Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

Studi Perbandingan Deteksi Intrusi Jaringan Menggunakan Machine Learning: (Metode SVM dan ANN)

Comparative Study of Network Intrusion Detection Using Machine Learning: (SVM and ANN Method)

Tony Tan[1], **Hendi Sama**[2], **Gautama Wijaya**[3], **Osei Enoch Aboagye**[4]
Program Studi Sistem Informasi, Universitas Internasional Batam, Indonesia[1234]
tony@uib.ac.id[1], hendi@uib.ac.id[2], gautama.wijaya@uib.ac.id[3], 2031172.osei@uib.edu*[4]

Abstrak

Machine Learning berkaitan dengan penggunaan algoritma untuk membuat mesin berfungsi. Algoritma supervised machine learning belajar pada dataset untuk membuat prediksi berdasarkan pengetahuan yang mereka peroleh saat belajar. Machine Learning memiliki dampak signifikan dalam keamanan siber. Sistem deteksi intrusi (IDS), sistem pencegahan intrusi (IPS) dan firewall tradisional membantu mendeteksi intrusi tetapi sayangnya, kebanyakan dari mereka memberi false alarm, dapat memiliki kerentanan dan dapat salah konfigurasi. Penggunaan algoritma machine learning telah terbukti lebih efektif dalam deteksi intrusi. Penelitian ini bertujuan untuk membandingkan efektivitas Algoritma Support Vector Machine (SVM) dan Artificial Neural Network (ANN) untuk intrusi deteksi. Penelitian ini menggunakan metode eksperimen dengan melatih dan menguji SVM dan ANN pada Dataset KDD Cup 99 di Google Colaboratory. Skor akurasi pelatihan dan pengujian, waktu pelatihan dan pengujian, Receiver Operating Characteristic Curve (Kurva ROC) dan kecepatan jaringan adalah parameter untuk perbandingan. Hasil dari eksperimen menunjukkan bahwa; Kedua model bagus untuk mendeteksi intrusi karena SVM dan ANN memiliki skor di atas 90%. SVM lebih efektif daripada ANN dalam deteksi intrusi dengan akurasi pelatihan dan pengujian 99,87% dan 99,81%. Juga AUC untuk SVM adalah 1 untuk semua lima kelas dan mengambil lebih sedikit waktu dalam melatih dataset.
Kata kunci: Machine Learning; Jaringan; Deteksi Intrusi; SVM; ANN.

Abstract

Machine Learning deals with the use of algorithms to make machines function. Supervised Machine learning algorithms learn on datasets to make predictions based on the knowledge they obtain while learning. Machine Learning have a significant impact on cybersecurity. Intrusion detection systems (IDS), Intrusion prevention systems (IPS) and traditional firewall help to detect intrusion but unfortunately, most of them suffer false alarms, could have vulnerabilities, and could be misconfigured. The use of machine learning models has proved to be more effective in intrusion detection. This study aims to compare the effectiveness of Support Vector Machine (SVM) and Artificial Neural Network (ANN) in detecting intrusions. The study uses an experimental method by training and testing SVM and ANN on the KDD Cup 99 Dataset in the Google Colaboratory. Train and test accuracy scores, train and test times, Receiver operating characteristic curve (ROC Curve), and internet speed were the parameters for the comparison. Results from the experiments show that; Both models are good for intrusion detection since SVM and ANN had scores above 90%. SVM is more effective than ANN in intrusion detection with a training and testing accuracy of 99.87% and 99.81%. Also AUC for SVM is 1 for all the five classes and took less time in training the dataset.
Keywords: Machine Learning; Network; Intrusion Detection; SVM; ANN.

Naskah diterima 16 Juli 2023; direvisi 16 Agustus 2023; dipublikasi 1 September 2023.

1. Pendahuluan

Machine learning berkaitan dengan perangkat-perangkat, data dan algoritma [1]. Dalam *machine learning*, data disediakan untuk algoritma untuk dipelajari dan diklasifikasikan [2]. Ada dua jenis *machine learning* yaitu; *supervised machine learning* dan *unsupervised machine learning* [15]. Seperti namanya, *supervised machine learning* berfungsi di bawah pengawasan. *supervised machine learning* harus dilatih untuk dapat memprediksi antar objek. *Unsupervised machine learning* tidak memerlukan pengawasan. Artinya, *unsupervised machine learning* dapat memprediksi data yang tidak berlabel tanpa melalui bentuk pelatihan apa pun.

Memastikan keamanan jaringan, data, dan sumber daya lainnya lebih menantang. Penerapan algoritma *machine learning* mengurangi kesulitan memeriksa *traffic* jaringan. Menerapkan *machine learning* dapat mengingatkan pengguna komputer tentang situs web berbahaya sebelum masuk ke masalah serius [3]. Dalam komputasi awan, *machine learning* memeriksa login pengguna yang tidak normal ke dalam aplikasi. *Machine*

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

learning mempelajari pola *traffic* jaringan untuk membedakan antara aktivitas normal dan tidak normal. Tanpa dekripsi, *machine learning* mampu memprediksi anomali dalam paket terenkripsi dalam jaringan. *Search engine* seperti *Chrome* menggunakan mekanisme *machine learning* untuk menjelajahi jaringan dan menyarankan situs web yang aman untuk pengguna. Dengan algoritma, *Chrome* dapat memisahkan situs web asli dari situs web berbahaya. Ini telah membantu melindungi pengguna komputer. Tanpa *machine learning*, keamanan komputer akan kehilangan lebih banyak fitur yang digunakan untuk perlindungan data.

Dalam keamanan siber, *machine learning* telah membantu dalam banyak hal. *Machine learning* telah membantu memungkinkan deteksi dan penyesuaian ancaman otomatis. Sangat sulit untuk menganalisis volume data yang besar. Mengelola data dalam *data mining* digabungkan dengan penggunaan machine learning. Sebagian besar langkah-langkah keamanan di perangkat menuntut *machine learning*. Pengenalan wajah menggunakan algoritma *machine learning* untuk menganalisis miliaran piksel dalam gambar. Sekali lagi, pengenalan suara, dan sidik jari semuanya keamanan langkah-langkah yang menggunakan *machine learning*. Bahkan sulit bagi manusia untuk membedakan antara beberapa suara dan wajah, serta bagi *mesin learning* untuk dapat mengklasifikasikan atau mencocokan detail data kompleks tersebut memberi tahu seberapa kuat ia bekerja.

Deteksi intrusi adalah salah satu aspek penting dari keamanan siber. Deteksi intrusi sangat sulit karena mengganggu menuntut keterampilan dan teknik tingkat tinggi. Tanda intrusi berbeda dalam banyak hal. Tanda tangan atau cetakan intrusi dalam sejumlah besar *traffic* jaringan dapat disembunyikan sepenuhnya dari administrator server [4]. *Traffic* jaringan terakumulasi untuk membentuk sejumlah besar data dan ini menimbulkan masalah bagi administrator.

[5] Mengunakan *machine learning* untuk deteksi intrusi. SVM dan ANN digunakan sering sekali [6]. Beberapa serangan dan teknik intrusi yang ditemukan dalam dataset yang digunakan entah bagaimana serupa. Meskipun dataset yang digunakan dalam penelitian sebelumnya seperti [7], [8], [9] berbeda namun semuanya cocok untuk proyek deteksi intrusi. Sebagian besar karya sebelumnya menyebutkan KDD Dataset. Dapat dilihat bahwa sebagian besar karya sebelumnya berbicara tentang serangan DOS [10], [11], [12]. Hal ini dikarenakan DOS sangat marak di internet. Hasil penelitian [13], [14], [15] menunjukkan bahwa pendekatan machine learning sangat efektif. Penerapan machine learning terbukti lebih berhasil.

