

## Implementation of Network and Server Monitoring on Mikrotik with Netwatch Via Telegram Bot

Yohanes Maestro Pasaribu<sup>1</sup>, Ferdiansyah<sup>2</sup>

<sup>1,2</sup>Informatics Engineering Study Program, Bina Darma University  
Jl. Jenderal Ahmad Yani No.3, 9/10 Ulu, District Seberang Ulu I, Palembang  
Email: yohanespsrb@gmail.com, Ferdi@binadarma.ac.id

### ABSTRACT

*Computer network is a communication between two or more computers that can exchange information data. The development of a network is supported by adequate devices in that place. When the network has developed, it must be able to periodically monitor all connected network devices. Methodist School 2 conducts online platform-based learning, namely JIA (Jelajah Ilmu Acer) for the junior high, high school and vocational levels. Methodist School 2 Palembang uses Mikrotik as its network router, a problem was found that it could not control the network at Methodist School 2 Palembang regularly. This will make it difficult for a network administrator to supervise because they have to control and analyze the performance of their devices and network communications manually. With this problem, the author made an Implementation of Network and Server Monitoring on Mikrotik with Netwatch via Telegram Bot, the author configured several logs that will be integrated into the telegram bot in the Mikrotik tool, Netwatch. The telegram bot will send message notifications about the status of connected devices in the proxy, with this project it is hoped that the author can help network administrators to be able to help control and analyze the network at Methodist School 2 Palembang.*

**Keywords:** *Network Monitoring, Mikrotik, Telegram Bot Integration, Netwatch*

### Introduction

Computer network According to Ahmodul Hadi, "It is a communication between two or more computers that can exchange information data". The development of a network is supported by adequate devices in that place [1]. With the development of the network, regular supervision is needed on all connected network devices. This will make it difficult for a network administrator to supervise because they have to control and analyze the performance of their devices and network communications manually [2]–[4].

Sekolah Methodist 2 Palembang is a Christian college school in Palembang that has implemented digitalization transformation in its learning process. With learning, the JIA (Jelajah Ilmu Acer) Platform supports all student learning, especially at the junior high, high school and vocational levels. With this way of learning from JIA, methodist school 2 supports paperless efforts. This is related to providing books digitally. Each child will be provided with a digital book that can be accessed on the JIA platform using the internet network. Therefore, by applying this learning method, Methodist School 2 Palembang needs the internet with full access control in network monitoring. Sekolah Methodist 2 Palembang uses Mikrotik as a server router to manage and control all activities and use of the available network. But in carrying out maintenance and monitoring of existing network servers, network administrators at Methodist School 2 Palembang must do it manually. The lack of mobility in maintenance makes it difficult for network administrators to detect problems that are happening because network administrators at Methodist School 2 Palembang are not always in the network control room [5]–[7].

With these conditions, researchers created a network and server monitoring system using the netwatch feature integrated into the telegram bot to send log notifications of the data needed. This monitoring system configuration is made on Mikrotik by inputting several scripts on Mikrotik and as a result the client / administrator can monitor the network and server via the Telegram bot with a request-sent system where the receiver will make requests to Mikrotik with the desired menu / option on the Telegram bot display and Mikrotik will send the desired data log to the receiver in the form of notifications via Telegram bot [8]–[10].

## Research Methods

### Research Methods

The author uses the ADDIE method for network topology development which includes Analysis, Design, Development, Implementation, Evaluation. This ADDIE model was developed by Dick and Carry in 1996 to design a learning system [11]–[13].

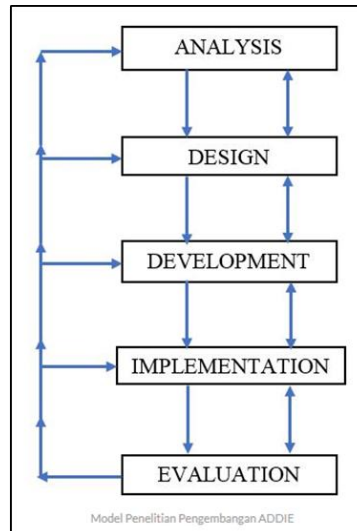


Figure 1 Research Methods

### Analysis

The first stage of the researcher first analyzes the existing network conditions in Methodist School 2 what is needed and shortcomings in Methodist School 2, then it will be developed and improved through this research [14]–[16].

