Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 21, No. 1, November 2021, pp. 43∼52
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v21i1.1078

Deteksi Serangan Denial of Service pada Internet of Things Menggunakan Finite-State Automata

Denial of Service Attack Detection on Internet of Things Using Finite-State Automata

Fery Antony[1], **Rendra Gustriansyah**[2]
Universitas Indo Global Mandiri, Indonesia

Informasi Artikel

Genesis Artikel:

Diterima, 12 Juni 2021
Direvisi, 10 Oktober 2021
Disetujui, 15 September 2021

Kata Kunci:

Deteksi
DoS
Finite-state automata
IoT
Pencegahan

ABSTRAK

Internet of things memiliki kemampuan untuk menghubungkan obyek pintar dan memungkinkan mereka untuk berinteraksi dengan lingkungan dan peralatan komputasi cerdas lainnya melalui jaringan internet. Namun belakangan ini, keamanan *jaringan internet of things* mendapat ancaman akibat serangan *cyber* yang dapat menembus perangkat *internet of things* target dengan menggunakan berbagai serangan *denial of service*. Penelitian ini bertujuan untuk mendeteksi dan mencegah serangan *denial of service* berupa *synchronize flooding* dan *ping flooding* pada jaringan *internet of things* dengan pendekatan *finite-state automata*. Hasil pengujian menunjukkan bahwa pendekatan *finite-state automata* berhasil mendeteksi serangan *synchronize flooding* dan *ping flooding* pada jaringan *internet of things*, tetapi pencegahan serangan tidak secara signifikan mengurangi penggunaan prosesor dan memori. Serangan *synchronize flooding* menyebabkan *delay* saat mengaktifkan/menonaktifkan peralatan *internet of things* sedangkan serangan *ping flooding* menyebabkan *error*. Implementasi *bash-iptables* berhasil mengurangi serangan *synchronize flooding* dengan efisiensi waktu pencegahan sebesar 55,37% dan mengurangi serangan *ping flooding* sebesar 60% tetapi dengan waktu yang tidak signifikan.

Keywords:

Detection
DoS
Finite-state automata
IoT
Prevention

ABSTRACT

The internet of things has the ability to connect smart objects and enable them to interact with other intelligent computing environments and equipment via the internet network. But lately, the internet of things network security has come under threat due to cyber-attacks that can penetrate the target the internet of things device using various denial of service attacks. This study aims to detect and prevent denial of service attacks in the form of synchronize flooding and ping flooding on the internet of things networks using the finite-state automata approach. The test results show that the finite-state automata approach successfully detects synchronize flooding and ping flooding attacks on the internet of things networks, but attack prevention does not significantly reduce processor and memory usage. The synchronize flooding attacks cause delays when turning the internet of things equipment on/off while ping flooding attacks cause errors. The implementation of bash-iptables succeeded in reducing synchronize flooding attacks with an efficiency of prevention time by 55.37% and reducing ping flooding attacks by 60% but with insignificant time.

Penulis Korespondensi:

Rendra Gustriansyah,
Program Studi Teknik Informatika,
Universitas Indo Global Mandiri,
Email: rendra@uigm.ac.id

1. PENDAHULUAN

Internet of Things (IoT) diusulkan pertama kali pada tahun 1999 oleh Kevin Ashton. IoT adalah suatu konsep dimana suatu obyek cerdas dapat mengirimkan data melalui jaringan tanpa melibatkan manusia. IoT mempunyai kemampuan yang dapat menghubungkan berbagai obyek pintar untuk berinteraksi dengan lingkungannya maupun peralatan komputasi cerdas lainnya melalui jaringan internet [1]. Saat ini sensor IoT telah disematkan pada perangkat seluler, peralatan industri, lingkungan, perangkat medis, dan lain-lain [2–4]. Implementasi IoT dalam berbagai aspek kehidupan manusia telah memberikan banyak keuntungan dan kemudahan. Peningkatan penggunaan aplikasi IoT ini memerlukan jaringan IoT yang aman untuk menangani ribuan atau jutaan sensor IoT [5].