Deteksi intrusi menggunakan pendekatan machine learning sangat efektif [16]. Algoritma *machine learning* menganalisis data besar, mengklasifikasikan, mencocokkan, mendeteksi, dan melaporkan jejak penyusupan. Algoritma *supervised machine learning* belajar pada *datataset* dan fitur-fiturnya [17]. Algoritma mampu mendeteksi apa yang telah mereka pelajari dan bahkan fitur yang baru. Algoritma *machine learning* yang diterapkan dalam deteksi intrusi selama bertahun-tahun meliputi; *Support Vector Machine* (SVM), *Artificial Neural Network* (ANN), *Decision Tree*, *Random Forest* [10] dan *Logistic Regression*. Algoritma telah diakui mampu mendeteksi intrusi secara akurat dengan jumlah *false alarm* yang lebih sedikit.

SVM dan ANN dipilih untuk penelitian ini karena keduanya adalah algoritma *supervised machine learning* yang mengklasifikasikan dataset dengan akurasi tinggi [18]. Salah satu aspek *machine learning* yang menunjukkan seberapa baik algoritma memprediksi adalah *True Positive Rate* (TPR) mereka. TPR yang tinggi menunjukkan bahwa algoritma melakukan prediksi yang benar. Penelitian sebelumnya telah mengidentifikasi SVM dan ANN sebagai algoritma yang memiliki akurasi dan TPR yang lebih tinggi [19]. SVM dan ANN sering digunakan dalam banyak penelitian untuk deteksi intrusi. Ini menunjukkan seberapa efektif mereka dalam mengerjakan tugas-tugas kompleks.

Machine learning telah diakui efektif dalam deteksi intrusi tetapi karena algoritma *machine learning* banyak, orang akan bingung tentang mana yang lebih efektif. Penelitian ini membandingkan SVM dengan ANN untuk mengetahui mana yang lebih efektif dan harus diimplementasikan dalam deteksi intrusi. Hasil dari proyek akan memberikan gambaran tentang bagaimana kedua algoritma memprediksi intrusi. Menyatakan seberapa efektif salah satu dari dua algoritma tersebut, akan menjadi panduan bagi para peneliti dan administrasi jaringan di berbagai perusahaan dalam pilihan algoritma mereka untuk deteksi intrusi. Proyek ini bertujuan untuk menyarankan algoritma untuk deteksi intrusi.

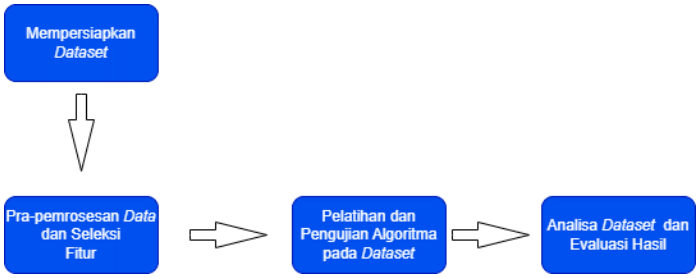
Ada harapan bahwa SVM dan ANN akan mendeteksi intrusi pada *dataset* dan memberikan hasil yang diinginkan. Memprediksi dengan akurasi yang diinginkan akan berkontribusi hingga keamanan siber. Ini akan memungkinkan administrator jaringan untuk mendeteksi tindakan penyerang yang tinggi dan kompleks. Jika serangan dalam dataset terdeteksi secara akurat setelah proyek penelitian, dapat diimplementasikan dalam situasi kehidupan nyata untuk meningkatkan keamanan komputer.

2. Metode Penelitian

Untuk mencapai tujuan penelitian ini [13], *dataset* yang sangat diakui dipilih untuk algoritma untuk melatih dan memprediksi. Penulis ikuti kerangka untuk keberhasilan proyek. Kerangka menguraikan semua

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

langkah yang diambil selama proses implementasi. Kerangka yang digunakan untuk penelitian deteksi intrusi ini dapat dilihat pada Gambar 1.



Gambar 1. Kerangka Deteksi Intrusi

2.1 Menyiapkan *Dataset*

Dalam proyek ini, *Dataset* KDD CUP 99 dianalisa menggunakan algoritma SVM dan ANN untuk mendeteksi intrusi. *Dataset* ini dikumpulkan di Laboratorium *MIT Lincoln*, yang dibawah *Defense Advanced Research Project Agency* (DARPA) [8], [20]. *Dataset* KDD CUP 99 dikelompokkan menjadi; data pelatihan dan, dan data pengujian [4]. Data pelatihan memiliki data yang dikumpulkan dalam tiga minggu. Minggu pertama dan ketiga tidak mengandung serangan. Minggu kedua memiliki serangan dari *dataset* 1998 dan banyak serangan baru yang berbeda. Data pengujian memiliki dua minggu serangan yang berbeda. Tahap ini memastikan ketersediaan *dataset*. *Dataset* diunduh dari situs web Kaggle. *Dataset* telah dapat diakses semua orang pada tautan https://www.kaggle.com/code/abhaymudgal/intrusion-detection-system/input. Data unduhan disimpan di *Google Drive* untuk proyek tersebut. Penulis membaca dataset untuk mengetahui apa isinya di *Google Colaboratory*. Membiasakan diri dengan semua komposisi atau fitur *dataset* sangat penting.

Dataset KDD Cup 99 memiliki 42 fitur (kolom) [21]. Dari 42, 41 fitur adalah fitur independen dan yang terakhir, 'normal' depenpen. Ini untuk memastikan hasil yang akurat [22]. Hanya 32 fitur yang digunakan untuk deteksi intrusi. Algoritma mengklasifikasikan serangan dari data normal (deteksi). Adapun total 42 fitur [1], [22], [25] pada *dataset* ditampilkan pada Tabel 1.

Tabel 1. Fitur *Dataset* KDD Cup 99

No	Fitur	Deskripsi
1	Durasi	Berapa lama koneksi
2	*Protocol_type*	Jenis protokol jaringan
3	*Service*	Jenis *service* pada destinasi
4	*Flag*	Status koneksi normal atau *error*
5	*Src_byte*	Total byte dari sumber
6	*Dst_byte*	Total byte dari destinasi
7	*Land*	1 jika koneksi dari/ke destinasi yang sama; 0 jika sebaliknya
8	*Wrong_fragment*	Total *fragment* yang salah
9	*Urgent*	Total *traffic* yang *urgent*
10	*Hot*	Total indikator yang *hot*
11	*Num_failed_logins*	Total login yang gagal
12	*Logged_in*	1 jika login sukses; 0 jika gagal
13	*Num_compromised*	Total kondisi yang dikompromi
14	*Root_shell*	1 jika *root shell* diperoleh; 0 jika tidak
15	*Su_attempted*	1 jika penyerang coba perintah su root; 0 jika tidak
16	*Num_root*	Total *root* yang diakses
17	*Num_file_creations*	Total *file* yang dibuat
18	*Num_shells*	Total perintah shell
19	*Num_access_files*	Total kegiatan pada *file*
20	*Num_outbound_cmds*	Total perintah *outbound*
21	*Is_host_login*	1 jika login tamu; 0 jika tidak
22	*Is_guest_login*	1 jika login tamu; 0 jika tidak
23	*Count*	Total koneksi ke tamu yang sama
24	*Srv_count*	Total koneksi ke *service* yang sama
25	*Serror_rate*	Persentase koneksi dengan *error* SYN