The needs of tools and materials needed in this study:

Table 1 Tools and Materials Needed in Research

Tools and Materials	Specifications	Sum	Information
Computer (Administrator)	AMD Ryzen 5 3500 NVIDIA GeForce GTX 1650 12 GB DDR4 SSD 256 GB Windows 11 Pro	1 Unit	Computers play an important role in this research in the form of configuration, checking network connections, testing configuration results.
Mikrotik RB3011UiAS-RM	QCA8337 1 GB 10-30 V	2 Units (Bridge)	Mikrotik that will manage the network that will be configured by <i>Netwatch</i> to the Telegram Bot
Smartphone (Administrator)	Mediatek Helio G90T 6 GB 2.4 Ghz 300 Mbps	1 Unit	Monitoring of implementation results
Access Point Unifi UAP AC Lite	5 GHz 867 Mbps Support 802.11 a/b/g/n/r/k/v/ac DC 24 V 0.5 A	6 Units	Access Point that will spread <i>Wi-Fi</i> signals to laptop or smartphone devices
ISP 1	150Mbps	1 Unit	ISP 1 acts as the Main ISP
ISP 2	300Mbps	1 Unit	ISP 2 acts as a Backup ISP
Winbox	Version 6.44.5	-	Winbox is used to configure the mikrotik router with the <i>GUI method</i>

### Design

At this design stage, researchers will develop existing networks in Methodist School 2 by applying some additions related to network and server monitoring by adding several scripts to Netwatch in Mikrotik. With the topology design, it can facilitate implementation in the field [17]–[19].

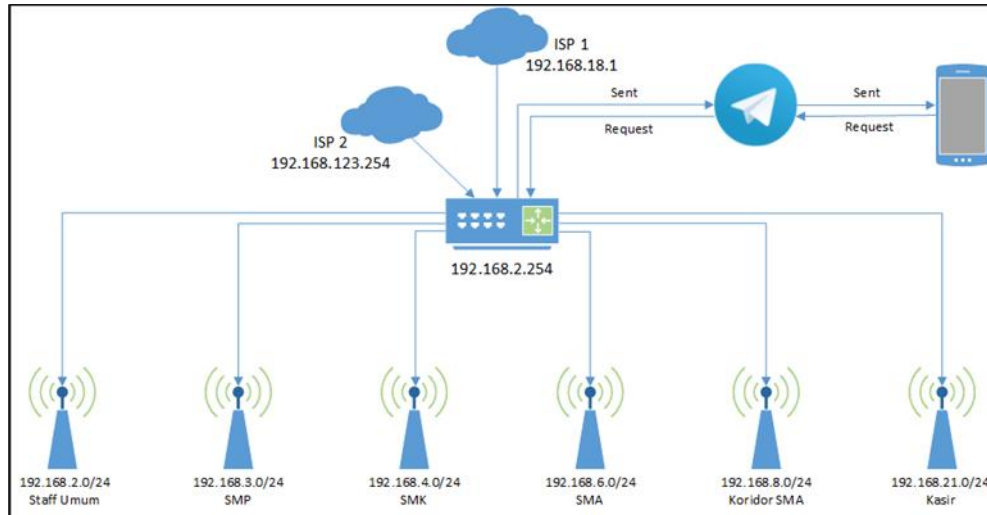


Figure 2 Network Topology Design

In the design above, the client / administrator will make a request according to the menu on the telegram bot and will be sent to the proxy, after making the request as desired the proxy will send the notification to the telegram which will be displayed on the client / administrator device [20]–[24].

### Development

At this stage the researcher will explain the performance of the network and some of the benefits of developing this research from the previous one.

With the implementation of this research, it is hoped that the network and servers at Methodist School 2 can be easier and more intense to monitor and can quickly handle problems sent notifications via telegram bots.

Researchers use tools on Mikrotik, namely Netwatch which will send log signals that have been configured in the script and integrated into Netwatch to be able to send notifications and requests to clients / administrators.

