IMPLEMENTASI *INTRUSION DETECTION SYSTEM* MENGGUNAKAN *SNORT* BERBASIS *PARALLEL COMPUTING* DENGAN METODE *LOG DISTRIBUTION*

SKRIPSI

Diajukan Untuk Mengajukan Salah Satu Syarat

Memperoleh Gelar Sarjana Komputer

Oleh

Ahmad Gimnastiar

09011282126050

JURUSAN SISTEM KOMPUTER

FAKULTAS ILMU KOMPUTER

UNIVERSITAS SRIWIJAYA

2025

HALAMAN PENGESAHAN

SKRIPSI

IMPLEMENTASI *INTRUSION DETECTION SYSTEM* MENGGUNAKAN *SNORT* BERBASIS *PARALLEL COMPUTING* DENGAN METODE *LOG DISTRIBUTION*

Sebagai salah satu syarat untuk penyelesaian studi di

Program Studi S1 Sistem Komputer

Oleh:

Ahmad Gimnastiar

09011282126050

Pembimbing 1 : **Dr. Ir. Ahmad Heryanto, M.T.**

NIP. 198701222015041002

Mengetahui,

Ketua Jurusan Sistem Komputer

Dr. Ir. H. Sukemi, M.T.

196612032006041001

AUTHENTICATION PAGE

SKRIPSI

IMPLEMENTATION OF INTRUSION DETECTION SYSTEM USING SNORT BASED ON PARALLEL COMPUTING WITH LOG DISTRIBUTION METHOD

As one of the requirements for completing the

Bachelor's Degree Program in Computer Systems

By

Ahmad Gimnastiar

09011282126050

Advisor 1 : **Dr. Ir. Ahmad Heryanto, M.T.**

NIP. 198701222015041002

Approved by,

Head of Computer System Department

Dr. Ir. H. Sukemi, M.T.

196612032006041001

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada :

Hari : Jumat

Tanggal : 11 Juli 2025

Tim Penguji

1.

2. **Ketua** : **Prof. Deris Stiawan, M.T., Ph.D.**

3. **Penguji** : **Dr. Ir. H. Sukemi, M.T.**

4. **Pembimbing** : **Dr. Ir. Ahmad Heryanto, M.T.**

Mengetahui, 4/9/25

Ketua Jurusan Sistem Komputer

Dr. Ir. H. Sukemi, M.T.

NIP. 196612032006041001

LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini :

Nama : Ahmad Gimnastiar

NIM : 09011282126050

Judul : Implementasi Intrusion Detection System Menggunakan Snort Berbasis Parallel Computing Dengan Metode Log Distribution

Hasil pemeriksaan *Software* Turnitin : 1%

Menyatakan bahwa laporan skripsi saya merupakan hasil karya sendiri dan bukan hasil penjiplakan atau plagiat. Apabila ditemukan unsur penjiplakan atau plagiat dalam laporan tugas akhir ini, maka saya bersedia menerima sanksi akademik dari Universitas Sriwijaya.

Demikian, pernyataan ini saya buat dalam keadaan sadar dan tanpa paksaan dari siapapun.

Palembang, Agustus 2025

Ahmad Gimnastiar

NIM. 09011282126050

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "IMPLEMENTASI *INTRUSION DETECTION SYSTEM* MENGGUNAKAN *SNORT* BERBASIS *PARALLEL COMPUTING* DENGAN METODE *LOG DISTRIBUTION*". Skripsi ini disusun sebagai salah satu syarat untuk menyelesaikan program Sarjana pada jurusan Sistem Komputer, Universitas Sriwijaya.