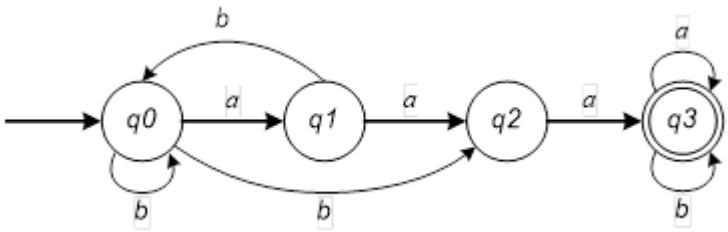
Namun, belakangan ini keamanan jaringan IoT terancam karena serangan *cyber* yang dapat menembus perangkat IoT target (pengguna) dengan menggunakan berbagai serangan *Denial of Service* (DoS) [6]. Ancaman ini terjadi karena perangkat IoT pengguna lebih mudah diakses karena memiliki tingkat keamanan yang lebih rendah dibandingkan dengan keamanan *server*. Bahkan, menurut Kaspersky Lab dan B2B International bahwa lebih dari 40% bisnis di dunia telah menjadi korban dari serangan DoS [7].

DoS merupakan serangan yang bertujuan untuk mempengaruhi trafik jaringan sehingga jaringan tersebut tidak dapat digunakan oleh pengguna yang berhak/sah [8]. Serangan DoS dilakukan dengan cara membanjiri *ip address* jaringan target dengan *request* sehingga sistem menjadi *crash* atau hang atau turun kinerjanya karena beban CPU tinggi [9]. Ini adalah salah satu metode serangan *cyber* paling populer dalam keamanan jaringan [10]. Selain itu, tipe serangan DoS yang lain adalah *Ping of Death*, *SYN Attack*, *Land Attack*, *UDP Flood*, dan *Smurf Attack* [11].

Adapun metode yang sering digunakan untuk mendeteksi serangan *Denial of Service* (DoS) dalam beberapa tahun terakhir ini adalah metode *Machine Learning/Deep Learning* [12–17] dan *Artificial Intelligence* [18–22]. Sementara, penelitian ini akan menggunakan model *Finite-State* Automata (FSA) untuk mendeteksi dan mencegah serangan DoS berupa SYN *flooding* dan *ping flooding* pada IoT. Penggunaan FSA ini merupakan pendekatan alternatif yang baru untuk mendeteksi serangan DoS.

Finite-State Automata (FSA) adalah model automata yang mempunyai kemampuan untuk menangkap pola dari suatu data dengan jumlah *state* terbatas karena tidak dilengkapi dengan memori sementara [23]. FSA sering diterapkan dalam pengolahan teks seperti analisis tulisan [24–26], pemisahan kata [27], kesamaan dokumen [28], *grammar* [29], *spelling* [30], dan penerjemah [31], [32]. Beberapa penelitian lain juga menggunakan FSA sebagai sistem pendeteksi/diagnosa [33–37], kontrol/otomatisasi [38–42] dan pengolahan i*mage/ watermarking* [43], [44].

Oleh karena kemampuan FSA untuk menangkap pola tersebut maka FSA diadopsi untuk mengindentifikasi pola SYN *flooding* dan *ping flooding* pada penelitian ini. Gambar 1 merupakan Contoh diagram FSA [45].



Gambar 1. Diagram FSA

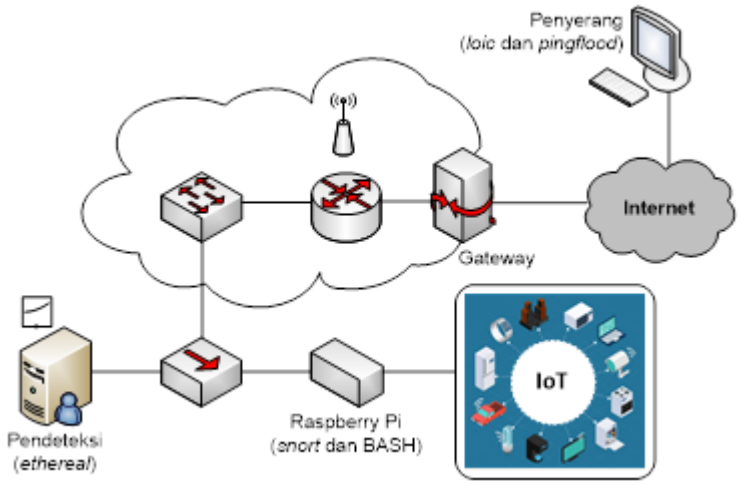
Dimana $\{q0, q1, q2, q3\}$ adalah *state*, $\{q0\}$ merupakan *state* awal, $\{q3\}$ merupakan *state* akhir, dan $\{a, b\} = \sum$ adalah simbol input dengan *read character* adalah babaabaaaba dan *search character* adalah $aaa = i$. Gambar 2 adalah Algoritma FSA sederhana untuk pencarian *string* dengan $\sum = \{a, b\}$ [45].

```
state = 0;
For (i =1,2, ... , n)
    State = (state, Ti)
    If (state == m)
        Macth success in pos i – m + 1; break;
    Else
Match failed;
```

