Laman web jurnal: https://ejournal.unama.ac.id/index.php/processor

Processor: Jurnal Ilmiah Sistem Informasi, Teknologi Informasi dan Sistem Komputer

P-ISSN: 1907-6738 | E-ISSN: 2528-0082

Pengenalan Pola Serangan pada Internet of Thing (IoT) Menggunakan Support Vector Mechine (SVM) dengan Tiga Kernel

[1]Tasmi, [2]Ferri Antony, [3]Dhamayanti, [4]Herri Setiawan, [5]Ahmad Fali Oklilas, [6]Panji Asmoro, [7]Taufik Hidayat

[1267] Program Studi Sistem Komputer Fakultas Ilmu Komputer dan Sain, Jl. Jendral Sudirman No.629 Km.4 , Palembang dan 30129, Indonesia
[3] Program Studi Sistem Informasi Fakultas Ilmu Komputer dan Sain, Jl. Jendral Sudirman No.629 Km.4 , Palembang dan 30129, Indonesia
[4] Program Studi Teknik Informatika Fakultas Ilmu Komputer dan Sain, Jl. Jendral Sudirman No.629 Km.4 , Palembang dan 30129, Indonesia
[5] Program Studi Sistem Komputer Fakultas Ilmu Komputer, Jl. Palembang-Prabumulih Km.32 , Indralaya dan 30862, Indonesia

ABSTRACT

Internet of things (IoT) technology is very popular these days around the world, with the development of IoT technology raises the impact of security threats and attacks on IoT devices. One of the most is the theft of data and information, one form of threat in IoT is malware. This research uses attacks in the form of bontet to detect attacks on the Internet of Things IoT) and uses Machine Learning to perform data detection of attacks on IoT devices. The method used in this research is Support Vector Mechine (SVM) by comparing three kernels namely Liner, polynominal and Radial Basis Function (RBF). This method is used to determine the level of accuracy in the detection process and compare between kernels. The results obtained are the accuracy value of 0.997 for the liner kernel, meaning that this kernel is able to separate the classes well, while the Polynomial kernel accuracy value of 0.993 is good in separating classes even though the value is smaller than the liner. Meanwhile, the RBF (Radial Basis Function) kernel has an accuracy of 1.0 (100%).
Keywords: Machine Learning, Internet of things (IoT), malware, Support Vector Mechine (SVM)

ABSTRAK

Internet of things (IoT) teknologi yang sangat populer akhir-akhir ini di seluruh dunia, Dengan berkembangnya teknologi IoT ini memunculkan dampak ancaman keamanan dan serangan terhadap perangkat IoT. Salah satu yang paling banyak adalah pencurian data dan informasi, salah satu bentuk ancaman dalam IoT adalah malware. Dalam penelitian ini menggunakan serangan dalam bentuk bontet untuk mendeteksi serangan pada Internet of Things IoT) dan menggunakan Machine Learning untuk melakukan deteksi data terhadap serangan pada perangkat IoT. Metode yang digunakan dalam peneltian ini adalah Support Vector Mechine (SVM) dengan membandingkan tiga kernel yaitu Liner, polynominal dan *Radial Basis Function* (RBF). Metode ini digunakan untuk mengetahui tingkat akurasi pada proses deteksi dan melakukan pebandingan antara kernel. Hasil yang didapatkan nilai akurasinya 0.997 untuk kernel liner artinya kernel ini mampu dengan baik untuk memisahkan kelas dengan baik, sedangan kernel Polynomial nilai akurasinya sebesar 0.993 ini hasilnya baik dalam memisahkan kelas walaupun nilai lebih kecil dari liner. Sedangkan Kernel RBF (*Radial Basis Function*) memiliki akurasi sebesar 1.0 (100%)
Keywords: Machine Learning, Internet of things (IoT), malware, Support Vector Mechine (SVM)

1. PENDAHULUAN

Internet of things (IoT) adalah jaringan cerdas yang menghubungkan semua hal ke internet dengan tujuan untuk bertukar infomasi dengan menggunakan aturan (*rule*) atau sering disebut juga *protocol* yang disepakati dengan harapan siapapun dan dimanapun bisa mengakasesnya [1]. Menurut [2] setiap perangkat yang terhubungan kejaringan IoT memiliki kemampuan untuk berkomunikasi, bekerja sama, memproses, dan mengirim data secara mandiri untuk membangun sebuah sistem yang dapat memantau dan mengontrol. Hal ini diperkuat oleh penelitian [3] yang menyatakan bahwa perkembangan IoT sangat signifikan pada *home and business application* dalam hal meningkatkan kualitas hidup dan menunjang perekonomian dunia. Sebagai contoh dalam bidang pendidikan, kesehatan, industri, dan *smart home.*

Akan tetapi, sebuah sistem IoT yang terhubung kejaringan global atau internet memiliki banyak tantangan yang dihadapi dalam pengembangannya seperti *standardization, power and energy efficiency, Big Data, security dan privacy, intelligence, integration methodology, pricing, network communications, storage dan scalability serta flexibility* [4]. Salah satu isu yang cukup berkembang adalah *security*, hal tersebut dikarenakan dapat merusak objek dari IoT dan juga dapat merugikan pihak pengguna IoT, seperti kehilangan data yang rahasia pada sebuah *account* transaksi, Selain itu, masalah keamanan dapat mempengaruhi kemampuan *device* dalam melakukan komunikasi data. Terdapat empat bagian masalah sistem *security* di IoT yaitu: (*i*) *Confidentialy, (ii*) *Integrity, (iii) Availability, dan (iv) Authenticity* [5]

Salah satu serangan yang paling serius di IoT adalah *malware*, pada penelitian terdahulu [6] menyebutkan *malware* adalah sebuah aplikasi jahat yang dirancang khusus untuk melakukan aktivitas berbahaya atau perusak perangkat lunak pada komputer seperti virus, trojan, dan lain-lain yang disebar melalui jaringan internet. Para penyerang sudah sering menggunakan jenis serangan ini yang dapat menimbulkan kerugian pada perangkat [7], selain itu serangan ini juga dapat mengambil alih kontrol alat dan mengambil data sensitive, [8][9].