Dalam penyusunan skripsi ini, penulis mendapatkan banyak bantuan, dukungan, dan bimbingan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT, yang telah melimpahkan Berkat dan Rahmat-Nya.
2. Keluarga saya yang telah menjadi salah satu pendukung dalam penulisan skripsi.
3. Bapak Prof. Dr. Erwin, M.Si. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. Sukemi, M.T. selaku Ketua Jurusan Sistem Komputer Universitas Sriwijaya.
5. Bapak Dr. Ir. Ahmad Heryanto, M.T. selaku Pembimbing Tugas Akhir.
6. Bapak Muhammad Ali Buchari, S.Kom., M.T. selaku Dosen Pembimbing Akademik.
7. Kak Angga Yunanda selaku admin Jurusan Sistem Komputer
8. Sahabat INGPO yang menemani serta berbagi tawa dan luka bersama-sama.
9. Teman-teman Tim Cloud yang berjuang bersama dari awal sampai akhir.
10. Dan semua pihak yang telah membantu dalam penulisan tugas akhir.

Skripsi ini membahas implementasi sistem deteksi intrusi pada jaringan lokal (LAN) menggunakan aplikasi Snort yang dioptimalkan dengan tekno*log*i *parallel computing*. Penelitian ini diharapkan dapat memberikan kontribusi positif

dalam bidang keamanan jaringan komputer, khususnya dalam mendeteksi dan menangani potensi ancaman terhadap jaringan lokal.

Penulis menyadari bahwa skripsi ini masih memiliki kekurangan. Oleh karena itu, penulis membuka diri untuk kritik dan saran yang membangun demi kesempurnaan di masa yang akan datang. Semoga skripsi ini dapat bermanfaat bagi pembaca dan memberikan inspirasi bagi pengembangan ilmu pengetahuan, khususnya dalam bidang keamanan jaringan dan *parallel computing*.

Palembang, Agustus 2025

Ahmad Gimnastiar

NIM. 09011282126050

IMPLEMENTASI *INTRUSION DETECTION SYSTEM* MENGGUNAKAN SNORT BERBASIS *PARALLEL COMPUTING* DENGAN METODE *LOG DISTRIBUTION*

AHMAD GIMNASTIAR (09011282126050)
Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya
Email : ahmadgimnastiar13@gmail.com

ABSTRAK

Penelitian ini mengembangkan *Intrusion Detection System* (IDS) berbasis Snort yang dioptimalkan menggunakan *parallel computing* dengan *Message Passing Interface* (MPI). Sistem dibangun dengan arsitektur master dan dua *worker* yang menjalankan Snort secara paralel untuk mendeteksi lalu lintas ICMP. *Log* hasil deteksi dikumpulkan melalui *shared folder* dan dimonitor oleh master menggunakan skrip Python, yang juga mengirimkan notifikasi email otomatis bila ancaman kritis terdeteksi. Pengujian dilakukan dalam empat skenario: tanpa *worker*, dengan 1 *worker*, 2 *worker*, dan 1 *worker* dengan tuning konfigurasi. Hasil menunjukkan peningkatan efisiensi penggunaan CPU, memori, dan energi secara signifikan. Sistem dengan 2 *worker* memberikan kinerja terbaik, sementara tuning pada 1 *worker* menghasilkan konsumsi energi paling rendah. Hasil ini membuktikan bahwa pendekatan *parallel computing* dapat meningkatkan efektivitas sistem IDS dan layak diterapkan pada jaringan berskala kecil hingga menengah.

Kata kunci: *Intrusion Detection System*, Snort, MPI, *Parallel Computing*, ICMP.

IMPLEMENTATION OF *INTRUSION DETECTION SYSTEM* USING SNORT BASED ON *PARALLEL COMPUTING* WITH *LOG DISTRIBUTION* METHOD

AHMAD GIMNASTIAR (09011282126050)
Department of Computer Systems, Faculty of Computer Science, Sriwijaya University
Email : ahmadgimnastiar13@gmail.com

ABSTRACT

This research develops an Intrusion Detection System (IDS) based on Snort, optimized using parallel computing with the Message Passing Interface (MPI). The system is built with an architecture consisting of one master and two workers running Snort in parallel to detect ICMP traffic. Detection logs are collected through a shared folder and monitored by the master using a Python script, which also sends automatic email notifications when critical threats are detected. Testing was carried out in four scenarios: without a worker, with 1 worker, 2 workers, and 1 worker with configuration tuning. The results show significant improvements in CPU, memory, and energy efficiency. The 2-worker configuration delivered the best performance, while tuning on the 1-worker setup resulted in the lowest energy consumption. These findings demonstrate that the parallel computing approach can effectively improve IDS performance and is suitable for deployment in small to medium-scale networks.