Gambar 2. Algoritma FSA

2. METODE PENELITIAN

Arsitektur jaringan IoT yang terdiri dari *RaspberryPi* (*snort* dan *Bash*), pendeteksi (*ethereal*), dan penyerang (*Ioic* dan *pingflood*) seperti yang ditunjukan pada Gambar 3. Arsitektur jaringan IoT ini merupakan pengembangan dari arsitektur jaringan IoT Chen [46] dan Cui [47].



Gambar 3. Arsitektur Jaringan IoT

2.1. *RaspberryPi* (Raspi)

Merupakan PC mini dengan sistem operasi *Raspbian* yang berfungsi sebagai *relay* untuk mengaktifkan/menonaktifkan perangkat IoT. Raspi ini akan dikendalikan melalui aplikasi *python* berbasis *web*. Nilai 1 yang dikirim ke Raspi akan diterjemahkan oleh aplikasi *web* sebagai indikator untuk mengaktifkan *relay* dan nilai 0 untuk menonaktifkan *relay*. Pada raspi juga di-*install* aplikasi *snort* untuk mengamati aktivitas dalam jaringan (*packet sniffing*) dan *Bash-iptables* untuk men-*dropping* ip *address* penyerang.

Setelah *snort* terpasang maka kode *ipvar HOME_NET any* pada *file snort.conf* di *directory /etc/snort/snort.conf* diganti dengan kode *ipvar HOME_NET* 192.168.0.0/24 yang merupakan *ip address* jaringan yang dipakai pada penelitian ini. Kemudian dibuat dua *rule* pada *file /etc/snort/rules/local.rules* untuk mendeteksi serangan *SYN flooding* dimana *alert* akan aktif jika terdeteksi serangan terhadap *port* 80 yang berisi *flags SYN* lebih dari 3 kali dalam 2 detik seperti yang terlihat pada Gambar 4.

```
alert tcp any any -> $HOME_NET 80 (flags: S; msg:"SYN Flooding terdeteksi"; flow:stateless;
sid:1000002; detection_filter:track by_dst, count 3, seconds 2;)
```

Gambar 4. Rule untuk mendeteksi serangan *SYN flooding*

Rule selanjutnya untuk mendeteksi serangan *ping flooding* dimana *alert* yang berisi *Ping Flooding* terdeteksi akan aktif jika terdeteksi serangan terhadap seluruh *port* dari *HOME_NET* yang memuat *request* ICMP lebih dari 3 kali dalam 1 detik seperti yang terlihat pada Gambar 5. *Rule* ini juga memuat ID, nomor *revisi* untuk mempermudah pemeliharaan *rule*, kategori *rule* sebagai *icmp-event*, dan pendeteksian dilakukan dengan cara melacak alamat *IP* tujuan.

```
alert icmp any any -> $HOME_NET any (msg:"Ping Flooding terdeteksi"; sid:1000001; rev:1;
classtype:icmp-event; detection_filter:track by_dst, count 3, seconds 1;)
```

Gambar 5. Rule untuk mendeteksi serangan ping flooding

2.2. Pendeteksi

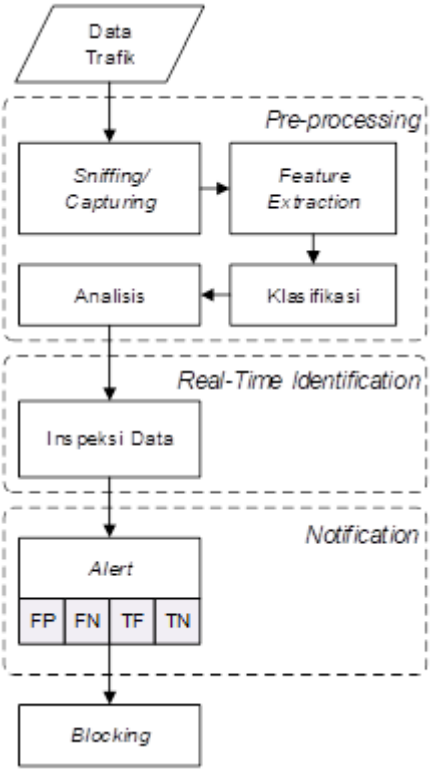
Merupakan PC yang menjalankan aplikasi *Ethereal*. *Ethereal* merupakan sebuah *network/protocol analysis tool* untuk *capturing/sniffing* data trafik. Data hasil *capturing* ini akan dianalisis polanya menggunakan algoritma FSA.