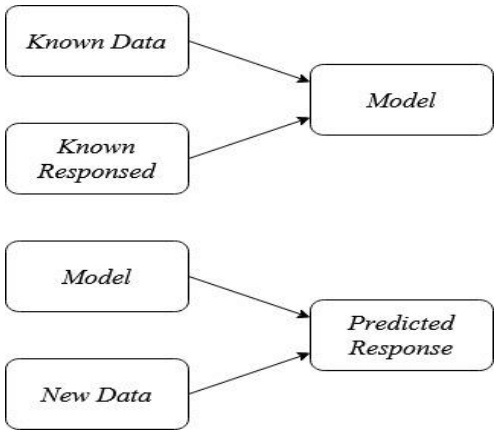
Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

Penelitian ini bertujuan untuk menghasilkan dataset yang didapat dari kondisi *real* pada jaringan IoT sehingga dataset ini dapat dijadikan sumber *learning* untuk model IDS yang akan dikembangkan dengan metode *machine learning*. Penelitian ini juga akan melakukan analisis serangan *malware* untuk memperoleh akurasi yang baik, serta mengukur kinerja jaringan IoT sebelum dan sesudah serangan *malware*. Untuk mencapai tujuan penelitan, maka akan menerapkan *machine learning* guna mendeteksi serangan pada jaringan IoT. Dengan demikian, hal tersebut digunakan untuk menjawab masalah berikut: (i) Bagaimana proses membagun sistem IoT yang *real* yang akan digunakan untuk pembuatan dataset, (ii) Bagaimana proses ekstraksi dataset *malware* untuk mendapat pola serangan dan mengurangi dimensi dataset dengan membuang fitur tidak relevan, dan (iii) Bagaimana mengukur akurasi deteksi *malware* dengan *machine learning* dalam menentukan pola serangan.

2. TINJAUAN PUSTAKA

2.1. Machine Learning dan SVM

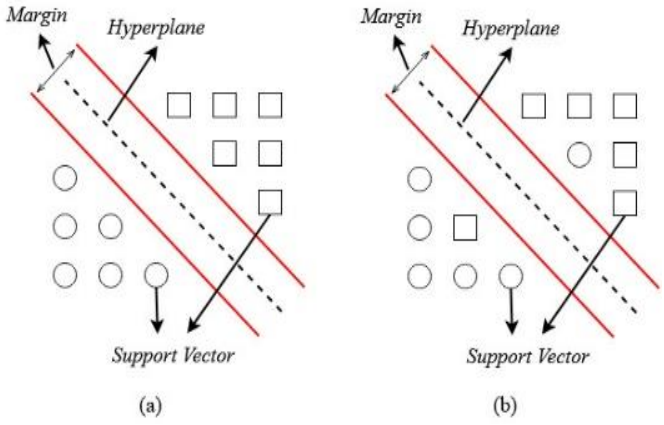
Machine Learning atau pembelajaran mesin merupakan salah satu kecerdasan buatan yang memungkinkan mesin melakukan pembelajaran berdasarkan contoh data, pembelajaran mesin memanfaatkan hubungan antar variabel dan probabilitas untuk menghasilkan prediksi [10].Berdasarkan masukan dan keluaran yang diharapkan, pembelajaran mesin terbagi menjadi dua kelompok yaitu : *Supervised Learning*, merupakan pembelajaran yang bertujuan untuk memetakan masukan dan keluaran yang diinginkan seperti pada pengelompokkan.



Gambar 1. *Deskripsi Supervised Learning* [11]

Gambar 1 menjelaskan prinsip kerja *Supervised Learning* yaitu dengan mempelajari sekumpulan contoh masukan dan keluaran dan menghasilkan sebuah model yang mampu memetakan masukan yang baru menjadi keluaran yang tepat. Terdapat beberapa algoritma yang dapat diterapkan antara lain, *Naïve Bayes, SVM* dan *Decision Tree. Unsupervised Learning*, merupakan pembelajaran yang memodelkan himpunan masukan untuk mempelajari dan mencari pola-pola tertentu pada masukan yang diberikan, *Clustering* atau penggolongan merupakan penerapan dari pembelajaran ini. Beberapa algoritma pada pembelajaran ini antara lain, *Birch, Cure* dan *K-Mean* [11].

Support Vector Machine (SVM) merupakan salah satu metode yang terdapat pada *Supervised Learning*, metode *SVM* pada penerapannya terdiri dari 2 tahapan yaitu *Training dan Testing.* Metode *SVM* bertujuan untuk membangun sebuah model yang mampu memprediksi data uji ke sebuah data baru dengan mempelajari hubungan antar fitur yang dilambangkan dengan x_i $(x \epsilon X)$ dan penentuan kelas yang dilambangkan dengan y_i $(y \epsilon Y)$, $F = f(x_i) = h_\Theta(x_i)$ merupakan fungsi pemetaan hubungan dimana Θ merupakan bobot matriks dari fungsi pemetaan. Untuk menentukan nilai *F*, harus didapatkan matriks Θ terlebih dahulu.



Gambar 2**.** (a). *Linear Classification* (b). *Non-Linear Classification* [12]

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

Gambar 2. menjelaskan jika data bersifat linier, *SVM* akan mencari linier *Hyperplane* untuk membagi data menjadi kelas-kelas biner dengan margin maksimum. (b). menjelaskan jika data bersifat tidak linier, maka dibutuhkan perubahan fitur awal ke ruang dimensi yang lebih tinggi. *SVM* menyediakan 4 kernel yaitu : *Linear Kernel, Polynomial Kernel, Radial Basis Function (RBF) Kernel* dan *Sigmoid Kernel* yang dapat digunakan dalam memecahkan permasalahan data yang tidak linier [12]

2.2. *Penelitian Terdahulu*

Telah banyak penelitian yang dilakukan dalam mencari dan menyelesaikan masalah serangan pada jaringan IoT, salah satunya menggunakan *machine learning* dalam deteksi serangan. Dalam pengembangan sistem deteksi serangan yang berbasis *machine learning* terdapat dua hal penting, yaitu dataset dan mengenali pola serangan tersebut [13]. Pada penelitan yang dilakukan oleh [14] menyebutkan ada dua cara dalam deteksi *malware* dengan *machine learning* yaitu: (1) *Static Analysis* adalah proses ekstraksi dataset *malware* yang telah dibuat dan (2) *Dynamic Analysis* adalah deteksi secara *real-time* dalam mendeteksi data serangan, sedangan teknik yang digunakan dalam deteksi *malware* menurut [15] adalah (i) *Signature-based detection*, (ii) *Behavioural-based detection*, dan (iii) *Heuristic-based detection.*