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

26	*Srv_serror_rate*	Persentase koneksi dengan *error* SYN
27	*Rerror_rate*	Persentase koneksi dengan *error* REJ
28	*Srv_rerror_rate*	Persentase koneksi dengan *error* REJ
29	*Same_srv_rate*	Persentase koneksi ke *service* yang sama
30	*Diff_srv_rate*	Persentase koneksi ke *service* yang beda
31	*Srv_diff_host_rate*	Persentase koneksi tamu yang beda
32	*Dst_host_count*	Persentase koneksi ke tamu yang sama
33	*Dst_host_srv_count*	Total koneksi ke *service* yang sama
34	*Dst_host_same_srv_rate*	Persentase koneksi ke *service* yang sama
35	*Dst_host_diff_srv_rate*	Persentase koneksi ke *service* yang beda dari tamu yang sama
36	*Dst_host_same_src_port_rate*	Persentase koneksi dari sumber port yang sama
37	*Dst_host_srv_diff_host_rate*	Persentase koneksi dari *service* yang sama ke tamu yang beda
38	*Dst_host_serror_rate*	Koneksi dengan *error* SYN
39	*Dst_host_srv_serror_rate*	Presentase koneksi dengan *error* SYN
40	*Dst_host_rerror_rate*	Presentase koneksi dengan *error* REJ
41	*Dst_host_srv_rerror_rate*	Presentase koneksi dengan *error* SYN
42	*Label*	1 atau 0; ya atau tidak

Ada total 22 serangan dalam *dataset* pelatihan. 22 serangan tersebut dikategorikan ke dalam empat (4) kelompok serangan utama: *Denial of Service* (DOS), *Remote to User* (R2L), *User to Root* (U2R) [2], [7] dan *Probing* [23]. Data pengujian berisi serangkaian serangan yang berbeda untuk memastikan algoritma dapat mendeteksi serangan baru. Sangat penting untuk mempertimbangkan era di mana dataset diambil - akhir 90-an. Serangan DOS sangat umum saat itu [24]. DOS memiliki hitungan 391458 yang bahkan lebih banyak dari koneksi normal. U2R tidak begitu umum. U2R tercatat hanya 52 koneksi. Tabel 2 menunjukkan jenis serangan dan kategori utama pada *Dataset* KDD Cup 99.

Tabel 2. Tipe Serangan pada Dataset KDD Cup 99

DOS	R2L	U2R	Probe
Back, Land, Neptune, Pod, Smurf, Teardrop	Ftp_write, Guess_passwd, Imap, Multihop, Phf, Spy, Warezclient, Warezmaster	Buffer_overflow, Loadmodule, Perl, Rootkit,	Ipsweep, Nmap, Portsweep, Satan,

Transmission Control Protocol (TCP), *User Datagram Protocol* (UDP) dan *Internet Control Message Protocol* (ICMP) [9] adalah tiga protokol yang ditargetkan untuk serangan tersebut [20]. Fungsi semua protokol mempengaruhi jumlah serangan yang mereka dapatkan. ICMP adalah yang paling ditargetkan karena fakta bahwa ia menangani manajemen dan pemecahan masalah. Penyerang ingin memiliki kontrol manajerial atas sistem. Protokol jaringan dan jumlah total koneksi yang sesuai pada *dataset* ditunjukkan pada Tabel 3.

Tabel 3. Protokol Jaringan pada *Dataset* KDD Cup 99

No	Protokol	Total Koneksi
1	ICMP	283602
2	TCP	190065
3	UDP	20354

Alarm dalam *Dataset* KDD Cup 99 ada pada Tabel 4. Dapat dilihat bahwa koneksi dan akhiran Normal (SF) adalah yang paling banyak. Koneksi dicoba, dilihat dan tidak ada respons (SO) adalah yang terbanyak kedua dan koneksi ditolak (REJ) adalah yang ketiga. Sisa bendera memiliki catatan lebih sedikit. Adapun 11 alarm yang berbeda pada *Dataset* KDD Cup 99 ditampilkan pada Tabel 4.

Tabel 4. Alarm *Dataset* KDD Cup 99

No	Alarm	Total Jumlah Alarm
1	SF	378440
2	S0	87007
3	REJ	26875

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

4	RSTR	903
5	RSTO	579
6	SH	107
7	S1	57
8	S2	24
9	RSTOS0	11
10	S3	10
11	OTH	8

2.2 Pra-pemrosesan Data dan Seleksi Fitur

Pada tahap ini, semua nilai *null dataset* dihapus [25]. Untuk memeriksa intrusi, fitur yang dianggap sebagai *noise* dihapus dari *dataset* (seleksi fitur). Menghapus nilai *null* memastikan hasil yang akurat [26]. Nilai *null* akan membuat kebisingan dalam pelatihan dan pengujian jika tidak dihapus. Tidak semua fitur *dataset* berguna. Beberapa fitur yang tidak diperlukan untuk memberikan yang diinginkan hasil tidak dipilih untuk penelitian. Tahap ini memilih semua fitur yang diperlukan untuk eksperimen.

2.3 Pelatihan dan Pengujian Algoritma pada *Dataset*

Proses dimulai dengan memasang di *MyDrive* untuk mengakses *Dataset* KDD Cup 99. *Dataset* kemudian dianalisis dengan SVM dan ANN untuk menemukan serangan. Algoritma mengklasifikasikan data yang normal dari serangan dengan jumlah yang sesuai.

Pada tahap ini, Algoritma dilatih pada data pelatihan yang memiliki serangkaian serangan intrusi. Penting untuk memahami apa arti pelatihan. Pelatihan adalah ketika algoritma belajar [27].

Pengujian dilakukan dengan menggunakan *dataset* pengujian. Pengujian adalah ketika algoritma memprediksi setelah belajar. Inilah sebabnya mengapa data pelatihan berbeda dari data pengujian. Serangan baru di data pengujian disajikan ke algoritma untuk dideteksi. Mampu memprediksi serangan baru dalam data uji berarti algoritma siap mendeteksi serangan baru dalam jaringan.

Setelah pelatihan dan pengujian, Kurva ROC diplot untuk algoritma untuk mengevaluasi kinerja mereka dan membuat kesimpulan akhir.

2.4 Evaluasi Hasil

Evaluasi kinerja algoritma dalam deteksi intrusi dilakukan berdasarkan beberapa parameter yaitu hasil dari pelatihan, pengujian, kekuatan jaringan dan Kurva ROC. Secara detail, parameter untuk mengukur kinerja algoritma adalah; akurasi pelatihan, akurasi pengujian, waktu pelatihan, waktu pengujian, kecepatan jaringan dan *Area Under the ROC Curve* (AUC). Jika suatu algoritma mendapatkan akurasi 90% ke atas, maka dinyatakan baik untuk deteksi intrusi.