2.3. Penyerang

Merupakan PC yang menjalankan dua *tool* yaitu aplikasi *loic* untuk mengirimkan serangan *SYN flooding* dan aplikasi *pingflood* untuk mengirimkan serangan *ping flooding* ke raspi dengan beberapa tingkat kecepatan serangan.

2.4. *Internet of Things*

Merupakan peralatan IoT yang dihubungkan ke raspi 3. Adapun metode pendeteksian dan pencegahan terhadap serangan DoS dibagi menjadi empat tahap yaitu tahap *preprocessing*, identifikasi, notifikasi dan *blocking* seperti yang terlihat pada Gambar 6.



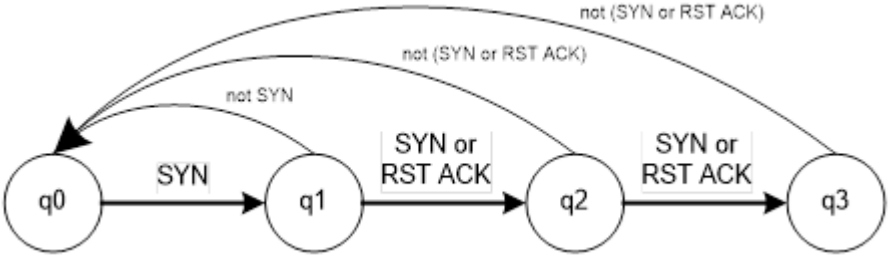
Gambar 6. Tahapan metode pendeteksian dan pencegahan serangan DoS

2.5. *Pre-processing*

Paket data trafik yang telah di-*capture/sniffing* oleh aplikasi *ethereal* akan diekstraksi berdasarkan fitur-fitur yang terdapat di dalam protocol TCP dan ICMP yang terdiri dari *ip address* penyerang, *ip address target*, *flag*, *source port* dan *destination port*. *File* hasil *capturing/sniffing* ini masih merupakan data mentah yang berupa *file pcap*. *File pcap* ini akan diekstraksi menjadi *file* CSV kemudian datanya akan diklasifikasikan tipe serangan. Data hasil klasifikasi akan dianalisis dengan algoritma FSA kemudian hasil analisis akan menjadi *rule* pada *snort* untuk mengidentifikasi serangan DoS (*SYN flooding atau ping flooding*).

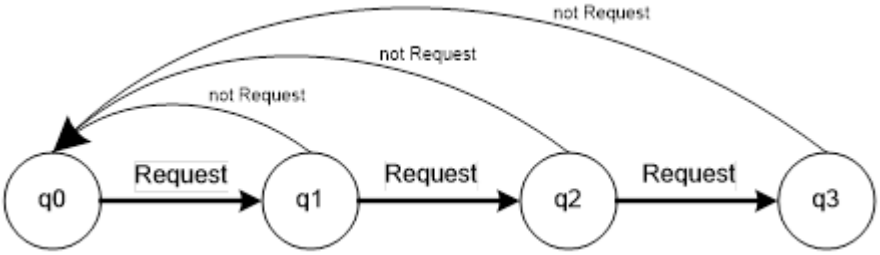
2.6. Identifikasi

Untuk menghindari penumpukan data pada tahap ini maka digunakan teknik TCP *follow stream* untuk *SYN flooding* dan teknik ICMP *follow stream* untuk *ping flooding* sehingga data yang akan diinspeksi menjadi lebih pendek karena data tersebut hanya berupa fitur data yang terindikasi sebagai *state* awal dari *SYN flooding* atau *ping flooding*. Proses identifikasi *SYN flooding* dan *ping flooding* akan dilakukan secara *real-time* dengan menggunakan algoritma FSA oleh aplikasi *snort*. Gambar 7 mengilustrasikan *state transition* dari *SYN flooding* untuk proses pencarian string pada data trafik dengan model FSA. Ketika penyerang mengirimkan rentetan paket SYN menuju *server*, dari *state* awal q0 menuju *state* q3 dan ditransmisikan melalui *state* q1 dan q2. Namun, penyerang tidak pernah membalas paket SYN-ACK yang dikirimkan oleh *server* maka *server* akan penuh dengan antrian paket SYN dan sesi koneksi tetap terbentuk sehingga *server* tidak dapat menerima paket SYN dari pengguna lain. Rentetan paket SYN pada trafik data inilah yang akan diidentifikasi oleh FSA sebagai *SYN flooding* berdasarkan *rule* pada Gambar 4.