Keywords: Intrusion Detection System, Snort, MPI, Parallel Computing, ICMP.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kemajuan tekno*log*i informasi dan komunikasi telah membawa perubahan besar dalam berbagai aspek kehidupan, termasuk dalam pengelolaan dan pengamanan jaringan komputer. Seiring dengan meningkatnya penggunaan jaringan lokal (LAN) di berbagai sektor, seperti pendidikan, pemerintahan, dan perusahaan, ancaman terhadap keamanan jaringan pun turut berkembang. Serangan seperti pencurian data, akses tidak sah, dan perusakan sistem menjadi tantangan serius yang memerlukan penanganan segera dan efektif.

Keamanan siber dianggap sebagai bagian penting dari jaringan terdistribusi dan sistem komputasi. Penyerang dari seluruh dunia terus berusaha mendapatkan akses tanpa izin ke berbagai lingkungan jaringan dengan tujuan mencuri, mengubah, mengekspos, menonaktifkan, atau menghancurkan data dan aplikasi. Para ahli keamanan siber merancang alat dan menerapkan praktik untuk melindungi pengguna serta mengurangi kerentanan jaringan [1].

Meskipun mereka berhasil menjaga keamanan lingkungan jaringan yang menjadi tanggung jawab mereka, solusi yang mereka terapkan dapat dengan cepat menjadi usang dan tidak dapat digunakan lagi ketika infrastruktur jaringan berubah. Hal ini menciptakan kebutuhan untuk merancang solusi adaptif yang dapat digeneralisasi dengan baik saat ancaman baru muncul dan ketika diterapkan di lingkungan jaringan yang baru [2].

Salah satu solusi untuk menghadapi tantangan tersebut adalah dengan menerapkan sistem deteksi intrusi (*Intrusion Detection System*/IDS). IDS bertugas memantau dan menganalisis lalu lintas jaringan untuk mendeteksi aktivitas mencurigakan yang berpotensi membahayakan sistem [3][4]. Salah satu aplikasi IDS yang populer adalah Snort, sebuah perangkat lunak *open-source* yang mampu mendeteksi ancaman dengan mengandalkan berbagai aturan (*rules*) yang telah ditentukan [5].

Penelitian oleh Erlacher dan Dressler (2020) memperkenalkan FIXIDS, yaitu sistem IDS berbasis flow yang menggunakan *signature* Snort dalam format IPFIX, dan menunjukkan performa yang lebih tinggi dibandingkan Snort biasa pada jaringan berkecepatan tinggi tanpa mengorbankan tingkat deteksi. Penelitian ini relevan karena menekankan pentingnya efisiensi dalam sistem IDS dan eksplorasi pendekatan baru seperti *flow-based detection*[2].

Namun, implementasi Snort pada jaringan dengan lalu lintas tinggi seringkali menghadapi kendala performa. Proses analisis data yang memerlukan waktu dan sumber daya yang besar dapat menyebabkan keterlambatan dalam mendeteksi serangan. Untuk mengatasi hal tersebut, tekno*log*i *parallel computing* dapat diterapkan untuk meningkatkan kinerja sistem. Dengan memanfaatkan pendekatan ini, proses pemantauan dan analisis lalu lintas jaringan dapat dilakukan secara simultan oleh beberapa unit pemrosesan (*worker*), sehingga mempercepat waktu respons terhadap ancaman [6].