In addition to the request-sent method, the researcher created a configuration of up and down interface connection status notifications that were sent to the telegram bot. The information submitted is in the form of ether and IP how much is down or up again with a description of the date and time of the incident. Researchers make this to be able to find out in real time the occurrence of problems in terms of down and up again on the proxy router interface.

## Results and Discussion

### Implementation

Starting with creating a telegram bot account, creating this telegram bot account as a place / platform to send log data that will be configured on the proxy to the client / administrator. Search in the telegram search bar "BotFather", select the command "/newbot", enter a name for the telegram bot, and create a username for the telegram bot using the word "Bot" at the end of the username. When finished, researchers will get a reply message from BotFather sending the telegram bot link and HTTP API access token from the telegram bot that has been created.

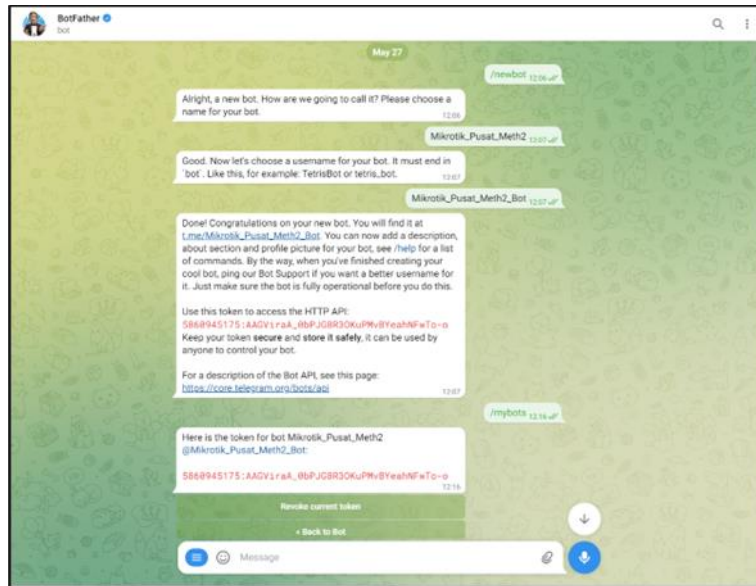


Figure 3 Telegram Bot Creation

Configure the IP Address on each ethernet port according to the topology that has been designed.

Table 2. IP Address on Each Ethernet Port

IP Allocation	IP Address	Interface
ISP 1	192.168.18.6/24	Ether2
ISP 2	192.168.123.235/24	Ether3
General Staff	192.168.2.254/24	Vlan Biz High School (eth 2)
JUNIOR	192.168.3.254/24	Vlan Biz High School (eth 3)
SMK	192.168.4.254/24	Vlan Biz High School (eth 4)
SMA	192.168.6.254/24	Vlan Biz High School (eth 6)
High School Corridor	192.168.8.254/24	Vlan Biz High School (eth 8)
Cashier	192.168.21.254/24	Ether9

Configure DNS so that clients can access domains on the internet, here researchers use ISP1 DNS 192.168.18.1, ISP2 192.168.123.254, and Cloudflare DNS 1.1.1.1

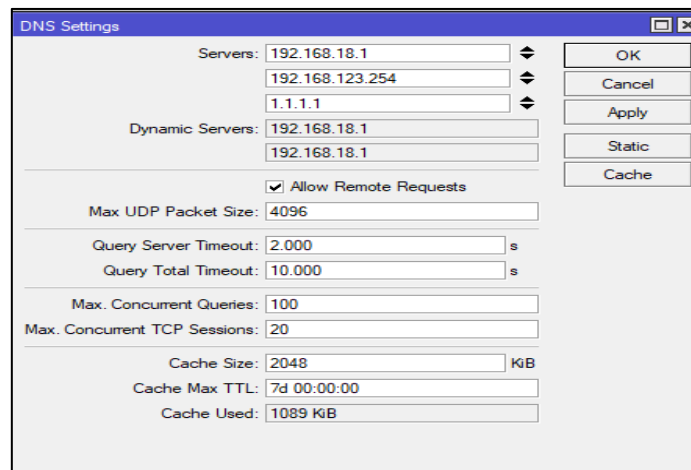


Figure 4 DNS

Next, configure the NAT firewall which is useful for making local IP access can connect to the internet by providing masquerade action on ISP 1 and ISP 2 on the NAT Firewall menu to the source interface connected to the internet, for ISP 1 add chain=srnat out interface = ether2 action=masquerade, for ISP 2 add chain=srnat out interface = ether3.