Kurva ROC, adalah *plot* grafis yang menggambarkan kemampuan diagnostik sistem pengklasifikasi biner karena ambang diskriminasinya bervariasi [6]. AUC menunjukkan seberapa baik dan akurat algoritma mengklasifikasikan kelas [28]. AUC dilakukan dengan metode *multiclass* menggunakan ovr (*one-versus-rest*). Status probalitas diatur ke *true* untuk algoritma karena probabilitas *false* secara *default*. Ada lima (5) kelas dan AUC selesai mengambil satu kelas melawan empat lainnya. Hasil akhir untuk proyek ditentukan pada Kurva ROC. Hasil dari algoritma diukur pada Kurva ROC. Di antara SVM dan ANN, yang memiliki hasil tertinggi pada Kurva ROC disimpulkan paling efektif untuk mendeteksi intrusi. Parameter untuk mengukur kinerja pada Kurva ROC adalah; *True Positive Rate* (TPR) dan *False Positive Rate* (FPR) [4]. TPR adalah jumlah *True Positive* (TP) dan FPR adalah jumlah *False Positive* (FP). Nilai AUC [26] yang tinggi menunjukkan kinerja yang baik. AUC 0,70-0,80 (dapat diterima), AUC 0,80-0,90 (sangat baik) dan AUC 0,90 ke atas (luar biasa). TPR dan FPR dapat dihitung menggunakan persamaan 1 dan 2 yang berbunyi [4];

$$\text{TPR} = \frac{\text{TP}}{\text{TP+FN}} \quad (1)$$

dan

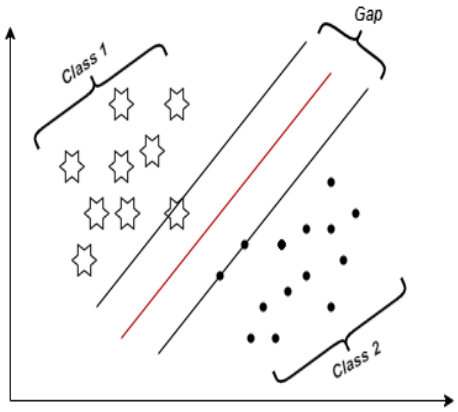
$$\text{TPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (2)$$

2.5 *Support Vector Machine* (SVM)

Algoritma SVM adalah algoritma *supervised machine learning* [29]. SVM sangat efektif dalam menganalisis data untuk klasifikasi. SVM menilai data secara intensif untuk mengklasifikasikannya menjadi

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

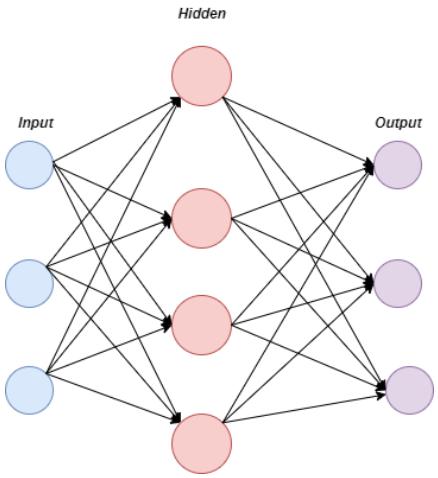
data normal dan data berbahaya. Karena kemampuannya untuk mengklasifikasikan, telah digunakan selama bertahun-tahun dalam deteksi intrusi. Algoritma SVM yang dikembangkan oleh Vlandimir Vapnik dipercaya oleh banyak administrator keamanan. SVM mengklasifikasikan data tidak linier. Arsitektur Algoritma SVM yang menunjukkan bagaimana SVM mengklasifikasikan data telah ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Algoritma SVM

2.6 Artificial Neural Network (ANN)

Algoritma ANN juga merupakan algoritma *supervised machine learning* [30]. ANN datang sebelum SVM dan telah diimplementasikan dalam keamanan siber untuk berbagai tujuan. ANN telah digunakan untuk mendeteksi intrusi selama beberapa tahun. Karena efektivitasnya dalam deteksi intrusi, ANN digolongkan sebagai salah satu algoritma yang paling banyak diterapkan. ANN memiliki lapisan dalam arsitekturnya. Lapisannya adalah input, hidden dan output. Lapisan input bereaksi terhadap data input. Lapisan hidden mengatur data sehingga dapat dipahami oleh lapisan output. Lapisan output bereaksi terhadap informasi tentang bagaimana algoritma mempelajari tugas. Dalam arsitektur, setiap neuron di lapisan hidden terhubung ke setiap neuron di lapisan input dan output [31]. ANN mengklasifikasikan data linier. Arsitektur Algoritma ANN ditunjukkan pada Gambar 3.



Gamber 3. Arsitektur Algoritma ANN

3.7 Perangkat Lunak yang Digunakan

Google Colaboratory digunakan untuk penelitian ini. Seperti namanya, *Google Colaboratory* adalah laboratorium *online* yang dapat diakses oleh semua orang yang menjalankan program *Python* yang dapat dieksekusi. Aplikasi online ini digunakan bersama dengan *Google drive* yang berisi dataset untuk eksperimen.

3. Hasil dan Pembahasan

Sesi ini berbicara tentang hasil dari eksperimen. Algoritma dibandingkan menggunakan akurasi pelatihan, akurasi pengujian [32], waktu pelatihan, waktu pengujian, AUC dan kecepatan jarigan. Algoritma dilatih dan

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

diuji pada *Dataset* KDD CUP 99 untuk menghitung parameter. Eksperimen menunjukkan hasil menarik yang dapat digunakan untuk membuat kesimpulan yang akurat. Setiap algoritma dalam eksperimen ini diberi sumber daya yang sama dan adil untuk melatih dan menguji pada *dataset*. *Intel® Core*(TM) i5-7300U CPU @ 2.60GHz, 2712 Mhz, 2 *Core*(*s*), 4 *Logical Processor*(s), 8GB RAM, 238.5 GB [5] penyimpanan adalah spesifikasi yang digunakan.

Ditemukan bahwa, SVM memiliki akurasi pelatihan dan akurasi pengujian 99,87% dan 99,81% masing-masing. Di sisi lain, ANN memiliki akurasi pelatihan dan akurasi pengujian 99,86% dan 99,85% masing-masing. Hasil ini menunjukkan SVM dan ANN baik untuk proyek deteksi intrusi karena baik SVM dan ANN memiliki akurasi di atas 90%. SVM berkinerja lebih baik dari ANN dalam akurasi pelatihan dengan *margin* 0,01% [33]. Juga, ANN berkinerja lebih baik dari pada SVM dalam akurasi pengujian dengan *margin* kecil 0,04%. *Margin* menunjukkan bahwa SVM belajar lebih akurat dari pada ANN. Di sisi lain, ANN terdeteksi dengan baik setelah belajar daripada SVM. Juga margin menunjukkan bahwa SVM relatif efektif daripada ANN. SVM memiliki kemampuan untuk mengatasi masalah dimensi tinggi [14]. Sementara itu, dalam penelitian [19], [34] menunjukkan bahwa ANN lebih efektif daripada SVM dalam deteksi intrusi. Disebutkan di metode bahwa data pengujian memiliki serangan di data pelatihan ditambah banyak serangan yang berbeda. Ini untuk memastikan bahwa algoritma dapat mendeteksi serangan baru. Karen keduanya memiliki skor akurasi pengujian di atas 90% berarti mereka siap mendeteksi serangan baru secara akurat. Tabel 5 menunjukan hasil akurasi pelatihan dan pengujian SVM dan ANN.