Gambar 7. State transition dari SYN flooding

Adapun, state transition dari *ping flooding* untuk proses pencarian *string* pada data trafik dengan model FSA dapat dilihat pada Gambar 8. Penyerang akan membanjiri *server* dengan mengirimkan *request* tanpa perlu sesi koneksi. Dimulai dari *state* awal q0 menuju *state* q3 dan ditransmisikan melalui *state* q1 dan q2 sehingga *server* akan kehabisan sumber daya dalam memberikan *response*. Trafik *request* berulang ini akan diidentifikasi sebagai *ping flooding* oleh FSA berdasarkan *rule* pada Gambar 5.



Gambar 8. State transition dari ping flooding

2.7. Notifikasi

Tahap ini merupakan tahap pemberian notifikasi (*alert*) jika pola paket data teridentifikasi pada sebagai SYN *flooding* dan/atau *ping flooding* berdasarkan *rule* pada aplikasi *snort* yang telah dikonfigurasi pada tahap sebelumnya. Notifikasi akan dibagi menjadi empat *alert* yaitu *False* Positif, *False* Negatif, *True* Positif, dan *True* Negatif.

2.8. *Blocking*

Ketika pola data trafik teridentifikasi sebagai serangan DoS dan notifikasi (*alert*) telah dihasilkan maka untuk mencegah serangan lanjutan dilakukan *blocking* dengan cara men-*droping ip address* penyerang menggunakan *Bash-iptabels* sebagai *rule*. Gambar 9 *Rule* yang digunakan untuk mem-*blocking ip address* penyerang yang melakukan serangan *SYN flooding*.

```
#!bin/sh
#Menghapus semua rule
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
#blok paket syn yang diidentifikasi
iptables -N synflood
iptables -A INPUT -p tcp –syn -j synflood
iptables -A synflood -m limit –limit 1/s –limit-burst 3 -j ACCEPT
iptables -A synflood -j DROP
```

Gambar 9. *Rule* untuk men-*dropping ip address* saat serangan *SYN flooding*

Adapun Gambar 10 *rule* yang digunakan untuk mem-*blocking ip address* penyerang yang melakukan serangan *ping flooding.*

```
#!bin/sh
#Menghapus semua rule
iptables -F
iptables -X
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
#blok paket ping(icmp) yang diidentifikasi
iptables -N pingflood
iptables -A INPUT -p icmp --icmp-type echo-request -j pingflood
iptables -A pingflood -m limit –limit 1/s –limit-burst 3 -j ACCEPT
iptables -A pingflood -j DROP
```

Gambar 10. *Rule* untuk men-*dropping ip address* saat serangan *ping flooding*

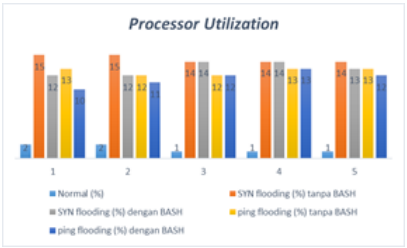
3. HASIL DAN ANALISIS

Ujicoba serangan DoS dilakukan pada *RaspberryPi* 3 Model B yang menggunakan OS Raspbian dengan *prosesor* 1,2 GHz 64 bit Quad-Core ARMv8 dan memori 1 GB SDRAM 400 MHz. Pengodean menggunakan bahasa *Python* dan *Bash*. Hasil Pengujian menunjukan bahwa serangan *SYN flooding* menyebabkan *delay* sedangkan serangan *ping flooding* menyebabkan *error* pada saat mengaktifkan/menonaktifkan peralatan IoT melalui perangkat raspi seperti yang terlihat pada Tabel 1. Adapun pencegahan dengan penerapan *Bash-iptables* berhasil mengurangi serangan *SYN flooding* dengan efisiensi waktu pencegahan sebesar 55,37% dan mengurangi 60% serangan *ping flooding*tetapi dengan waktu yang tidak signifikan.