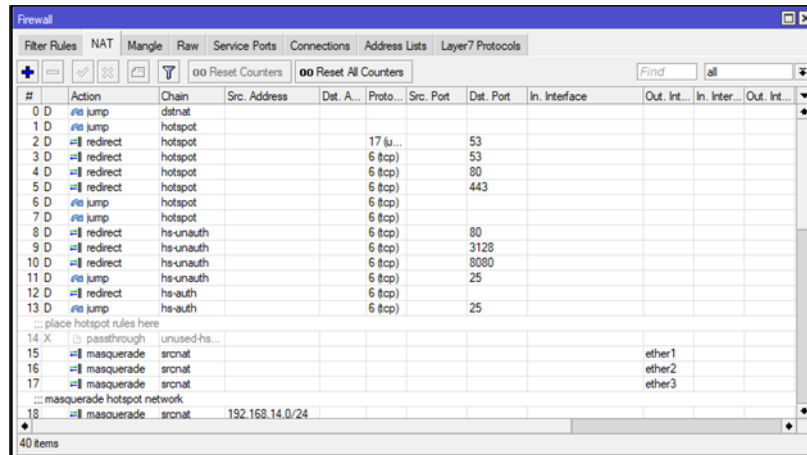


Figure 5 NAT firewall

The next configuration in the Scripts menu, in this menu will be entered the configuration script that will be used or synchronized to the telegram bot. This configuration works so that later the client / administrator can make requests for what menu you want to display or send by the proxy to the telegram bot.

1. func\_fetch configuration serves to send and retrieve data via telegram bot url.
2. func\_lowercase configuration works for automatic upper and lower case matching when entering commands in telegram bot.
3. The reboot-report configuration serves as a report when the proxy router is rebooted will display the time, date and time of occurrence which will be sent from the proxy to the telegram bot.
4. The tg\_cmd\_batal configuration functions as a command in the telegram bot to cancel the command request in the telegram bot menu.
5. The tg\_cmd\_cpu configuration serves as a command to see the updated conditions of the use of the Mikrotik router CPU.
6. Configuration tg\_cmd\_help works when the client / administrator wants to know what commands are on the telegram bot.
7. tg\_cmd\_hotspot configuration serves to check hotspots that have been created in the proxy in the form of user hotspots, disconnect all hotspots, create and delete users.
8. The tg\_cmd\_interface configuration serves to check the bandwidth of the connected ethernet, the available ethernet port.
9. The tg\_cmd\_ping configuration serves to perform ping tests to the internet or other websites using IP addresses.
10. The tg\_cmd\_ppp configuration serves to control the device of someone connected to the network through a telegram bot.
11. The tg\_cmd\_reboot configuration serves to reboot the Mikrotik router via telegram bot.
12. tg\_cmd\_stop configuration serves to stop running commands / sent by Mikrotik to the telegram bot.
13. Configure tg\_config place to configure the bot by connecting or connecting the proxy to the telegram bot by entering the Chat Bot ID and Telegram API Bot client / administrator.
14. Configure tg\_getUpdates to update bot incoming messages.
15. Configure tg\_getkey to retrieve the registered bot key.
16. Configuration tg\_sendMessage script to be able to send messages from mikrotik to telegram bots and vice versa.

After configuring in Scripts, then configure the Netwatch by adding a list of IP addresses that have been registered previously with one of the IP examples to be configured on Netwatch "192.168.2.254".