Tabel 5. Akurasi Pelatihan dan Pengujian

Algoritma	Akurasi Pelatihan	Akurasi Pengujian
SVM	99.87%	99.81%
ANN	99.86%	99.85%

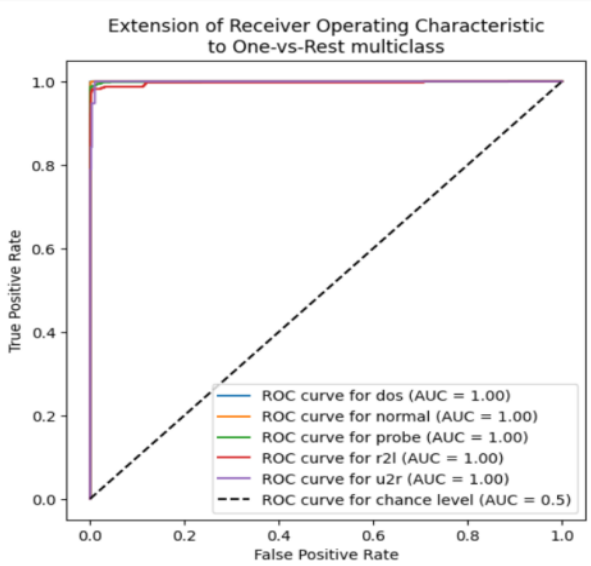
Sehubungan dengan waktu, SVM mengambil 120 detik (2 menit) dan 60 detik (1 menit) untuk melatih dan menguji *dataset* masing-masing. Sementara itu, ANN sesuai dengan waktu latihan dan pengujian 660 detik (11 menit) dan 10s. Di sini dapat melihat bahwa SVM membutuhkan lebih sedikit waktu untuk berlatih. ANN mengambil tambahan 9 menit untuk berlatih dibandingkan dengan SVM. 9 menit adalah perbedaan besar. ANN mengalahkan SVM dalam waktu pengujian dengan 50 detik waktu yang tidak terlalu signifikan. Waktu latihan dan pengujian sesuai dengan akurasi pelatihan dan pengujian. Waktu pelatihan adalah waktu yang digunakan untuk memisahkan data, memproses data, dan mengevaluasi data oleh algoritma. SVM membutuhkan waktu 2 menit untuk mempelajari *dataset* dengan akurasi 99,87%. Ini adalah kinerja yang luar biasa. Untuk menguji, SVM membutuhkan waktu 1 menit. Sementara itu, ANN menggunakan 11 menit untuk belajar pada *dataset* untuk akurasi 99,86%. Setelah belajar, ANN hanya menggunakan 10 detik waktu untuk memprediksi. [35] Mengatakan dalam penelitian mereka bahwa algoritma yang efektif membutuhkan waktu lebih sedikit untuk bekerja pada dataset. Tabel 6 menunjukan hasil waktu pelatihan dan pengujian SVM dan ANN.

Tabel 6. Waktu Pelatihan dan Pengujian

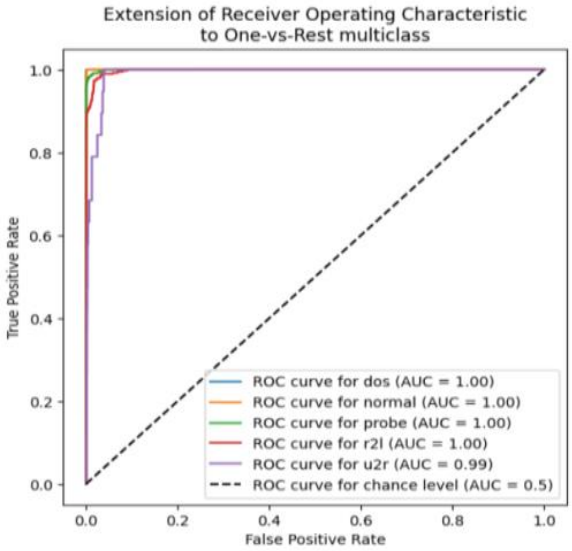
Algoritma	Waktu Pelatihan	Waktu Pengujian
SVM	2 menit	1 menit
ANN	11 menit	10 detik

AUC menunjukkan bahwa SVM memiliki kinerja yang lebih besar daripada ANN. Pada Kurva ROC, AUC untuk DOS adalah 1,00, AUC untuk normal adalah 1,00, AUC untuk *Probe* adalah 1,00, AUC untuk R2L adalah 1,00 dan AUC untuk U2R juga 1,00 dalam kasus SVM. Untuk ANN, AUC untuk DOS adalah 1,00, AUC untuk normal adalah 1,00, AUC untuk *Probe* adalah 1,00, AUC untuk R2L adalah 1,00 dan AUC untuk U2R adalah 0,99. Dapat disimpulkan bahwa SVM melakukan klasifikasi yang baik dibandingkan ANN dalam penelitian ini karena SVM dapat AUC 1.00 untuk semua kelas tapi ANN tidak. Karena jika model mendapat AUC sekitar 0,90 menunjukkan kinerja yang luar biasa [21], baik SVM dan ANN memiliki prediksi yang sangat baik. Hasilnya menunjukkan bahwa keduanya bagus. *Dataset* juga memiliki pengaruh besar pada kinerja mereka. *Dataset* KDD CUP 99 telah diakui sebagai *dataset* dengan kualitas yang membantu dalam deteksi intrusi yang akurat. Ini adalah salah satu dataset yang paling banyak digunakan dalam deteksi intrusi [36]. Meskipun teknik dalam intrusi terus berubah namun teknik dalam *dataset* masih digunakan sampai sekarang. Hasil AUC untuk SVM dan ANN dapat dilihat pada Gambar 4 dan Gamber 5 masing-masing.

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2



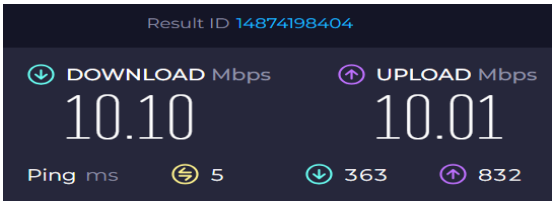
Gambar 4. AUC ROC SVM



Gambar 5. AUC ROC ANN

Karena algoritma dilatih dan diuji secara online menggunakan *Google Colaboratory*, kualitas jaringan memiliki pengaruh pada hasil. Pelatihan dan Pengujian membutuhkan jaringan yang cepat dan kuat. Penelitian ini dilakukan dengan menggunakan *Wireless Fidelity (WI-FI)* kampus dengan pita jaringan 5 GHz. Jarak dari laptop ke adaptor WI-FI adalah 2 meter. Adaptor diposisikan di koridor dan laptop ditempatkan di ruangan yang berdekatan. 5GHz tidak pergi jauh dan tidak memiliki kekuatan tembus dinding dibandingkan dengan 2,4 GHz, namun karena jaraknya pendek, kecepatannya masih utuh. Tes kecepatan diperiksa selama pelatihan dan pengujian semua algoritma (baik SVM dan ANN). Kecepatan unduh 10,10 Mbps dan kecepatan unggah 10,1 Mbps direkam untuk SVM dan ANN. Karena kecepatannya sama untuk kedua agoritma, perbandingan bergantung pada parameter lain. Hasil dari tes kecepatan jaringan WI-FI yang digunnakan telah ditunjukkan pada Gambar 6.

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2



Gambar 6. Kecepatan Jaringan

Selama bertahun-tahun, perusahaan telah menggunakan banyak mekanisme untuk keamanan. Upaya perusahaan untuk mengurangi intrusi dan melindungi data meliputi:

Enkripsi. Enkripsi data adalah proses pengkodean data. Dua *status* di mana data dapat dienkripsi adalah enkripsi transit dan enkripsi saat istirahat. Enkripsi transit mengenkripsi data yang bergerak (data dikirim dari satu tempat ke tempat lain) [21] dan enkripsi saat istirahat mengenkripsi data pada drive. Smartphone meminta pin untuk mendapatkan akses ke konten mereka. Di pc, beberapa aplikasi dapat digunakan untuk mengunci folder dan drive. Itu cotoh-contoh enkripsi istirahat. Enkripsi mengubah teks biasa menjadi teks sandi yang tidak dapat dibaca tanpa dekripsi [19]. Enkripsi dan dekripsi menggunakan kunci dan algoritma untuk melindungi data. *Virtual Private Network* (VPN) digunakan untuk mengenkripsi data di jaringan publik. VPN sangat berguna karena membantu melindungi data dari penyerang seperti *man-in-the-middle*.