Dalam penelitian ini, Snort akan diintegrasikan dengan tekno*log*i *parallel computing* menggunakan *Message Passing Interface* (MPI). Metode ini memungkinkan pembagian tugas di antara beberapa *worker* yang bekerja secara bersamaan untuk memantau segmen jaringan tertentu [7]. Hasil deteksi dari setiap *worker* akan dikumpulkan dalam sebuah *shared folder*, sehingga dapat diakses dan dianalisis oleh *master node* untuk menentukan langkah mitigasi yang diperlukan, seperti memutuskan koneksi atau menonaktifkan perangkat yang terinfeksi.

Monteiro dan Santos (2023) menggarisbawahi pentingnya *parallel computing* untuk menyelesaikan perhitungan intensif dalam konteks finansial, namun aplikasinya juga relevan untuk pengolahan data IDS dalam volume besar. Mereka menggunakan GPU dan metode non-parametrik dalam komputasi risiko, memperlihatkan bahwa komputasi paralel dapat mempercepat proses estimasi kompleks yang memerlukan iterasi tinggi. Lebih lanjut, Gao et al. (2025) memperkenalkan pendekatan baru dalam runtime paralel tasking berbasis arsitektur *message passing* (EMP) untuk mengurangi *overhead* pada pemrosesan tugas granular. Meskipun studi ini tidak secara langsung membahas IDS, teknik yang

ditawarkan sangat relevan untuk meningkatkan efisiensi pemrosesan paralel dalam sistem distribusi seperti IDS berbasis MPI

Penelitian ini menghadirkan kebaruan dalam penerapan sistem deteksi intrusi (IDS) dengan mengintegrasikan Snort ke dalam arsitektur *parallel computing* menggunakan *Message Passing Interface* (MPI). Pendekatan ini memungkinkan proses deteksi ancaman dilakukan secara paralel oleh beberapa *node* (*worker*), yang sebelumnya jarang diterapkan dalam implementasi IDS berbasis *open-source*. Selain itu, penelitian ini membangun sistem sinkronisasi *log* melalui *shared folder* yang memungkinkan *master* memantau hasil deteksi dari seluruh *worker* secara *real-time*. Fitur tambahan berupa sistem *monitoring* menggunakan Python juga dikembangkan untuk mendeteksi *log* dengan tingkat ancaman tinggi dan mengirimkan notifikasi secara otomatis melalui email, meningkatkan responsivitas sistem terhadap serangan kritis. Penelitian ini juga melakukan optimalisasi performa Snort melalui *tuning* konfigurasi *rules* dan penggunaan metode pencarian yang lebih efisien, serta untuk menstabilkan penggunaan CPU. Evaluasi performa dilakukan secara kuantitatif dengan membandingkan sistem IDS *parallel* dan tradisional berdasarkan konsumsi sumber daya seperti CPU, memori, *disk*, dan suhu prosesor. Hasil penelitian menunjukkan bahwa pendekatan *parallel* memberikan efisiensi dan kinerja yang lebih baik dalam menangani lalu lintas jaringan yang padat.

Dengan latar belakang tersebut, penelitian ini bertujuan untuk mengembangkan sistem deteksi intrusi berbasis Snort yang dioptimalkan menggunakan *parallel computing*. Sistem ini diharapkan mampu meningkatkan efisiensi dan kecepatan dalam mendeteksi ancaman pada jaringan LAN, serta memberikan kontribusi nyata dalam bidang keamanan jaringan komputer.

1.2 Tujuan dan Manfaat Penelitian

Berikut tujuan dan manfaat dari penelitian ini adalah sebagai berikut:

1.2.1. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1. Mengembangkan sistem deteksi intrusi pada jaringan LAN dengan memanfaatkan tekno*log*i *parallel computing* berbasis MPI.
2. Mengoptimalkan kinerja aplikasi Snort dalam mendeteksi ancaman pada jaringan dengan lalu lintas data yang tinggi.
3. Memberikan solusi efisien dalam pembagian tugas *monitoring* jaringan antar *worker* untuk meningkatkan kecepatan dan akurasi deteksi ancaman.
4. Menentukan kinerja sistem yang paling efisien dengan membandingkan beberapa *worker* yang bekerja dalam IDS berbasis *parallel computing*.