Penelitian oleh [16] menggunakan *machine learning* seperti *Support Vector Machine* (SVM), *logistic regression* serta *random forest* dalam menganalisis pola akses memori untuk deteksi *malware* pada jaringan IoT. Pekerjaan yang dilakukan oleh [17] menggunakan *graph of operational codes* (OpCode) yang digabung dengan metode *Power Iteration* yang di *embedded* ke dalam *engine* dan juga menggunakan KNN dan SVM untuk mengklasifikasi serangan *malware*. Penelitian yang dikerjakan oleh [18] melakukan serangan *brute-force* dengan merancang prototipe alat yang melakukan serangan kode autentikasi secara otamatis tanpa meng-*update firmware*. Penelitian [19] menggunakan metode baru untuk mendeteksi malware PDF menggunakan visualisasi gambar dengan melakukan proses ekstraksi file dan juga menggunakan metode *Random Forests, Decision Trees and K-Nearest Neighbou*r untuk deteksi file yang terinfeksi malware. Penelitian yang dilakukan oleh [20] melakukan deteksi serangan malware pada IoT dengan menggunakan *machine learning* dan *deep learning* dimana hasil yang didapatkan lebih baik dari hasil peneltian sebelumnya. Namun demikian, hasil tersebut hanya mendapat nilai akurasi yang masih rendah. Penelitian [21] mendeteksi *malware* menggunakan pendekatan *low access memory* dan juga melakukan proses klasifikasi delapan kelas dengan metode SVM dan KNN. Dengan semakin banyak nya jenis malware yang mampu merusakan sistem IoT, maka sejalan dengan para peneliti dibidang keamaan IoT melakukan perbaikan dalam deteksi malware pada IoT berbasis Deep Learning, penelitian pada tahun 2022 yang dilakukan oleh [22] mendeteksi malware pada IoT CCN Inception-v3 untuk mengidentifikasi malware perangkat IoT dengan memanfaatkan tampilan gambar malware berwarna dari Android Dalvik Executable File (DEX), pada sebuah lapoan ilmiah (*scientific reports*) yang dibuat oleh [23] mereka pengembangan arsitektur baru untuk mendeteksi malware pada perangkat *Internet of Things* (IoT) yang menggunakan pendekatan berbasis *Convolutional Neural Network* (CNN), penelitian ini menggunakan arsitektur iMDA mempelajari kumpulan fitur yang beragam. Penelitian yang dibuat oleh [24] mendeteksi malware pada protokol *IP Flow Information Export* (IPFIX) dengan membuat kombinasi *machine learning* dan *Manufacturer Usage Description* (MUD) dengan tujuan mengurangi atau memperkecil nilai *false positive* sehingga nilai *true positive rate* menjadi tinggi. Penelitian untuk deteksi malware untuk menghasilkan akurasi yang tinggi juga dilakukan oleh [25] dengan *graph embedding* dan fitur struktural pada proses ekstraksi yang mengabungkan fitur lokal dan global dari *function-call graphs* untuk meningkatkan keandalan deteksi malware. Deteksi malware dengan model CNN juga dilakukan oleh [26] dengan pendekaatan analisis dinamis yang lakukan dalam tiga tahap yaitu ekstraksi fitur, merubah fitur menjadi image dan klasifikasi, hasilnya pendekatan ini memberikan cara yang efekti dibandingkan dengan model deteksi malware lainnya

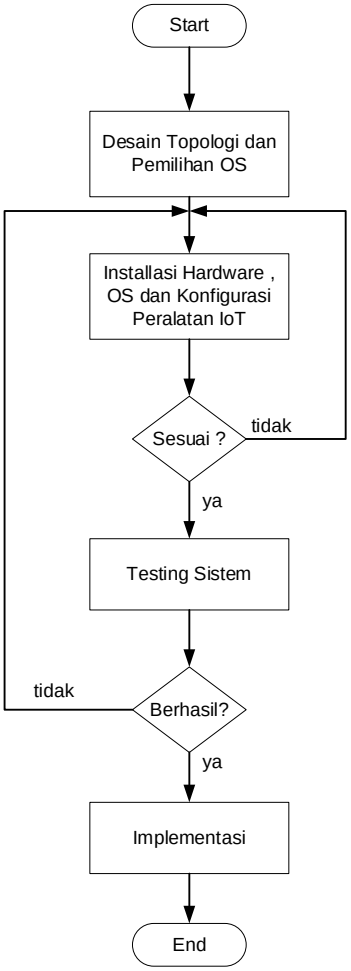
3. METODOLOGI PENELITIAN

3.1 Peracangan Sistem

Tahap pertama terdiri dari dua kegiatan utama, yaitu membangun sistem IoT yang terdiri dari tiga input yang suhu, kelembaban dan kualitas udara, selanjutnya konfigurasi dan koding sistem IoT. Hasil yang diharapkan dari tahap ini adalah untuk menghasilkan sistem monitoring dari tiga kondisi ditas. Gambar 1 menyajikan secara rinci kegiatan pada tahap pertama, adapun langkah-langkah yang dilakukan adalah:

1. Menentukan jenis sensor yang sesuai dengan penelitian, serta sistem operasi dan aplikasi yang dapat digunakan dalam sistem IoT. Penelitian ini kami menggunakan dua buah sensor yaitu Sensor DHT11 yang digunakan untuk deteksi suhu dan kelambaban, sensor MQ-2 serta ESP32 sebagai pusat kendali dalam sistem IoT
2. Pemasanga dan konfigurasi perangkat seperti (i) perakitan perangan yaitu sensor, Wifi, ESP32, Komputer Server yang akan digunakan untuk penyimpanan data hasil monitoring, (2) Setting IP address, (3) Mengaktifkan *WebServer* yang akan digunakan untuk menampilkan visualisasi data monitoring (4) Setting Mysql Server unttuk membuat db yang akan digunakan untuk menampung data hasil monitroing
3. Melakukan testing setiap peralatan yang digunakan untuk menyakinkan bahwa sistem berjalan dengan baik,
4. Memeriksa apakah sistem berjalan dengan baik, jika ada konfigurasi tidak berjalan dengan baik atau tidak berfungsi maka kembali ke langkah tiga.

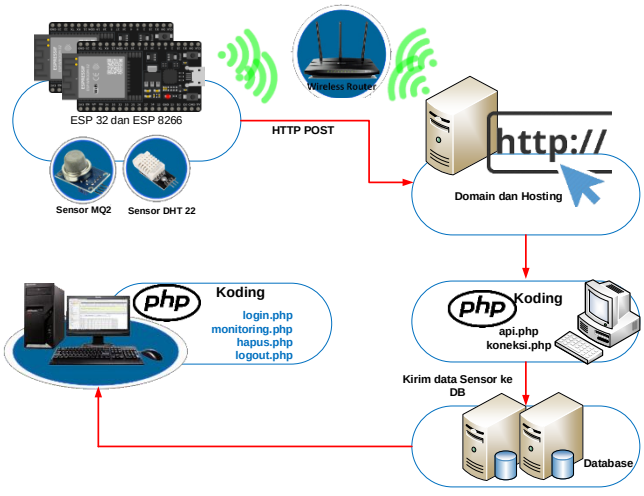
Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023