Selain enkripsi, perusahaan menggunakan firewall. Firewall adalah aplikasi atau perangkat yang menganalisis *traffic* jaringan untuk mengizinkan atau menolaknya menggunakan seperangkat aturan [37]. Aturan firewall didasarkan pada alamat IP, nama domain, protokol, program, port, dan kata kunci. Firewall yang diinstal pada host adalah firewall berbasis *host*. *Microsoft Windows Defender* dan aplikasi pihak ketiga adalah contoh firewall berbasis *host*. Firewall berbasis *host* melindungi *host* yang diinstal. Firewall berbasis jaringan melindungi seluruh sumber daya dalam jaringan. Firewall dapat menjadi perangkat mandiri, perangkat bawaan yang digunakan untuk menilai traffic. Jenis firewall tingkat aplikasi meliputi; firewall penyaringan paket, firewall stateful, firewall dinamis, dan firewall proxy. *Next-Generation Firewall* adalah firewall baru dan canggih. Firewall ini memiliki inspeksi tingkat aplikasi, sistem pencegahan intrusi, intelijen ancaman eksternal, penyaringan lokasi sumber daya universal (URL), pemindaian email dan pencegahan kehilangan data (DLP). Cara-cara itu membantu perusahaan tetapi tidak cukup untuk deteksi intrusi [38].

Sekarang penelitian ini telah mencapai akurasi tinggi deteksi intrusi dengan algoritma, dapat menerapkan untuk mendeteksi intrusi di perusahaan. SVM dan ANN telah membuktikan bahwa mereka dapat menganalisis paket jaringan dan mendeteksi serangan yang telah mereka latih dan bahkan serangan baru. Masyarakat menghadapi banyak teknik intrusi. Serangan intrusi yang dihadapi perusahaan saat ini meliputi:

Rute asimetris. Router adalah perangkat yang menggabungkan dua atau lebih jaringan bersama-sama. Rute adalah proses memilih rute untuk paket di jaringan. Penentuan jalur yang diambil oleh paket tergantung pada bagaimana administrator jaringan mengkonfigurasi router untuk bekerja. Router menerima paket dari jaringan dan mengirimkannya ke jaringan lain. Router diperlukan karena jaringan memiliki protokol jaringan (IP) dan *gateway* default yang berbeda. Ada rute statis, rute dinamis dan rute default. Perutean statis menggunakan perintah 'IP *route*...' dan dinamis menggunakan 'router eigrp...' Dengan perutean asimetris, penyerang menggunakan banyak rute ke target untuk menghindari deteksi [39]. Jaringan yang tidak dikonfigurasi untuk rute asimetris akan mudah ditangkap oleh metode ini.

Teknik intrusi kedua adalah serangan khusus protokol. Perangkat memiliki banyak protokol berbeda. Banyak port dan slot membantu mereka mengetahui ke mana paket harus dikirim dan tujuan lainnya. Beberapa protokol yang digunakan meliputi; IP, TCP, UDP, Protokol Resolusi Alamat (ARP), ICMP. Protokol ini dapat dibuka, ditutup, dan menerima atau menolak paket. Dalam serangan khusus protokol, penyerang menemukan protokol terbuka atau protokol yang hidup untuk intrusi. Banyak alat atau perintah yang digunakan untuk mengumpulkan informasi protokol dari suatu sistem. Beberapa di antaranya termasuk pemindaian port atau alat sidik jari TCP/IP. Contohnya adalah Nmap, Nessus, *Angry IP Scanner*, *Wireshark*. Setelah *footprinting*, pencacahan, pemindaian, dan analisa kerentanan selesai, penyerang akan menemukan jalan ke dalam sistem.

Buffer Overflow adalah teknik intrusi yang lebih efektif. Dengan metode ini, perintah ditulis dalam beberapa sesi memori sistem [40]. Perintah akan digunakan dalam aktivitas penyerang. Hal ini dapat menyebabkan *Distributed Denial of Service* (DDOS) atau *backdoor* dalam sistem. Cara ini sulit dicegah jika ukuran buffer dijaga tetap kecil.

Skrip antarmuka *gateway* umum adalah intrusi juga. Teknik digunakan untuk memastikan komunikasi antara klien dan server tetapi mereka dapat meninggalkan jalan bagi penyusup [41].

Jurnal Teknologi dan Informasi (JATI)
Volume 13 Nomor 2 Edisi September 2023
P-ISSN 2088-2270, E-ISSN 2655-6839
DOI 10.34010/jati.v13i2

Traffic flooding adalah metode di mana penyerang mengirim banyak paket ke port TCP. Hal ini menyebabkan banyak *traffic* dan beberapa memaksa layanan *server* web berhenti atau tidak tersedia (DOS) [42].

Karena algoritma *supervised machine learning*, SVM dan ANN dapat mempelajari semua teknik intrusi dan mendeteksi mereka dan teknik baru. SVM dan ANN terdeteksi dengan AUC tinggi. Pada AUC, *True Positive* lebih tinggi daripada *True Negative*. Ini menunjukkan bahwa semua alarm yang ditiup oleh algoritma akurat. Keakuratan itu berarti masalah false alarm yang ditunjukkan oleh alat IDS dan IPS dapat dihilangkan jika algoritma digunakan untuk mendeteksi intrusi daripada IDS dan IPS tersebut. Akurasi deteksi adalah apa yang dibutuhkan karena jika alat yang tersedia memiliki lebih banyak *false* alarm, administrator TI membuang lebih banyak waktu untuk menganalisis paket yang bukan koneksi buruk atau intrusi. Buang-buang waktu sangat mengurangi produksi di perusahaan.

Penggunaan algoritma *machine learning* untuk deteksi intrusi telah mendapatkan banyak popularitas dalam beberapa tahun terakhir [24]. Itulah mengapa penting untuk meneliti metode ini untuk mendapatkan lebih banyak pengetahuan dan juga mengajarkan masyarakat tentang seberapa efektif algoritma *machine learning* dalam deteksi intrusi. Ada banyak *dataset* untuk proyek deteksi intrusi dan dataset meliputi; CAIDA, CICIDS 2017 dan ADFA-LD dan ADFA-WD. Algoritma kadang-kadang digabungkan untuk metode hibrida untuk deteksi intrusi. Metode hibrida terbukti lebih efektif juga. Misalnya, dalam penelitian ini, SVM dan ANN digunakan secara terpisah untuk mendeteksi intrusi dan dalam beberapa proyek, SVM dan ANN dapat digabungkan untuk bekerja sama. Harus diketahui bahwa algoritma memiliki fitur yang berbeda. Algoritma memiliki kekuatan dan kelemahan. Karena itu, metode hibrida mencakup kelemahan. Data dapat berupa gambar dan teks, linier dan tidak linier dan beberapa algoritma bekerja dengan baik dengan gambar dan yang lain dengan teks.