1.2.2. Manfaat Penelitian

Berikut manfaat dari penelitian ini diantaranya sebagai berikut.

1. Memberikan kontribusi pada pengembangan ilmu pengetahuan di bidang keamanan jaringan dan *parallel computing*, serta menjadi referensi bagi penelitian serupa di masa depan.
2. Menghasilkan sistem deteksi intrusi yang lebih efisien dan cepat, yang dapat diterapkan dalam lingkungan jaringan lokal dengan lalu lintas tinggi.
3. Meningkatkan kesadaran akan pentingnya keamanan jaringan, serta memberikan solusi tekno*log*i yang dapat digunakan untuk melindungi data dan aset digital dari ancaman siber.
4. Mengurangi potensi kerugian yang diakibatkan oleh serangan siber melalui penerapan sistem deteksi intrusi yang lebih efektif.

1.3 Perumusan dan Batasan Masalah

Penelitian ini memiliki rumusan dan batasan masalah seperti berikut.

1.3.1. Rumusan Masalah

Berikut rumusan masalah pada penelitian ini.

1. Bagaimana implementasi Snort pada jaringan LAN dengan tekno*log*i *parallel computing* berbasis MPI?
2. Bagaimana cara membagi tugas *monitoring* jaringan secara efisien di antara beberapa *worker*?
3. Bagaimana efektivitas sistem ini dalam mendeteksi ancaman jaringan dibandingkan dengan metode tradisional?

1.3.2. Batasan Masalah

Agar penelitian ini lebih terfokus dan terarah, maka penelitian ini dibatasi oleh beberapa hal berikut.

1. Sistem deteksi intrusi hanya diimplementasikan pada jaringan lokal (LAN) dengan protokol ICMP (*ping*) sebagai kasus uji utama.
2. Penggunaan Snort difokuskan pada mode Network *Intrusion Detection System* (NIDS) dengan konfigurasi tertentu untuk mendeteksi ancaman berbasis aturan (*rule*-based *detection*) [8].
3. Tekno*log*i *parallel computing* yang digunakan adalah *Message Passing Interface* (MPI) dengan pembagian tugas antar *worker*.
4. Analisis kinerja sistem mencakup penggunaan sumber daya seperti *processor*, *memory*, dan *disk* serta akurasi deteksi ancaman.
5. Penelitian tidak mencakup mitigasi ancaman secara otomatis, namun hanya memberikan notifikasi ke *master node* untuk pengambilan tindakan manual.
6. Penelitian dilakukan pada lingkungan simulasi dengan infrastruktur *virtual* menggunakan perangkat lunak dan perangkat keras terbatas.

1.4 Metodo*log*i Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yang melibatkan implementasi tekno*log*i *parallel computing* berbasis MPI pada sistem deteksi intrusi menggunakan Snort. Tahapan metodo*log*i penelitian ini adalah sebagai berikut:

1.4.1. Studi Literatur

Pada tahap ini, peneliti mengumpulkan berbagai referensi dari jurnal ilmiah, buku, dan artikel yang berkaitan dengan sistem deteksi intrusi, tekno*log*i *parallel computing*, serta implementasi perangkat lunak Snort. Referensi yang diperoleh digunakan sebagai dasar pemahaman terhadap prinsip kerja Snort dan cara mengintegrasikan MPI dalam membangun sistem deteksi terdistribusi. Pengetahuan dasar ini menjadi fondasi untuk tahapan selanjutnya dalam perancangan dan pengembangan sistem.

1.4.2. Perancangan Sistem

Tahap ini mencakup proses mendesain arsitektur sistem yang terdiri dari satu master node dan beberapa *worker* node. Pada tahap ini pula ditentukan metode pembagian tugas monitoring lalu lintas jaringan antar node menggunakan pendekatan *parallel*. Perancangan juga mencakup penentuan struktur dan format penyimpanan *log* hasil deteksi pada *shared folder*, agar dapat diakses dengan mudah oleh master untuk keperluan analisis dan tindak lanjut.