Gambar 3. Tahapan dalam pengembangan Sistem IoT

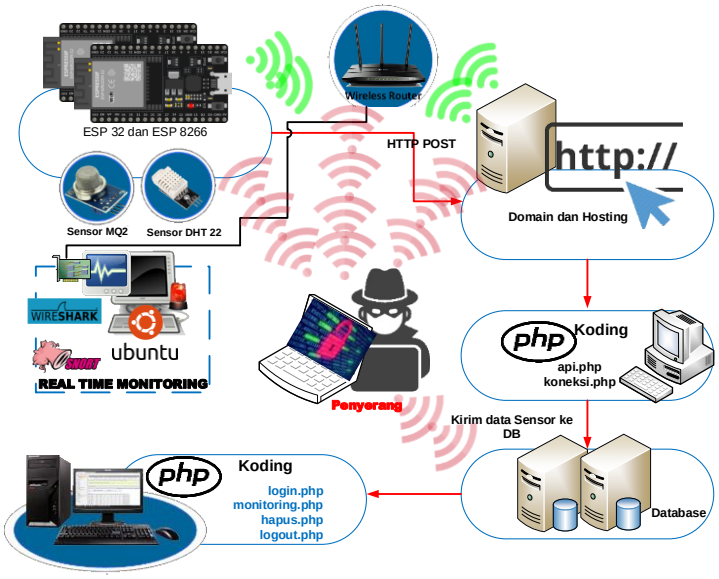
3.2 Peracangan Sistem

Dalam penelitian ini sistem monitoring dilakukan pada laboratorium robotika Prodi Sistem komputer Fakultas Ilmu Komputer Universitas Indo Global Mandiri Palembang. Sistem yang dibangun disajikan pada gambar 4 dimana ada empat kegiatan besar yang dilakuakn yaitu 1) Sistem pusat monitoring, 2) Membuat Domain dan db, 3) mengkonfigurasi wifi sebagain media transmisi data, sistem Application Programming Interface (API) dan sistem koneksi db ke sistem php, 4) dan sistem monitoring real-time dengan berbasis php



Gambar 4. Sistem Monitoring Suhu, Kelembaban dan Kualitas Udara

Setelah sistem berjalan dengan baik tahap selanjutnya menambahkan dua peralatan seperti gambar 5 yang akan digunakan sebagai *sniffing* paket dalam sistem IoT dan sebagai penyeranga dalam penelitian ini



Gambar 5. Sistem Monitoring Suhu, Kelembaban, Kualitas Udara dan Serangan

Tabel 1 dan Tabel 2 merupakan kebutuhan perangkat keras dan perangkat lunak yang digunakan dalam penelitian

Tabel 1 Spesifikasi Kebutuhan Perangkat Keras

Perangkat	**Spesifikasi**	**Konfigurasi**
ESP 32 **IP : 192.168.0.101**	Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz. 520 KB of SRAM, 448 KB of ROM and 16 KB	Koding kontrol untuk membaca inputan sensor dan sebagai input yang akan dikirim ke server, dengan kata lain alat ini sebagai pusat kontrol dalam sistem IoT
Ubuntu Server **IP: 192.168.0.103**	Intel *Core* i5	Web server, Mysql-Server, Open-SSL, DNS server
Wifi **IP : 192.168.0.1**	WR840N (300mbps)	Konfigurasi AP
Sensor DHT11	Tegangan kerja 3.3V-5V, Arus 2.5mA, kelembaban 20%-80% dan suhu 0°C-50°C.	Sensor suhu dan kelembaban
Sensor MQ135	Tegangan input: 5V DC, Arus 150mA, DO output: TTL digital 0 (0.1V) dan 1 (5V) dan AO output: 0.1 ~ 0.3 V	Sensor Kualitas udara
Kali Linux **IP: 192.168.0.104**	Intel *Core* i7,	Konfigurasi tool serangan ke sistem
Windows 10 : **192.168.0.105**	Intel *Core* i3,	Wireshark dan TCPDump digunakan untuk sniffing traffik

Tabel 2 Spesifikasi Kebutuhan Perangkat Keras

Sistem	***Tools/ Framework***	**Keterangan**
Real_time Monitoring	TCPDump dan Wireshark	
Web Server	Apache, *PHP, dan Mysql Server*	
SSL	Open-SSL	

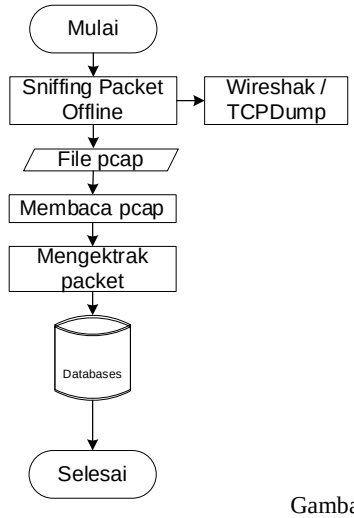
Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

3.3. Sniffing Paket

Pada mesin unix ataupun windows umumnya telah memiliki sistem untuk melakukan capturing data seperti *TCP Dump*, data yang diperoleh akan disimpan dalam bentuk data mentah dan kemudian dilakukan ekstraksi untuk memudahkan dalam proses Analisa. Sniffing paket adalah sebuah aplikasi yang digunakan untuk memantau trafik pada sebuah jaringan melintas. Pada penelitian ini menggunakan tool wireshak untuk menangkap trafik normal dan serangan. Penelitian ini juga akan dikondisikan satu komputer yang melakukan aktifitas serangan, lama waktu capturing serangan dari mesin attacker yang berjalan secara serentak dengan mesin komputer dari pengguna yang sah

3.4. Ekstraksi Fitur dan Labeling

Proses ekstraksi fitur dibutuhkan untuk melakukan analisa terhadap paket data untuk mencari pola dan signature dari serangan port scanning. Proses ekstraksi fitur yaitu proses konversi pada paket data, data yang masih berbentuk data mentah hasil dari capturing diubah kedalam bentuk clear text, sehingga proses analisa kedepan akan lebih mudah. Gambar 4 merupakan pseudo-code sederhana untuk mengkonversi raw data ke dalam format csv.




```
Procedure WRITETOCSV
INPUT
READ_DATA ← Open_ pcap
WRITE_DATA ← Write_toFileCSV
PROT ← TCP
BEGIN
        WHILE READ_DATA ←1
            FOR each row on READ_DATA
                FOR each column on  READ_DATA
                    IF PROTOCOL position (row, column) is occupied THEN
                        IF PROT ←  1 THEN
                            WRITE_DATA
                        END IF
                        ELSE RETURN 0
                    END IF
                    ELSE RETURN 0
```

Gambar 6. Flowchrat dan Pseudo-code proses ekstraksi data

3.5 Pengenal Pola dan Pengujian Serangan di IoT.

Pendeteksian suatu serangan pada sistem IoT dapat dilakukan salah satunya dengan pencarian pola atau pattern dari serangan yang akan dideteksi. Pola serangan biasanya memiliki atribut yang unik yang bisa digunakan sebagai parameter, seperti contohnya Flag, besar data, ip source, ip destination dan range waktu. Pada penelitian ini ada beberapa poin yang dapat digunakan dalam pengenalan pola serangan pada jaringan IoT,diantaranya sebagai berikut :