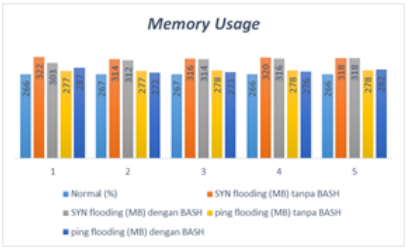
Tabel 1. Hasil pengujian serangan DoS dengan dan tanpa *Bash-iptables*

Normal (detik)	*SYN flooding* (detik)		*Ping flooding* (detik)	
	tanpa *iptables*	dengan *iptables*	tanpa *iptables*	dengan *iptables*
0.60	1.84	0.95	error	error
0.70	1.64	0.84	error	15.5
0.40	1.36	0.86	error	8.5
0.40	1.72	0.92	error	error
0.50	2.38	1.38	error	8.7
0.52	1.79	0.99	-	-

Dampak lain dari serangan DoS adalah peningkatan pemakaian prosesor dan memori seperti yang dapat dilihat pada Gambar 11 dan 12. Hasil pengujian menunjukan bahwa penggunaan prosesor rata-rata meningkat 11-13% dan penggunaan memori rata-rata meningkat 11-51MB. Gambar 11 dan 12 menunjukan bahwa model berhasil mendeteksi serangan *SYN flooding* dan *ping flooding* menggunakan *rule* pada *snort*, namun pencegahan serangan tidak berhasil menurunkan pemakaian prosesor dan memori secara signifikan.



Gambar 11. Processor Utilization



Gambar 12. Memory Usage

4. KESIMPULAN

Serangan DoS (*SYN flooding* dan *ping flooding*) dapat mempengaruhi kinerja jaringan IoT. Serangan *SYN flooding* dapat menyebabkan *delay* dalam sistem komunikasi perangkat dan serangan *ping flooding* dapat menyebabkan *error* dalam sistem komunikasi perangkat. Hasil ujicoba menunjukan bahwa algoritma FSA dapat digunakan untuk mendeteksi dan mengurangi efek serangan *SYN flooding* dan *ping flooding* pada jaringan IoT. Penerapan *Bash-iptables* berhasil mengurangi serangan *SYN flooding* dengan efisiensi waktu pencegahan sebesar 55,37% dan mengurangi 60% serangan ping flooding tetapi dengan waktu yang tidak signifikan. Sementara penggunaan prosesor rata-rata meningkat 11-13% dan penggunaan memori rata-rata meningkat 11-51MB.

Penelitian berikutnya dapat menggunakan algoritma FSA untuk mendeteksi variasi serangan DoS yang lain dan melibatkan metode *machine learning* untuk mengklasifikasi dan mencegah serangan DoS.

REFERENSI

[1] R. Paudel, T. Muncy, and W. Eberle, "Detecting DoS Attack in Smart Home IoT Devices Using a Graph-Based Approach," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, dec 2019, pp. 5249–5258.

[2] Y. Al-Hadhrami and F. K. Hussain, "A Machine Learning Architecture Towards Detecting Denial of Service Attack in IoT," in *Advances in Intelligent Systems and Computing*. Springer, Cham, 2020, pp. 417–429.

[3] A. Bamou, M. Khardioui, M. D. El Ouadghiri, and B. Aghoutane, "Implementing and Evaluating an Intrusion Detection System for Denial of Service Attacks in IoT Environments," in *Lecture Notes in Networks and Systems*. Springer, Cham, 2020, pp. 167–178.

[4] G. R. Andreica, L. Bozga, D. Zinca, and V. Dobrota, "Denial of Service and Man-in-the-Middle Attacks Against IoT Devices in a GPS-Based Monitoring Software for Intelligent Transportation Systems," in *2020 19th RoEduNet Conference: Networking in Education and Research (RoEduNet)*. IEEE, dec 2020, pp. 1–4.

[5] R. Ande, B. Adebisi, M. Hammoudeh, and J. Saleem, "Internet of Things: Evolution and technologies from a security perspective," *Sustainable Cities and Society*, vol. 54, p. 101728, mar 2020.

[6] S. S. Kumar and K. Kulothungan, "An Anomaly Behavior based Detection and Prevention of DoS Attack in IoT Environment," in *2017 Ninth International Conference on Advanced Computing (ICoAC)*. IEEE, dec 2017, pp. 287–292.

[7] W. Ashford, "Businesses blame rivals for DDoS attacks," *Computer Weekly*, mar 2017.

[8] L. Liang, K. Zheng, Q. Sheng, and X. Huang, "A Denial of Service Attack Method for an IoT System," in *2016 8th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, dec 2016, pp. 360–364.