1.4.3. Implementasi Sistem

Implementasi dimulai dengan proses instalasi dan konfigurasi Snort pada setiap node, baik master maupun *worker*. Setelah itu, MPI diintegrasikan ke dalam sistem untuk mendistribusikan tugas pemantauan jaringan secara paralel ke beberapa *worker*. Penyesuaian aturan deteksi Snort juga dilakukan agar dapat mengenali lalu lintas ICMP (ping) yang dijadikan sebagai skenario utama dalam pengujian sistem.

1.4.4. Pengujian Sistem

Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun. Pengujian dilakukan dengan cara mengirimkan lalu lintas jaringan berupa perintah ping ke IP yang dipantau oleh *worker*. Selanjutnya, diperiksa apakah *log* hasil deteksi dapat tersimpan di *shared folder* sesuai dengan konfigurasi sistem. Sistem juga diuji untuk memastikan bahwa master node dapat menerima notifikasi ancaman dari *worker* ketika terdeteksi adanya aktivitas mencurigakan.

1.4.5. Analisis Hasil

Setelah pengujian dilakukan, data yang diperoleh dianalisis untuk mengevaluasi performa sistem. Evaluasi dilakukan berdasarkan kecepatan deteksi ancaman serta penggunaan sumber daya seperti prosesor, memori, disk, dan konsumsi energi. Hasil sistem *parallel* kemudian dibandingkan dengan pendekatan tradisional untuk mengukur efektivitas dan efisiensi dari penerapan *parallel computing* dalam sistem deteksi intrusi ini.

1.4.6. Penyusunan Laporan

Tahap akhir dari metodo*log*i ini adalah pendokumentasian seluruh proses penelitian, dimulai dari studi literatur hingga analisis hasil. Penulisan laporan mencakup penyusunan kesimpulan dan pemberian rekomendasi berdasarkan temuan yang diperoleh. Dengan metode ini, diharapkan hasil penelitian dapat memberikan kontribusi praktis terhadap pengembangan sistem keamanan jaringan lokal.

1.5 Sistematika Penulisan

Adapun Sistematika Penulisan dalam tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bagian ini memberikan uraian awal dari suatu penulisan, meliputi latar belakang, perumusan dan batasan masalah, tujuan dan manfaat, metodo*log*i penelitian, serta sistematikan penulisan.

BAB II TINJAUAN PUSTAKA

Bagian ini memaparkan teori-teori dasar yang menjadi landasan penelitian yang dilakukan.

BAB III METODO*LOG*I PENELITIAN

Bagian ini berisi penjelasan detail mengenai teknik, metode, serta alur proses yang digunakan dalam penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bagian ini menjelaskan hasil pengujian yang diperoleh dan menjelaskan analisa terhadap hasil penelitian yang telah dilakukan.

BAB V KESIMPULAN

Bagian ini menjelaskan hasil pengujian yang diperoleh dan menjelaskan analisa terhadap hasil penelitian yang telah dilakukan dan rekomendasi untuk perbaikan dan pengembangan penelitian di masa depan.

DAFTAR PUSTAKA

[1] P. Sarantos, J. Violos, and A. Leivadeas, “Enabling semi-supervised learning in intrusion detection systems,” *J. Parallel Distrib. Comput.*, vol. 196, no. November 2024, p. 105010, 2025, doi: 10.1016/j.jpdc.2024.105010.

[2] F. Erlacher and F. Dressler, “On High-Speed Flow-Based Intrusion Detection Using Snort-Compatible Signatures,” *IEEE Trans. Dependable Secur. Comput.*, vol. 19, no. 1, pp. 495–506, 2022, doi: 10.1109/TDSC.2020.2973992.

[3] Q. Sun, “Intrusion Detection in High-Performance Computing,” pp. 1–18, 2024.