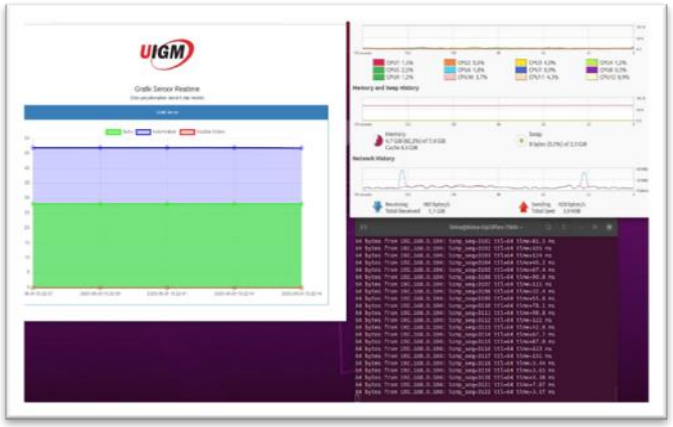
1. Koleksi data dengan File dataset format packet capture (PCAP) yang telah dibuat dalam beberapa skenario.
2. Feature extraction yang akan menghasilkan output beberapa atribut yang disimpan kedalam beberapa variabel dan disimpan ke file dengan csv.
3. Preprocessing data adalah tahapan dimana dilakukan proses pembersihan data, mencari data yang menyimpang (outlier), mencari data yang kosong dan normalisasi data dan balancing data
4. Split data untuk membagi dalam proses traning dan testing data dengan komposisi sebagai berikut 70 persen untuk training dan 30 persen untuk tersing dan komposisi kedua 80 persen untuk training dan 20 persen
5. Prediksi model dan evalusi model dengan mengguanak machine learning untuk mendapat nilai confusion matrix, acuracy, recall

4. HASIL DAN PEMBAHASAN

4.1 Sistem **Io**T

Setelah di rancang topologi Internet of Things (IoT) sedemikian rupa sesuai desain pada gambar 4 dan 5 untuk pembuatan dataset, dimana setiap end-node memiliki satu buah sensor yang akan dikirimkan ke dashboard monitoring. Penelitan [27] melakukan penelitian pengukuran suhu ruangan dengan DHT11 dan sensor WSN, selanjut penelitian [28] membangun sistem monitoring suhu dan kelembaban dengan membandingkan Arduino UNO dengan raspberry serta sensor DHT11 dengan hasil akurasi 95.85% dan loss 4.14% 88,678% dan 11,322% yang dihasilkan oleh raspberry pi. Hasil yang didapatkan penelitian ini disajikan pada gambar 5 dengan akurasi sebenar 94.4% dan 5.42% untuk paket yang tidak tercapture.

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023



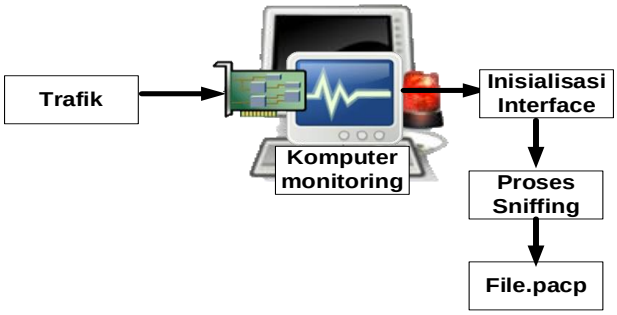
id	suhu	kelembaban	udara
1238	28.5	45.3	0
1239	28.6	45.2	0
1240	28.6	45.2	0
1241	28.6	45.2	0
1243	28.7	45.3	28
1244	28.7	45.3	0
1245	28.7	45.3	0
1246	28.7	45.3	10
1247	28.7	45.3	55
1248	28.7	45.4	46

Gambar 7. Hasil visual dan capture data sensor DHT11 dan MQ135

4.2. Packet Sniffing

Sniffing paket adalah sebuah aplikasi yang digunakan untuk memantau trafik pada sebuah jaringan melintas. Menurut [29] Sniffing paket adalah bagian dari sebuah *hardware* atau *software* yang digunakan untuk memantau semua trafik di jaringan. Sedangan menurut [30] informasi yang didapat dari sniffing paket dapat digunakan oleh seorang administrator untuk mengidentifikasi paket yang aneh dan juga untuk menjaga transmis data pada jaringan agar tetap efisien. Penelitian [31] melakukan sniffing paket dengan parameter Media Access Control (MAC) Address sumber dan tujuan, Ethernet Addresses, Ip Addresses User Datagram Protocol (UDP) dan Hypertext Transfer Protocol (HTTP), sedangkan penelitian yang dilakukan oleh [32] Jumlah user yang mengakses intenet, bandwidth serta besar paket yang digunakan

Merujuk pada penelitian sebelumnya, penelitian ini juga menghasilkan dataset dengan cara sniffing paket dengan menghasil 16 atribut yang terdiri dari waktu, protokol, IP sumber dan tujuan, port Sumber dan tujuan, ACK, SEQ, WIN, Flags, TTL, IP Length, IP_Ceksum, IP_ID, IP_OFF, TCP/UDP. Untuk proses sniffing paket penelitian ini tersaji pada gambar 8, dimana komputer yang didefiniskan sebagai capture paket data harus terlebih dahulu ini mendefenisikan *interface* yang digunakan, dan *output* menghasilkan *"file.pcap"*.



Gambar 8. Proses sniffing paker data

Tabel 3 Hasil Paket Sniffing berdasarkan Protokol komunikasi

Percobaan	Jumlah Paket	Data File Size (MB)	TCP	UDP	ICMP
1	723032	48.9	722840	139	13
2	398127	544	397629	176	57
3	644145	910			

4.3 Feature Extraction

Setelah melakukan proses sniffing paket data selanjutnya dilakukasn proses *field-field* hasil dari sniffing, hasil dari proses ini menghasilkan data mentah (*raw data*) yang sulit di mengerti oleh manusia (*human readable*) ini dikerenakan header IPv4 mempunyai struktur unik dan juga protokol-protokol yang mempunyai lapisan-lapisan yang tersembunyi serta proses *encapsulated*. Dengan pengembang semua program yang berfungsi membaca *raw data* dengan harapkan menghasilkan pola paket normal dan serangan, hasil dari proses ini akan mengahasilkan file.csv, dan nantinya file .csv akan digunakan untuk proses training dan testig karena dengan tipe file ini akan lebih memudahkan dalam proses training dan testing

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

Penelitian sebelumnya yang dilakukan oleh [33] menyatakan ada dua bagian pada panjang variabel paket data, pertama adalah header yang berisi antara 20 sampai 60 byte yang berisi tentang informasi paket routing dan yang kedua adalah *payload* yang berisi konten dari sebuah paket data. Sedangkan [34] yang berhasil mengekstraks paket sebanyak 7 fitur yaitu, (1) Jumlah paket UDP, (2) Jumlah Koneksi Host, (3) Jumlah Paket ICMP, (4) Jumlah SYN yang salah, (5) Windows Size, (6) Flag dan (7) Jenis Layanan, serta [35] yang berhasil mengekstraks semua dalam header IP, TCP, UDP, ICMP, ARP dan IGMP