Dalam proyek ini, SVM berkinerja lebih baik daripada ANN. Tetapi karena hasilnya menunjukkan bahwa kedua algoritma tersebut baik untuk deteksi intrusi, SVM dan ANN dapat dipilih dan diimplementasikan dalam deteksi intrusi hibrida. Metode hibrida bekerja lebih efektif karena menggabungkan dua atau lebih algoritma. Penulis akan melanjutkan untuk mengimplementasikan SVM dan ANN dalam intrusi hibrida menggunakan hasil proyek ini sebagai dasar.

4. Kesimpulan

Dalam penelitian ini, algoritma *machine learning* diimplementasikan untuk mendeteksi intrusi pada *Dataset* KDD Cup 99 digunakan. *Dataset* KDD Cup 99 dikumpulkan di MIT *Lincoln Laboratory*, yang di bawah *Defense Advanced Research Project Agency* (DARPA). *Support Vector Machines* (SVM) dan *Artificial Neural Network* (ANN) adalah dua algoritma *supervised machine learning* yang digunakan dalam proyek ini. Parameter yang berbeda digunakan untuk mengukur kinerja algoritma. Parameter meliputi; akurasi pelatihan, akurasi pengujian, waktu pelatihan, waktu penguujian, AUC dan kecepatan jaringan. AUC dilakukan dalam metode *multiclass* menggunakan ovr (*one-versus-rest*). *Status* probabilitas diatur ke *true* untuk algoritma karena probabilitas *false* secara default. Ada lima (5) kelas dan AUC selesai mengambil satu kelas melawan empat lainnya. Hasil dari eksperimen (pelatihan, pengujian dan AUC) pada *Dataset* KDD CUP 99 yang diunduh dari Kaggle menunjukkan bahwa SVM dan ANN dapat secara efektif mendeteksi serangan intrusi. SVM memiliki akurasi pelatihan dan pengujian masing-masing 99,87% dan 99,81%. SVM dilatih dan diuji masing-masing dalam waktu 2 menit dan 1 menit. Pada AUC, SVM memiliki 1,00 untuk semua kelas (normal, DOS, U2R, R2L dan Probe). ANN memiliki akurasi pelatihan dan pengujian masing-masing 99,86% dan 99,85%. ANN dilatih dan diuji dalam waktu masing-masing 11 menit dan 10 detik. Pada AUC, ANN memiliki 1,00 untuk normal, DOS, Probe dan R2L tetapi 0,99 untuk U2R. Melihat hasil di atas, dapat disimpulkan bahwa kedua algoritma tersebut baik untuk deteksi intrusi tetapi SVM lebih efektif daripada ANN. Oleh karena itu saya merekomendasikan SVM untuk deteksi intrusi. Proyek ini dapat diimplementasikan di perusahaan untuk menganalisis paket jaringan, mendeteksi dan melaporkan serangan di sistem mereka. Proyek ini sangat berguna bagi semua orang. Mahasiswa yang mempelajari keamanan siber dan administrator server dapat memperoleh banyak wawasan tentang deteksi intrusi dari proyek ini. Metode hibrida bekerja lebih efektif karena menggabungkan dua atau lebih algoritma. Pada proyek selanjutnya, penulis akan menggunakan SVM dan ANN dalam hybrid intrusion detection. Keterbatasan penelitian ini adalah bahwa ia menggunakan dataset lama untuk proyek tersebut.

Daftar Pustaka

[1] C. Annamalai, "Combinatorial and Multinomial Coefficients and its Computing Techniques for Machine Learning and Cybersecurity," *The Journal of Engineering and Exact Sciences*, vol. 8, no. 8, pp. 14713–01i, 2022, doi: 10.18540/jcecvl8iss8pp14713-01i.

[2] M. Ahsan, R. Gomes, M. M. Chowdhury, and K. E. Nygard, "Enhancing Machine Learning

Prediction in Cybersecurity Using Dynamic Feature Selector," *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 199–218, 2021, doi: 10.3390/jcp1010011.

[3] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, *Machine Learning and Deep Learning Approaches for CyberSecurity: A Review*, vol. 10, no. March 2021. Springer International Publishing, 2022. doi: 10.1109/ACCESS.2022.3151248.

[4] I. D. Aiyanyo, H. Samuel, and H. Lim, "A systematic Review of Defensive and Offensive Cybersecurity With Machine Learning," *Applied Sciences (Switzerland)*, vol. 10, no. 17, pp. 1–26, 2020, doi: 10.3390/app10175811.

[5] L. Yang, J. Li, L. Yin, Z. Sun, Y. Zhao, and Z. Li, "Real-time Intrusion Detection in Wireless Network: A Deep Learning-based Intelligent Mechanism.," *IEEE Access*, vol. 8, no. August 2020, pp. 170128–170139, 2020, doi: 10.1109/ACCESS.2020.3019973.

[6] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Learning Classifiers for Intrusion Detection on Power Grids," *Ieee Transactions on Network and Service Management*, vol. 18, no. 1, pp. 1104–1116, 2021.

[7] L. Le Jeune, T. Goedeme, and N. Mentens, "Machine Learning for Misuse-Based Network Intrusion Detection: Overview, Unified Evaluation and Feature Choice Comparison Framework," *IEEE Access*, vol. 9, no. April 2021, pp. 63995–64015, 2021, doi: 10.1109/ACCESS.2021.3075066.

[8] A. Kim, M. Park, and D. H. Lee, "AI-IDS: Application of Deep Learning to Real-Time Web Intrusion Detection," *IEEE Access*, vol. 8, pp. 70245–70261, 2020, doi: 10.1109/ACCESS.2020.2986882.

[9] C. Liu, Z. Gu, and J. Wang, "A Hybrid Intrusion Detection System Based on Scalable K-Means+ Random Forest and Deep Learning," *IEEE Access*, vol. 9, no. May 2021, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[10] T. Moulahi, S. Zidi, A. Alabdulatif, and M. Atiquzzaman, "Comparative Performance Evaluation of Intrusion Detection Based on Machine Learning in In-Vehicle Controller Area Network Bus," *IEEE Access*, vol. 9, no. 4, pp. 99595–99605, 2021, doi: 10.1109/ACCESS.2021.3095962.

[11] L. Nie *et al.*, "Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach," *IEEE Transactions on Computational Social Systems*, vol. 9, no. 1, pp. 134–145, 2022, doi: 10.1109/TCSS.2021.3063538.

[12] R. Vijayanand and D. Devaraj, "A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network," *IEEE Access*, vol. 8, no. March 2020, pp. 56847–56854, 2020, doi: 10.1109/ACCESS.2020.2978035.

[13] M. Wang, K. Zheng, Y. Yang, and X. Wang, "An Explainable Machine Learning Framework for Intrusion Detection Systems," *IEEE Access*, vol. 8, no. April 2020, pp. 73127–73141, 2020, doi: 10.1109/ACCESS.2020.2988359.

[14] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, "Implementing a Deep Learning Model for Intrusion Detection on Apache Spark Platform.," *IEEE Access*, vol. 8, no. August 2020, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[15] A. Fatani, M. A. Elaziz, A. Dahou, M. A. A. Al-Qaness, and S. Lu, "IoT Intrusion Detection System Using Deep Learning and Enhanced Transient Search Optimization," *IEEE Access*, vol. 9, no. August 2021, pp. 123448–123464, 2021, doi: 10.1109/ACCESS.2021.3109081.

[16] S. Zhang, X. Xie, and Y. Xu, "A Brute-Force Black-Box Method to Attack Machine Learning-Based Systems in Cybersecurity," *IEEE Access*, vol. 8, no. July 2021, pp. 128250–128263, 2020, doi: 10.1109/ACCESS.2020.3008433.

