacher learning about the next generation science standards*, Journal of Science Teacher Education, 33 (2022), pp. 55–82.

[12] X. Meng, S. Kundu, A. K. Kanuparthi, and K. Basu, *Rtl-contest: Concolic testing on rtl for detecting security vulnerabilities*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41 (2021), pp. 466–477.

[13] A. O'Connor, A. Tai, Z. Kopsaftis, and K. Carson-Chahhoud, *Qualitative study protocol: Augmented reality technology to deliver asthma inhaler technique training for children and adolescents with asthma*, International Journal of Qualitative Methods, 20 (2021), pp. 1–11.

[14] G. Tang, L. Yang, S. Ren, L. Meng, F. Yang, and H. Wang, *An automatic source code vulnerability detection approach based on kelm*, Security and Communication Networks, 2021 (2021), pp. 1–12.

[15] Z. WU, C. JIAO, and L. CHEN, *Tire defect detection method based on improved faster r-cnn*, Journal of Computer Applications, 41 (2021), pp. 1939–1946.

[16] ———, *Tire defect detection method based on improved faster r-cnn*, Journal of Computer Applications, 41 (2021), pp. 1939–1946.

[17] X. Yang, L. Shu, J. Chen, M. A. Ferrag, J. Wu, E. Nurellari, and K. Huang, *A survey on smart agriculture: Development modes, technologies, and security and privacy challenges*, IEEE/CAA Journal of Automatica Sinica, 8 (2021), pp. 273–302.

[18] J. Zhang, Y. Jin, B. Sun, Y. Han, and Y. Hong, *Study on the improvement of the application of complete ensemble empirical mode decomposition with adaptive noise in hydrology based on rbfnn data extension technology.*, CMES-Computer Modeling in Engineering & Sciences, 126 (2021), pp. 755–770.
[19] D. Zhao, *Choice of environmental and economic path for building a supply chain financial cloud ecosystem under the background of "internet+"*, Wireless Communications and Mobile Computing, 2021 (2021), pp. 1–11.
[20] R. Zheng, H. Ma, Q. Wang, J. Fu, and Z. Jiang, *Assessing the security of campus networks: the case of seven universities*, Sensors, 21 (2021), p. 306.

*Edited by:* C. Venkatesan
*Special Issue:* Next Generation Pervasive Reconfigurable Computing for High Performance Real Time Appls
*Received:* Mar 20, 2023
*Accepted:* Aug 30, 2023