[9] A. Sanmorino and R. Gustriansyah, "An alternative solution to handle ddos attacks," *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 3, pp. 657–667, 2018.

[10] G. B. Gunawan, P. Sukarno, and A. G. Putrada, "Pendeteksian Serangan Denial of Service (DoS) pada Perangkat Smartlock Berbasis Wifi Menggunakan SNORT IDS," *eProceedings of Engineering*, vol. 5, no. 3, pp. 7875–7884, 2018.

[11] R. Hermawan, "Analisis Konsep dan Cara Kerja Serangan Komputer Distributed Denial of Service (DDOS)," *Faktor Exacta*, vol. 5, no. 1, pp. 1–14, 2012.

[12] M. Ge, N. F. Syed, X. Fu, Z. Baig, and A. Robles-Kelly, "Towards a deep learning-driven intrusion detection approach for Internet of Things," *Computer Networks*, vol. 186, p. 107784, feb 2021.

[13] J. Gao, L. Gan, F. Buschendorf, L. Zhang, H. Liu, P. Li, X. Dong, and T. Lu, "Omni SCADA Intrusion Detection Using Deep Learning Algorithms," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 951–961, jan 2021.

[14] T.-H. Lee, L.-H. Chang, and C.-W. Syu, "Deep Learning Enabled Intrusion Detection and Prevention System over SDN Networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, jun 2020, pp. 1–6.

[15] P. Nagar, H. K. Menaria, and M. Tiwari, "Novel Approach of Intrusion Detection Classification Deeplearning Using SVM," 2020, pp. 365–381.

[16] B. Susilo and R. F. Sari, "Intrusion Detection in IoT Networks Using Deep Learning Algorithm," *Information*, vol. 11, no. 5, p. 279, may 2020.

[17] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the IoT," *Journal of Information and Telecommunication*, vol. 4, no. 4, pp. 482–503, oct 2020.

[18] K. Pradeep Mohan Kumar, M. Saravanan, M. Thenmozhi, and K. Vijayakumar, "Intrusion detection system based on GAfuzzy classifier for detecting malicious attacks," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 3, feb 2021.

[19] S. Ramesh, C. Yaashuwanth, K. Prathibanandhi, A. R. Basha, and T. Jayasankar, "An optimized deep neural network based DoS attack detection in wireless video sensor network," *Journal of Ambient Intelligence and Humanized Computing*, jan 2021.

[20] S. Latif, Z. Zou, Z. Idrees, and J. Ahmad, "A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network," *IEEE Access*, vol. 8, pp. 89 337–89 350, 2020.

[21] M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6882–6897, aug 2020.

[22] J. Pacheco, V. H. Benitez, L. C. Felix-Herran, and P. Satam, "Artificial Neural Networks-Based Intrusion Detection System for Internet of Things Fog Nodes," *IEEE Access*, vol. 8, pp. 73 907–73 918, 2020.

[23] K. Zhang, L. Zhang, and L. Xie, "Detectability of Finite-State Automata," in *Communications and Control Engineering*. Springer, Cham, 2020, ch. Discrete-T, pp. 179–192.

[24] A. A. K. O. Sudana, W. P. Buana, and T. Wulandari, "Web-based Implementation of Finite State Automata Method on Lyrics Recognition System of Balinese Song Pupuh," *International Journal of Computer Applications*, vol. 149, no. 4, pp. 32–37, 2016.

[25] Y. Akbari, K. Nouri, J. Sadri, C. Djeddi, and I. Siddiqi, "Wavelet-based gender detection on off-line handwritten documents using probabilistic finite state automata," *Image and Vision Computing*, vol. 59, pp. 17–30, mar 2017.

[26] Y. M. R. Putra, "Sentence Analysis With Artificial Intelligence Machine Learning Using Finite State Automata," *Proxies*, vol. 1, no. 1, pp. 1–6, 2017.

[27] B. Fernandus, "Separating of the Words using Finite State Automata," Ph.D. dissertation, 2016.

[28] M. AbuSafiya, "Measuring Documents Similarity using Finite State Automata," in *2020 2nd International Conference on Mathematics and Information Technology (ICMIT)*. IEEE, feb 2020, pp. 208–211.

[29] C. Kara-Mohamed, A. Hamdi-Cherif, H. Al'Alwi, K. Al-Khalifa, and N. Al-Harbi, "Grammatical Inference System for Finite State Automata - GIFSA," in *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, dec 2016, pp. 1274–1279.