[17] G. Apruzzese *et al.*, "The Role of Machine Learning in Cybersecurity," *Digital Threats: Research and Practice*, vol. 4, no. 1, 2023, doi: 10.1145/3545574.

[18] I. Chua, T.H., Salam, "Evaluation of Machine Learning Algorithms in Network-Based Intrusion Detection System.," *Lecture Notes in Electrical Engineering*, vol. 740, no. 2021, pp. 229–236, 2021, doi: 10.1007/978-981-33-6393-9_24.

[19] S. Sajja, G., Mustafa, M., Ponnusamy, R., Abdufattokhov, "Machine Learning Algorithms in Intrusion Detection and Classification," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 6, pp. 12211–12219, 2021, [Online]. Available: http://annalsofrscb.ro/index.php/journal/article/view/7837

[20] K. S. Kajal, A., Nandal, "A Hybrid Approach for Cyber Security: Improved Intrusion Detection System Using ANN-SVM," *Indian Journal of Computer Science and Engineering*, vol. 11, no. 4, pp. 412–425, 2020, doi: 10.21817/indjcse/2020/v11i4/201104300.

[21] V. Sstla, V. K. K. Kolli, L. K. Voggu, R. Bhavanam, and S. Vallabhasoyula, "Predictive Model for Network Intrusion Detection System Using Deep Learning," *Revue d'Intelligence Artificielle*, vol. 34, no. 3, pp. 323–330, 2020, doi: 10.18280/ria.340310.

[22] A. M. Salih, A.A., Abdulazeez, "Evaluation of Classification Algorithms for Intrusion Detection

System: A Review," *Journal of Soft Computing and Data Mining*, vol. 2, no. 1, pp. 31–40, 2021, doi: 10.30880/jscdm.2021.02.01.004.

[23] C. C. O'Mahony, D.G., Curran, T.J., Harris, J.P., Murphy, "Interference and Intrusion in Wireless Sensor Networks," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 2, pp. 1–14, 2020, doi: 10.1109/MAES.2020.2970262.

[24] G. C. Padmaja, B., Sravan, K.S., Patro, E.K.R., Sekhar, "A System to Automate the Development of Anomaly-Based Network Intrusion Detection Model," *Journal of Physics: Conference Series*, vol. 2089, no. 1, pp. 1–15, 2021, doi: 10.1088/1742-6596/2089/1/012006.

[25] B. Venkataramana, K. C. Mouli, and A. Eenaja, "Network Intrusion Detection By SVM & ANN With Feature Selection," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 8, no. 6, pp. 3704–3708, 2020, [Online]. Available: http://ijcrt.org/viewfull.php?&p_id=IJCRT2006506

[26] N. Alagrash, Y., Drebee, A., Zirjawi, "Comparing the Area of Data Mining Algorithms in Network Intrusion Detection," *Journal of Information Security*, vol. 11, no. 1, pp. 1–18, 2020, doi: 10.4236/jis.2020.111001.

[27] T. S. Ustun, S. M. Suhail Hussain, A. Ulutas, A. Onen, M. M. Roomi, and D. Mashima, "Machine Learning-based Intrusion Detection for Achieving Cybersecurity in Smart Grids Using IEC 61850 GOOSE Messages," *Symmetry*, vol. 13, no. 5, pp. 1–15, 2021, doi: 10.3390/sym13050826.

[28] A. Imran, H., Zulfiqar, M.A., Arshad, "Machine Learning Based Intrusion Detection System," *Journal of Computational and Cognitive Engineering*, vol. 2, no. 2, pp. 140–149, 2021, doi: 10.4018/978-1-7998-3327-7.ch011.

[29] M. M. Mijwil, I. E. Salem, and M. M. Ismaeel, "The Significance of Machine Learning and Deep Learning Techniques in Cybersecurity: A Comprehensive Review," *Iraqi Journal for Computer Science and Mathematics*, vol. 4, no. 1, pp. 87–101, 2023, doi: 10.52866/ijcsm.2023.01.01.008.

[30] C. F. M. Maseer, K.Z., Yusof, R., Bahaman, N., Mostafa, S.A., Foozy, "Benchmarking of Machine Learning for Anomaly Based Intrusion Detection Systems in the CICIDS2017 Dataset," *IEEE Access*, vol. 9, no. 22 January, pp. 22351–22370, 2021, doi: 10.1109/ACCESS.2021.3056614.

[31] M. Elsisi, M. Q. Tran, K. Mahmoud, Di. E. A. Mansour, M. Lehtonen, and M. M. F. Darwish, "Towards Secured Online Monitoring for Digitalized GIS against Cyber-Attacks Based on IoT and Machine Learning," *IEEE Access*, vol. 9, no. May 2021, pp. 78415–78427, 2021, doi: 10.1109/ACCESS.2021.3083499.

[32] Y. Kasongo, M.S., Sun, "Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020, doi: 10.1186/s40537-020-00379-6.

[33] S. B. Dilip, R., Samanvita, N., Pramodhini, R., Vidhya, G.S., Telkar, "Performance Analysis of Machine Learning Algorithms in Intrusion Detection and Classification," *Communications in Computer and Information Science*, vol. 1591 CCIS, no. May, pp. 283–289, 2022, doi: 10.1007/978-3-031-07012-9_25.

[34] J. Lansky *et al.*, "Deep Learning-Based Intrusion Detection Systems: A Systematic Review," *IEEE Access*, vol. 9, no. July 2021, pp. 101574–101599, 2021, doi: 10.1109/ACCESS.2021.3097247.

[35] K. Shin, Y., Kim, "Comparison of Anomaly Detection Accuracy of Host-Based Intrusion Detection Systems Based on Different Machine Learning Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 252–259, 2020, doi: 10.14569/ijacsa.2020.0110233.

[36] Y. K. Rokade, M.D., Sharma, "MLIDS: A Machine Learning Approach for Intrusion Detection for Real Time Network Dataset," *2021 International Conference on Emerging Smart Computing and Informatics, ESCI 2021*, vol. 15, no. March, pp. 533–536, 2021, doi: 10.1109/ESCI50559.2021.9396829.

[37] S. Dini, P., Saponara, "Analysis, Design, and Comparison of Machine-Learning Techniques for Networking Intrusion Detection," *Designs*, vol. 5, no. 1, pp. 1–22, 2021, doi: 10.3390/designs5010009.

[38] K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A Survey on Machine Learning Techniques for Cyber Security in the Last Decade," *IEEE Access*, vol. 8, no. 2020, pp. 222310–222354, 2020, doi: 10.1109/ACCESS.2020.3041951.

[39] A. Sharma, P.K., Gosain, D., Sagar, H., Kumar, C., Dogra, "SiegeBreaker: An SDN Based Practical Decoy Routing System," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 3, pp. 243–263, 2020, doi: 10.2478/popets-2020-0051.

[40] Y. Butt, M., Ajmal, Z., Khan, Z.I., Idrees, M., Javed, "An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques," *Applied Sciences (Switzerland)*, vol. 12, no. 13, pp. 1–31, 2022,

doi: 10.3390/app12136702.

[41] P. Tanakas, A. Ilias, and N. Polemi, "A Novel System for Detecting and Preventing SQL Injection and Cross-Site-Script," *International Conference on Electrical, Computer, and Energy Technologies, ICECET 2021*, vol. 1, no. September, pp. 1–7, 2021, doi: 10.1109/ICECET52533.2021.9698688.

[42] 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, 2020, doi: 10.1080/24751839.2020.1767484.