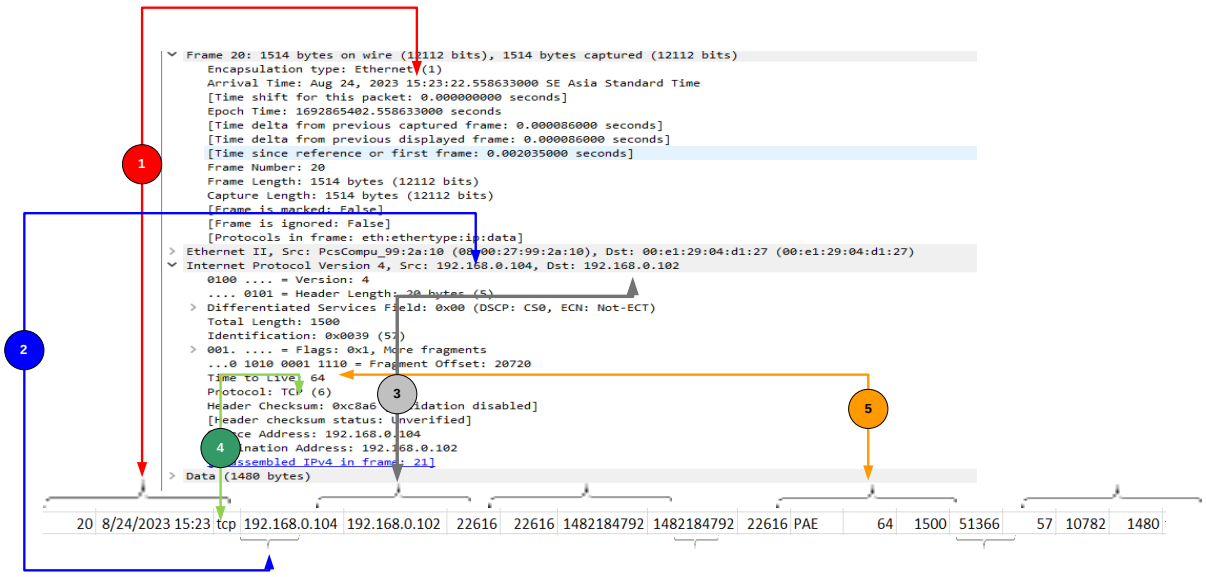
Tabel 4 merupakan enam belas attribut attribut yang dihasilkan *feature extraction*, dari enam belas attribut yang dihasilkan dua attribut yaitu Prototol dan Label nanti akan digunakan untuk validasi data paket normal dan serangn pada sistem IoT.waktu, protokol, IP sumber dan tujuan, port Sumber dan tujuan, ACK, SEQ, WIN, Flags, TTL, IP Length, IP_Ceksum, IP_ID, IP_OFF, TCP/UDP

Tabel 4 Attribut hasil *feature extraction*

No	Attribut
1	*no_packets*
2	*protokol*
3	*timestemp*
4	*ip_src*
5	*ip_dst*
6	*TTL*
7	*flag*
8	*seq_number*
9	*windows*
10	IP Length
11	*Port sumber*
12	*Port Tujuan*
13	IP_Ceksum
14	IP_ID
15	IP_OFF
16	TCP/UDP
17	ACK

4.4 Koreksi Hasil Pengujian Feature Extraction

Untuk menguji keberhasilan dari program yang dibangun untuk proses *feature extraction*, maka dalam penelitian ini dilakukan validasi data antara program *feature extraction* dengan *raw data (pcap)* dengan menggunakan tool *wireshak*. Gambar 8 menyajikan kolerasi antara hasil *feature extraction* dengan *Real Time Monitoring system*



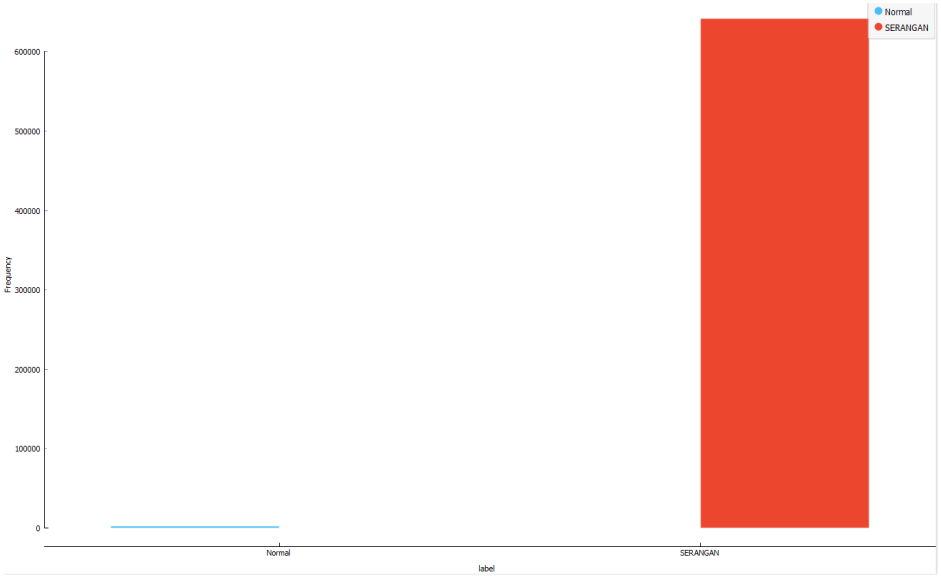
Gambar 9. Validasi Data *Feature Extraction* dengan file pcap

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

Pada setiap gambar harus diberikan keterangan di bawah gambar. Keterangan pada tabel diberikan di atas tabel. Keterangan dituliskan dengan huruf kecil kecuali pada karakter pertama pada tiap kalimat. Seluruh gambar harus diberi penomoran secara berurutan. Gambar diletakkan di tengah halaman (*center aligned*), sedangkan tabel diawali di pinggir kiri (*left aligned*) halaman.