[30] J. M. R. Imperial, C. G. V. Ya-On, and J. C. Ureta, "An experimental Tagalog Finite State Automata spellchecker with Levenshtein edit-distance feature," in *2019 International Conference on Asian Language Processing (IALP)*. IEEE, nov 2019, pp. 240–243.

[31] I. B. T. T. Murti, C. Janis, and I. G. A. Sudhana, "Transliteration Balinese Script using Finite State Automata (FSA) Algorithm," *Journal of Physics: Conference Series*, vol. 1165, no. 1, p. 012002, feb 2019.

[32] P. N. Crisnapati, P. D. Novayanti, G. Indrawan, K. Y. E. Aryanto, and M. S. Wibawa, "Accuracy Analysis of Pasang Aksara Bot using Finite State Automata Transliteration Method," in *2018 6th International Conference on Cyber and IT Service Management (CITSM)*. IEEE, aug 2018, pp. 1–6.

[33] A. Swetapadma and A. Yadav, "An innovative finite state automata based approach for fault direction estimation in transmission lines," *Measurement*, vol. 99, pp. 13–22, mar 2017.

[34] T. Peng, L. Dai, Z. Chen, C. Ye, and X. Peng, "A Probabilistic Finite State Automata-based Fault Detection Method for Traction Motor," in *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*. IEEE, jun 2020, pp. 1199–1204.

[35] F. Settele, A. Weber, and A. Knoll, "Plant Model-Based Fault Detection during Aircraft Takeoff Using Non-Deterministic Finite-State Automata," *Aerospace*, vol. 7, no. 8, p. 109, jul 2020.

[36] C. Bhattacharya, S. Dharmadhikari, A. Basak, and A. Ray, "Early Detection of Fatigue Crack Damage in Ductile Materials: A Projection-Based Probabilistic Finite State Automata Approach," *ASME Letters in Dynamic Systems and Control*, vol. 1, no. 4, oct 2021.

[37] T. Y. Pribadi, K. Handayani, and W. Gata, "Diagnosis of Heart Disease Using Automata Finite State Algorithm," *Techno Nusa Mandiri*, vol. 18, no. 1, pp. 17–24, 2021.

[38] F. J. Kaunang and J. Waworundeng, "Implementation of Finite State Automata in an Amusement Park Automatic Ticket Selling Machine," *Abstract Proceedings International Scholars Conference*, vol. 7, no. 1, pp. 1776–1785, 2019.

[39] K. Handayani, D. Ismunandar, S. A. Putri, and W. Gata, "Penerapan Finite State Automata pada Vending Machine Susu Kambing Etawa," *MATICS*, vol. 12, no. 2, pp. 87–92, mar 2021.

[40] E. Erni, F. Titiani, S. A. Putri, and W. Gata, "Penerapan Konsep Finite State Automata Pada Aplikasi Simulasi Vending Machine Jamu Tradisional," *Jurnal Informatika*, vol. 7, no. 2, pp. 141–147, sep 2020.

[41] L. Sauer, D. Henrich, and W. Martens, "Towards Intuitive Robot Programming Using Finite State Automata," 2019, pp. 290–298.

[42] M. Jaluvka, E. Volna, and M. Kotyrba, "Motion Controlling Using Finite-State Automata," 2019, pp. 455–464.

[43] R. Karmakar, S. S. Jana, and S. Chattopadhyay, "A Cellular Automata Guided Finite-State-Machine Watermarking Strategy for IP Protection of Sequential Circuits," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–1, 2020.

[44] R. O. Reddy, K. D. Dhruve, R. N. Reddy, M. Radha, and N. S. Vani, "A Novel Approach in Adopting Finite State Automata for Image Processing Applications," *International Journal of Computer Vision and Image Processing*, vol. 8, no. 1, pp. 59–74, jan 2018.

[45] S. A. Valianta, "Identifikasi Serangan Port Scanning dengan Metode String Matching," *Annual Research Seminar (ARS)*, vol. 2, no. 1, pp. 466–471, 2016.

[46] Q. Chen, H. Chen, Y. Cai, Y. Zhang, and X. Huang, "Denial of Service Attack on IoT System," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, oct 2018, pp. 755–758.

[47] Y. Cui, Q. Liu, K. Zheng, and X. Huang, "Evaluation of Several Denial of Service Attack Methods for IoT System," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, oct 2018, pp. 794–798.