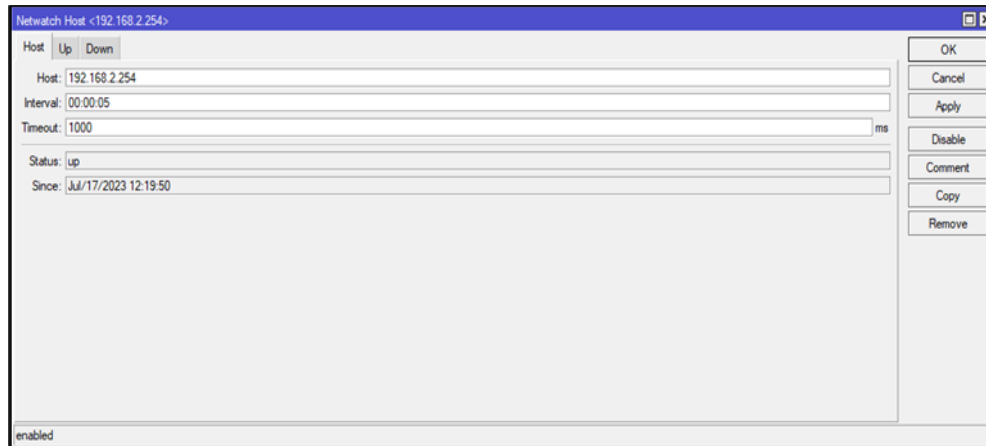


Figure 6 Netwatch IP Router Configuration

After configuring and registering several IP addresses for monitoring, then configure the Schedule to be able to send or update log data in Scripts using time intervals.

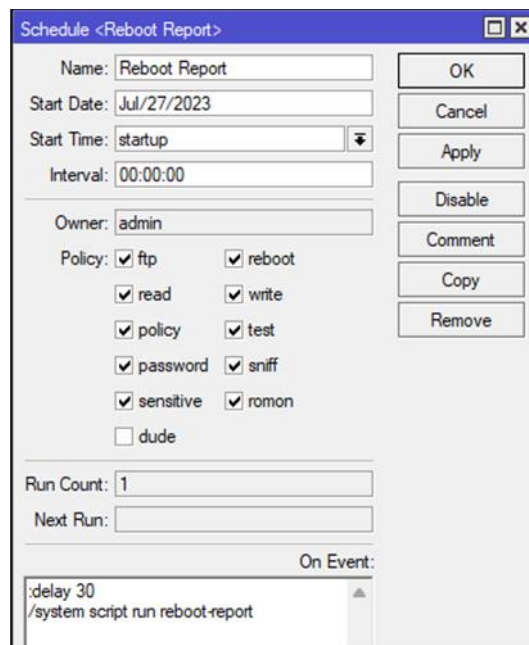


Figure 7 Configuring the Schedule reboot-report

### Evaluation

After implementing the topology that has been designed, there are several things that need to be done to try the performance of monitoring via this telegram bot.

Menu views available to client/administrator are created on tg\_config scripts.

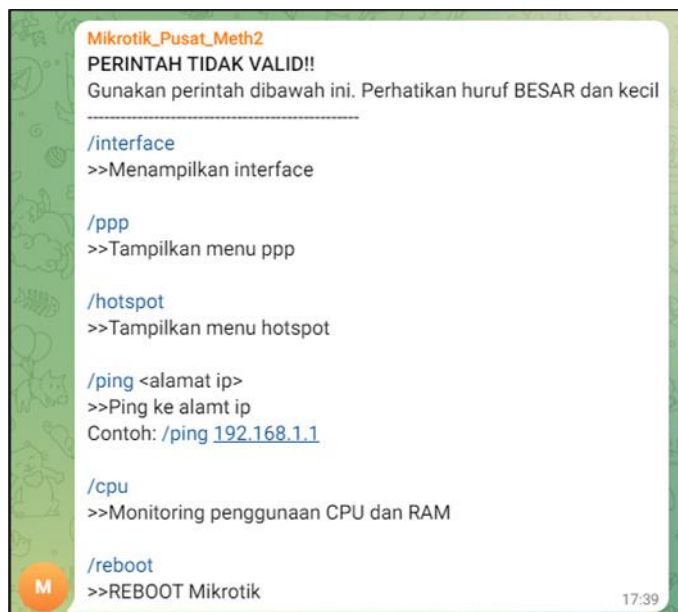


Figure 8 Basic Menu Display

Menu display / interface to be able to display what commands can be used. Figure 9 "/ether1 traffic interface" to be able to display traffic on ether1.