[4] R. Dhahbi and F. Jemili, “A Deep Learning Approach for Intrusion Detection,” *2021 IEEE 23rd Int. Conf. High Perform. Comput. Commun. 7th Int. Conf. Data Sci. Syst. 19th Int. Conf. Smart City 7th Int. Conf. Dependability Sensor, Cl*, no. December 2021, pp. 1211–1218, 2022, doi: 10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00186.

[5] U. Snort, “Using SNORT 8,” 1998, doi: 10.1016/B978-0-12-803584-9/00008-1.

[6] B. Rexworthy, “Intrusion detections systems - an outmoded network protection model,” *Netw. Secur.*, vol. 2020, no. 6, pp. 17–19, 2020, doi: 10.1016/S1353-4858(09)70065-0.

[7] W. Gropp, E. Lusk, N. Doss, and A. Skjellum, “A high-performance, portable implementation of the MPI message passing interface standard,” *Parallel Comput.*, vol. 22, no. 6, pp. 789–828, 1996, doi: 10.1016/0167-8191(96)00024-5.

[8] U. Intrusion *et al.*, “Implementing Intrusion Detection Systems,” *Secur. Sage’s Guid. to Hardening Netw. Infrastruct.*, pp. 369–425, 2004, doi: 10.1016/b978-193183601-2/50014-2.

[9] P. Sarantos, J. Violos, and A. Leivadeas, “Enabling semi-supervised learning in intrusion detection systems,” *J. Parallel Distrib. Comput.*, vol. 196, no. October 2023, p. 105010, 2025, doi: 10.1016/j.jpdc.2024.105010.

[10] R. Agustina, “OPTIMALISASI ALGORITMA LSTM PADA SISTEM PENDETEKSI SERANGAN BRUTEFORCE MENGGUNAKAN METODE BIDIRECTIONAL LSTM (Bi-LSTM) SKRIPSI,” *Repository.Unsri.Ac.Id*, 2023.

[11] A. M. Monteiro and A. A. F. Santos, “Parallel computing in finance for estimating risk-neutral densities through option prices,” *J. Parallel Distrib. Comput.*, vol. 173, pp. 61–69, 2023, doi: 10.1016/j.jpdc.2022.11.010.

[12] X. Gao, L. Chen, H. Wang, H. Cui, and X. Feng, “Scalable tasking runtime with parallelized builders for explicit message passing architectures,” *Parallel Comput.*, vol. 123, no. March 2024, p. 103124, 2025, doi: 10.1016/j.parco.2024.103124.

[13] A. Meryem and B. EL Ouahidi, “Hybrid intrusion detection system using machine learning,” *Netw. Secur.*, vol. 2020, no. 5, pp. 8–19, 2020, doi: 10.1016/S1353-4858(20)30056-8.

[14] J. Kizza and F. Migga Kizza, “Intrusion Detection and Prevention Systems,” *Secur. Inf. Infrastruct.*, pp. 239–258, 2011, doi: 10.4018/978-1-59904-379-1.ch012.

[15] E. Shoop, S. J. Matthews, R. Brown, and J. C. Adams, “Hands-on parallel & distributed computing with Raspberry Pi devices and clusters,” *J. Parallel Distrib. Comput.*, vol. 196, no. October 2024, p. 104996, 2025, doi: 10.1016/j.jpdc.2024.104996.

[16] K. B. Ferreira and S. Levy, “Evaluating MPI resource usage summary statistics,” *Parallel Comput.*, vol. 108, no. August, p. 102825, 2021, doi: 10.1016/j.parco.2021.102825.

[17] M. G. F. Dosanjh *et al.*, “Implementation and evaluation of MPI 4.0 partitioned communication libraries,” *Parallel Comput.*, vol. 108, no.

August, p. 102827, 2021, doi: 10.1016/j.parco.2021.102827.

[18] E. Nuriyev, J. A. Rico-Gallego, and A. Lastovetsky, “Model-based selection of optimal MPI broadcast algorithms for multi-core clusters,” *J. Parallel Distrib. Comput.*, vol. 165, pp. 1–16, 2022, doi: 10.1016/j.jpdc.2022.03.012.