4.4 Pengujian dengan Machine learning

Pada hasil tahap awal ini penelitian ini belum melakukan proses preprocessing dan balancing data dikarena mau melihat sejauh mana data yang didapat menghasilkan akurasi yang baik. Tahap ini digunakan algoritma Support Vector Machine (SVM), menurut penelitian [36] *SVM* adalah pada waktu proses yang sangat berpengaruh pada besarnya data, semakin besar data maka waktu proses akan semakin lama. Data awal yang diuji pada hasil ini adalah percobaan tiga dengan klasfikasi sebagai berikut untuk data serangan 641736 sedangkan data normal sebesar 2295



Gambar 10. Perbandingan data serangan dan normal

Dari data sebesar 398000 baris ada sekitar 11,3 persen terdapat missing value artinya pada tahap selanjutnya akan dilakukan proses pre processing terutama untuk menghilangkan nilai yang kosong. Pada tahap pengujian awal ini dilakukan perbandingan tiga kernel yaitu Liner, Sigmod dan *Radial Basis Function* (RBF). Penelitian sebelumnya yang dilalukan oleh [37] deteksi serangan botnet denagn SVM menggunakn kernel liner hasil yang dapat untuk nilai akurasinya adalah sebesar 92,91%, sedangan penelitian yang dilakukan oleh [38] melakukan deteksi serangan dengan SVM pada WSN dan IoT dan hasil yang didapat dalm penelitian ini adalah 98.8% untuk akurasinya.Hasil pengujian dalam penelitian ini dengan membandingan tiga kernel SVM dengan nilai parameter C adalah c=1.0, sebelum melakukan proses klasifikasi dilakukan prepocessing untuk menghasilkan data yang baik. Adapun proses yang dilakukan menghapus atribut nomor, IP sumber dan tujuan hal ini dilakukan krn ip type data adalah string. Selanjutnya missing values untuk mendeteksi data yang kosong dan menghasilkan tanpa ada data yang kosong. Tahap selanjtunya adalah proses Outliers yang digunakan penyimpangan dari data yang akan digunakan, karena semua sifat varibel konitu maka hampir semua varibel mempunyai outliers-nya. Pada penelitian ini hasil outliernya hanya mendeteksi dan untuk proses lebih lanjut akan digunakan pada penelitian selanjutnya. Setelah proses prepossing selesai dilanjutkan dengan menguji algoritma SVM denan tiga kernel dan hasinya ditampilkan data dibawah ini

Tabel 5. Hasil pengujian tiga kernel SVM

Kernel	Nilai C	Akurasi
Linier	1.0	0.9998
RBF	1.0	0.9997
sigmoid kernel	1.0	0.9856

Dari hasil tabel 5 dapat dilihat untuk jenis data yang uji masih posisi terbaik adalah Liner, hasil ini disebabkan kernel liner data dipisahkan dengan garis lurus, sedangkan bagian kernel RBF mendapatkan nilai yang hal ini disebabkan data yang dimasukan tidak berbanding sama dalam atribut label.Dari hasil yang didapatkan nilai akurasinya dari tiga kernel didapatkan nilai akurasinya 0.998 untuk kernel liner artinya kernel ini mampu dengan baik untuk memisahkan kelas dengan baik, sedangan kernel Sigmod nilai akurasinya sebesar 0.9856 ini hasilnya baik dalam memisahkan kelas walaupun nilai lebih kecil dari liner. Sedangkan Kernel RBF (*Radial Basis Function*) memiliki akurasi sebesar 0.998. step selanjut dalam penelitian ini menghitung matrik klasifikasi dengan tujuan untuk mengevaluasi model klasifikasi dengan data yang dihasilksan sebagai berikut Classification accuracy : 0.9970, Classification error : 0.0030 Precision : 1.0000 dan Recall or Sensitivity : 0.9970

5 KESIMPULAN DAN SARAN

Dari dataset dari peracangan sistem IoT yang dijalankan menghasilkan 94.4% unutk data yang masuk dalam sistem penyimpanan dan 5.42% untuk paket yang hilang hal ini disebabkan karena belum sistem otomatis pada saat mulai dan berheti pada proses pengiriman data ke server. Dari hasil pengujian yang dilakukan dengan algortima SVM dengan membandingkan tiga kernel akurasi yang paling baik adalah kerne liner 0.998, kedua Sigmod nilai akurasinya sebesar 0.9856 dan terahir RBF 0.997. kedepannya penelitian akan fokus pada proses balancing data untuk mengahasilkan akurasi yang baik dan juga akan menggunakn metode yang lain sebagai pembanding

DAFTAR PUSTAKA

[1] L. Newcombe, P. Yang, C. Carter, and M. Hanneghan, "Internet of things enabled technologies for behaviour analytics in elderly person care: a survey," *Proc. - 2017 IEEE Int. Conf. Internet Things, IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-SmartData 2017*, vol. 2018-Janua, pp. 863–870, 2018, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.133.

[2] E. A. Shammar and A. T. Zahary, "The Internet of Things (IoT): a survey of techniques, operating systems, and trends," *Libr. Hi Tech*, vol. 38, no. 1, pp. 5–66, Jan. 2020, doi: 10.1108/LHT-12-2018-0200.

[3] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0268-2.

[4] E. Bou-Harb and N. Neshenko, "Cyber threat intelligence for the internet of things," *Cyber Threat Intell. Internet Things*, pp. 1–89, 2020, doi: 10.1007/978-3-030-45858-4.

[5] P. Anand, Y. Singh, A. Selwal, M. Alazab, S. Tanwar, and N. Kumar, "IoT vulnerability assessment for sustainable computing: Threats, current solutions, and open challenges," *IEEE Access*, vol. 8, pp. 168825–168853, 2020, doi: 10.1109/ACCESS.2020.3022842.

[6] H. Wang *et al.*, "An Evolutionary Study of IoT Malware," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15422–15440, 2021, doi: 10.1109/JIOT.2021.3063840.

[7] E. M. Karanja, S. Masupe, and J. Mandu, "Internet of Things Malware : A Survey," *Int. J. Comput. Sci. Eng. Surv.*, vol. 8, no. 3, pp. 1–20, 2017, doi: 10.5121/ijcses.2017.8301.

[8] Z. Liu *et al.*, "An Integrated Architecture for IoT Malware Analysis and Detection BT - IoT as a Service," 2019, pp. 127–137.

[9] H. Takase, R. Kobayashi, M. Kato, and R. Ohmura, "A prototype implementation and evaluation of the malware detection mechanism for IoT devices using the processor information," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 71–81, 2020, doi: 10.1007/s10207-019-00437-y.

[10] R. Boutaba *et al.*, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1. SpringerOpen, pp. 1–99, 2018. doi: 10.1186/S13174-018-0087-2.

[11] M. Shafiq, X. Yu, A. A. Laghari, L. Yao, N. K. Karn, and F. Abdessamia, "Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms," *2016 2nd IEEE Int. Conf. Comput. Commun. ICCC 2016 - Proc.*, pp. 2451–2455, 2017, doi: 10.1109/CompComm.2016.7925139.

[12] L. Kong, G. Huang, K. Wu, Q. Tang, and S. Ye, "Comparison of Internet Traffic Identification on Machine Learning Methods," in *2018 International Conference on Big Data and Artificial Intelligence (BDAI)*, Jun. 2018, pp. 38–41. doi: 10.1109/BDAI.2018.8546682.

[13] H. Hindy *et al.*, "A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 104650–104675, 2020, doi: 10.1109/ACCESS.2020.3000179.

[14] S. Kakati, D. Chouhan, A. Nag, and S. Panja, "Survey on Recent Malware Detection Techniques for IoT," *Lect. Notes Electr. Eng.*, vol. 888, no. September, pp. 647–659, 2022, doi: 10.1007/978-981-19-1520-8_53.