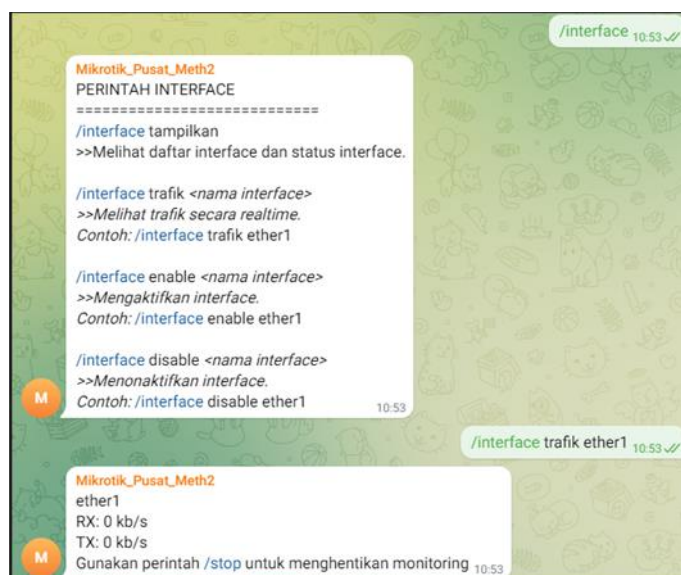


Figure 9 Command /interface display

### Recommendations

As for what was obtained from the research above, one thing that can be a recommendation for researchers is to be able to add a monitoring list to network security, one example is detecting user logins on Mikrotik via Winbox, where the access port is 8192. With this recommendation, you can quickly find out when someone logs in via Winbox and which IPs are detected logging in via Winbox.

### Conclusion

Can find out in detail from all command choices made and can be directly executed through telegram bots for choices such as creating, deleting or editing configurations on the Mikrotik router.

In this study, clients / administrators can easily monitor and control the entire network in Methodist School 2 through telegrams that have been connected to the API bot on Netwatch. In this study, researchers

created a group connected to Mikrotik using the Chat\_id in the telegram group, so that they could invite several IT staff from various levels who needed monitoring.