[15] A. Wolsey, "The State-of-the-Art in AI-Based Malware Detection Techniques: A Review," *arXiv Prepr. arXiv2210.11239*, pp. 1–18, 2022, [Online]. Available: https://arxiv.org/abs/2210.11239%0Ahttps://arxiv.org/pdf/2210.11239

[16] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," *Proc. 2017 Des. Autom. Test Eur. DATE 2017*, pp. 169–174, 2017, doi: 10.23919/DATE.2017.7926977.

[17] H. Hashemi, A. Azmoodeh, A. Hamzeh, and S. Hashemi, "Graph embedding as a new approach for unknown malware detection," *J. Comput. Virol. Hacking Tech.*, vol. 13, no. 3, pp. 153–166, 2017, doi: 10.1007/s11416-016-0278-y.

[18] D. Wang, X. Zhang, J. Ming, T. Chen, C. Wang, and W. Niu, "Resetting Your Password Is Vulnerable: A Security Study of Common SMS-Based Authentication in IoT Device," *Wirel. Commun. Mob. Comput.*, vol. 2018, 2018, doi: 10.1155/2018/7849065.

[19] A. Corum, D. Jenkins, and J. Zheng, "Robust PDF Malware Detection with Image Visualization and Processing Techniques," in *Proceedings - 2019 2nd International Conference on Data Intelligence and Security, ICDIS 2019*, 2019, pp. 108–114. doi: 10.1109/ICDIS.2019.00024.

[20] H. Naeem *et al.*, "Malware detection in industrial internet of things based on hybrid image visualization and deep learning model," *Ad Hoc Networks*, vol. 105, p. 102154, 2020, doi: 10.1016/j.adhoc.2020.102154.

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023

[21] M. Hirano and R. Kobayashi, “Machine Learning-based Ransomware Detection Using Low-level Memory Access Patterns Obtained From Live-forensic Hypervisor,” in *2022 IEEE International Conference on Cyber Security and Resilience (CSR)*, 2022, pp. 323–330. doi: 10.1109/CSR54599.2022.9850340.
[22] H. Naeem, B. M. Alshammari, and F. Ullah, “Explainable Artificial Intelligence-Based IoT Device Malware Detection Mechanism Using Image Visualization and Fine-Tuned CNN-Based Transfer Learning Model,” *Comput. Intell. Neurosci.*, vol. 2022, 2022, doi: 10.1155/2022/7671967.
[23] M. Asam *et al.*, “IoT malware detection architecture using a novel channel boosted and squeezed CNN,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–12, 2022, doi: 10.1038/s41598-022-18936-9.
[24] M. Nakahara, N. Okui, Y. Kobayashi, and Y. Miyake, “Malware Detection for IoT Devices using Automatically Generated White List and Isolation Forest,” *Int. Conf. Internet Things, Big Data Secur. IoTBDS - Proc.*, vol. 2021-April, no. IoTBDS, pp. 38–47, 2021, doi: 10.5220/0010394900380047.
[25] C. Y. Wu, T. Ban, S. M. Cheng, B. Sun, and T. Takahashi, “IoT Malware Detection Using Function-Call-Graph Embedding,” *2021 18th Int. Conf. Privacy, Secur. Trust. PST 2021*, 2021, doi: 10.1109/PST52912.2021.9647806.
[26] J. Jeon, J. H. Park, and Y. S. Jeong, “Dynamic Analysis for IoT Malware Detection with Convolution Neural Network Model,” *IEEE Access*, vol. 8, pp. 96899–96911, 2020, doi: 10.1109/ACCESS.2020.2995887.
[27] F. Sharmin *et al.*, “Humidity Based Automated Room Temperature Controller Using IoT,” in *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 226–231. doi: 10.1109/I-SMAC47947.2019.9032624.
[28] A. Indu and S. M. Kumar, “An Approach for Implementing Innovative Weather Monitoring System with DHT11 Sensor and Arduino Uno Tool based on IoT,” in *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2022, pp. 274–278. doi: 10.1109/I-SMAC55078.2022.9987289.
[29] H. Patel, “Network Traffic Analysis Using Packet Sniffer Pallavi Asrodia *, Hemlata Patel **,” vol. 2, no. 3, pp. 854–856, 2012.
[30] R. Spangler, “Packet sniffer detection with antisniff,” *... Wisconsin, Dep. Comput. ...*, no. May, 2003.
[31] O. n Henry and M. A. Agana, “Intranet Security Using A LAN Packet Sniffer to Monitor Traffic,” pp. 57–68, 2019, doi: 10.5121/csit.2019.90806.
[32] A. Siswanto, A. Syukur, E. A. Kadir, and Suratin, “Network Traffic Monitoring and Analysis Using Packet Sniffer,” in *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2019, pp. 1–4. doi: 10.1109/COMMNET.2019.8742369.
[33] R. Alshammari and A. N. Zincir-Heywood, “Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?,” *Comput. Networks*, vol. 55, no. 6, pp. 1326–1350, 2011, doi: 10.1016/j.comnet.2010.12.002.
[34] A. R. Vasudevan, E. Harshini, and S. Selvakumar, “SSENet-2011: A Network Intrusion Detection System dataset and its comparison with KDD CUP 99 dataset,” *Asian Himalayas Int. Conf. Internet*, 2011, doi: 10.1109/AHICI.2011.6113948.
[35] M. Y. Su, “Real-time anomaly detection systems for Denial-of-Service attacks by weighted k-nearest-neighbor classifiers,” *Expert Syst. Appl.*, vol. 38, no. 4, pp. 3492–3498, 2011, doi: 10.1016/j.eswa.2010.08.137.
[36] S. Hao, J. Hu, S. Liu, T. Song, J. Guo, and S. Liu, “Improved SVM method for internet traffic classification based on feature weight learning,” *ICCAIS 2015 - 4th Int. Conf. Control. Autom. Inf. Sci.*, pp. 102–106, 2015, doi: 10.1109/ICCAIS.2015.7338641.
[37] R. G. Azhari, V. Suryani, R. R. Pahlevi, and A. A. Wardana, “The Detection of Mirai Botnet Attack on the Internet of Things (IoT) Device Using Support Vector Machine (SVM) Model,” in *2022 10th International Conference on Information and Communication Technology (ICoICT)*, 2022, pp. 397–401. doi: 10.1109/ICoICT55009.2022.9914830.
[38] S. Mishra, “Detection and mitigation of attacks in SDN-based IoT network using SVM,” *Int. J. Comput. Appl. Technol.*, vol. 65, no. 3, pp. 270–281, Jan. 2021, doi: 10.1504/IJCAT.2021.116009.

Submitted: 08 September 2023; Reviewed: 29 September 2023; Accepted: 11 Okober 2023; Published: 31 Oktober 2023