### References

- [1] A. Ferrando, "Combine Model Checking and Runtime Verification in Multi-Agent Systems (short paper)?," *CEUR Workshop Proceedings*, vol. 3072. pp. 302–310, 2021. [Online]. Available: [https://api.elsevier.com/content/abstract/scopus\\_id/85123275549](https://api.elsevier.com/content/abstract/scopus_id/85123275549)
- [2] X. Xie, "Automated Rule Checking for MEP Systems Based on BIM and KBMS," *Buildings*, vol. 12, no. 7, 2022, doi: 10.3390/buildings12070934.
- [3] S. Lazreg, "Verification of Variability-Intensive Stochastic Systems with Statistical Model Checking," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13703. pp. 448–471, 2022. doi: 10.1007/978-3-031-19759-8\_27.
- [4] P. Ovsianikova, "Towards user-friendly model checking of IEC 61499 systems with counterexample explanation," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2021. 2021. doi: 10.1109/ETFA45728.2021.9613491.
- [5] V. V Shkaruplyo, "On Applicability of Model Checking Technique in Power Systems and Electric Power Industry," *Studies in Systems, Decision and Control*, vol. 399. pp. 3–21, 2022. doi: 10.1007/978-3-030-87675-3\_1.
- [6] D. V Efanov, "The Weight-Based Sum Codes in the Residue Ring by Arbitrary Modulus for Synthesis of Self-Checking Digital Computing Systems," *2021 IEEE East-West Design and Test Symposium, EWDTS 2021 - Proceedings*. 2021. doi: 10.1109/EWDTS52692.2021.9581032.
- [7] P. Nov, "Developing an automated system for checking the strut arrangement in deep excavation," *Eng. J.*, vol. 25, no. 1, pp. 99–124, 2021, doi: 10.4186/ej.2021.25.1.99.
- [8] M. M. Hossain, "Developing an automated safety checking system using BIM: a case study in the Bangladeshi construction industry," *Int. J. Constr. Manag.*, vol. 22, no. 7, pp. 1206–1224, 2022, doi: 10.1080/15623599.2019.1686833.
- [9] Q. Chen, "Compliance Checking Based Detection of Insider Threat in Industrial Control System of Power Utilities," *Proceedings - 2022 7th Asia Conference on Power and Electrical Engineering, ACPEE 2022*. pp. 1142–1147, 2022. doi: 10.1109/ACPEE53904.2022.9784085.
- [10] N. Jnoub, "Fact-checking reasoning system for fake review detection using answer set programming," *Algorithms*, vol. 14, no. 7, 2021, doi: 10.3390/a14070190.
- [11] A. Valmari, "Automated Checking of Flexible Mathematical Reasoning in the Case of Systems of (In)Equations and the Absolute Value Operator," *International Conference on Computer Supported Education, CSEDU - Proceedings*, vol. 2. pp. 324–331, 2021. [Online]. Available: [https://api.elsevier.com/content/abstract/scopus\\_id/85121660387](https://api.elsevier.com/content/abstract/scopus_id/85121660387)
- [12] P. Burggräf, "Creation of an expert system for design validation in BIM-based factory design through automatic checking of semantic information," *Procedia CIRP*, vol. 99. pp. 3–8, 2021. doi: 10.1016/j.procir.2021.03.012.
- [13] M. M. de Lucena, "An analysis of the Gateway Integrity Checking Protocol from the perspective of Intrusion Detection Systems," *Des. Autom. Embed. Syst.*, vol. 25, no. 2, pp. 89–111, 2021, doi: 10.1007/s10617-020-09240-8.
- [14] J. Lee, "Unlocking Conspiracy Belief Systems: How Fact-Checking Label on Twitter Counters Conspiratorial MMR Vaccine Misinformation," *Health Commun.*, 2022, doi: 10.1080/10410236.2022.2031452.
- [15] L. He, "Petri-Net-Based Model Checking for Privacy-Critical Multiagent Systems," *IEEE Trans. Comput. Soc. Syst.*, 2022, doi: 10.1109/TCSS.2022.3164052.
- [16] L. Cheng, "Checking is Believing: Event-Aware Program Anomaly Detection in Cyber-Physical Systems," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 2, pp. 825–842, 2021, doi: 10.1109/TDSC.2019.2906161.
- [17] P. Ballarini, "Formal analysis of production line systems by probabilistic model checking tools," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2021. 2021. doi: 10.1109/ETFA45728.2021.9613494.
- [18] B. Kim, "How Robust are Fact Checking Systems on Colloquial Claims?," *NAACL-HLT 2021 - 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*. pp. 1535–1548, 2021. [Online]. Available: [https://api.elsevier.com/content/abstract/scopus\\_id/85118031376](https://api.elsevier.com/content/abstract/scopus_id/85118031376)
- [19] F. Mercaldo, "MODEL CHECKING FOR REAL-TIME ATTACK DETECTION IN WATER DISTRIBUTION SYSTEMS," *Informatics Autom.*, vol. 21, no. 2, pp. 219–242, 2022, doi:



- 10.15622/IA.21.2.1.
- [20] Y. Wang, “Exciton-plasmon interactions gated self-checking functional photoelectrochemical system for adenosine monitoring,” *Sensors Actuators, B Chem.*, vol. 336, 2021, doi: 10.1016/j.snb.2021.129746.
- [21] D. Liu, “Flexible Data Integrity Checking with Original Data Recovery in IoT-Enabled Maritime Transportation Systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2618–2629, 2023, doi: 10.1109/TITS.2021.3125070.
- [22] A. A. Hamza, “HSAS-MD Analyzer: A Hybrid Security Analysis System Using Model-Checking Technique and Deep Learning for Malware Detection in IoT Apps,” *Sensors*, vol. 22, no. 3, 2022, doi: 10.3390/s22031079.
- [23] N. Drawel, “Transformation-based model checking temporal trust in multi-agent systems,” *J. Syst. Softw.*, vol. 192, 2022, doi: 10.1016/j.jss.2022.111383.
- [24] S. Ahmed, “Fact Checking: An Automatic End to End Fact Checking System,” *Studies in Computational Intelligence*, vol. 1001. pp. 345–366, 2022. doi: 10.1007/978-3-030-90087-8\_17.