[19] G. A. V. Lavik and G. Sivertsen, “Erih Plus - Making the Ssh Visible, Searchable and Available,” *Procedia Comput. Sci.*, vol. 106, no. 1877, pp. 61–65, 2020, doi: 10.1016/j.procs.2017.03.035.

[20] J. Sabin, “The future of security in a remote-work environment,” *Netw. Secur.*, vol. 2021, no. 10, pp. 15–17, 2021, doi: 10.1016/S1353-4858(21)00118-5.

[21] M. Aldinucci *et al.*, “Practical parallelization of scientific applications with OpenMP, OpenACC and MPI,” *J. Parallel Distrib. Comput.*, vol. 157, pp. 13–29, 2021, doi: 10.1016/j.jpdc.2021.05.017.

[22] C. Y. Chen, “Scheduling coflows for minimizing the total weighted completion time in heterogeneous parallel networks,” *J. Parallel Distrib. Comput.*, vol. 182, p. 104752, 2023, doi: 10.1016/j.jpdc.2023.104752.

[23] A. Jocksch, N. Ohana, E. Lanti, E. Koutsaniti, V. Karakasis, and L. Villard, “An optimisation of allreduce communication in message-passing systems,” *Parallel Comput.*, vol. 107, no. June, p. 102812, 2021, doi: 10.1016/j.parco.2021.102812.

[24] Y. Wang, W. Meng, W. Li, J. Li, W. X. Liu, and Y. Xiang, “A fog-based privacy-preserving approach for distributed signature-based intrusion detection,” *J. Parallel Distrib. Comput.*, vol. 122, pp. 26–35, 2018, doi: 10.1016/j.jpdc.2018.07.013.

[25] A. Gupta and M. Kalra, “Intrusion detection and prevention system using cuckoo search algorithm with ANN in cloud computing,” *PDGC 2020 - 2020 6th Int. Conf. Parallel, Distrib. Grid Comput.*, pp. 66–72, 2020, doi: 10.1109/PDGC50313.2020.9315771.

[26] Y. Y. Aung and M. M. Min, “An analysis of random forest algorithm based network intrusion detection system,” *Proc. - 18th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2017*, pp. 127–132, 2017, doi: 10.1109/SNPD.2017.8022711.

[27] Y. Zhao, “Network intrusion detection system model based on data mining,” *2016 IEEE/ACIS 17th Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2016*, pp. 155–160, 2016, doi: 10.1109/SNPD.2016.7515894.

[28] S. N. Ghabel, M. Z. Lighvan, A. M. Baklou, and L. M. Khanli, “Search acceleration in preprocessing mechanism of network intrusion detection systems using graphics processors,” *2015 7th Conf. Inf. Knowl. Technol. IKT 2015*, 2015, doi: 10.1109/IKT.2015.7288744.

[29] J. Nam, M. Jamshed, B. Choi, D. Han, and K. S. Park, “Haetae: Scaling the performance of network intrusion detection with many-core processors,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9404, pp. 89–110, 2015, doi: 10.1007/978-3-319-26362-5_5.

[30] N. Z. Tarapore, D. B. Kulkarni, and V. K. Prasad, “Implementation of parallel algorithm for support vector machine applied to intrusion detection systems,” *Int. Conf. Comput. Anal. Secur. Trends, CAST 2016*, pp. 179–184, 2017, doi: 10.1109/CAST.2016.7914962.

[31] A. R. Widianto, C. Lim, and I. E. Kho, “Improving performance of intrusion detection system using OpenCL based general-purpose computing on Graphic Processing Unit (GPGPU),” *CONMEDIA 2015 - Int. Conf. New Media 2015*, pp. 3–7, 2016, doi: 10.1109/CONMEDIA.2015.7449146.

[32] R. Peng, W. Li, T. Yang, and K. Huafeng, “An internet of vehicles intrusion detection system based on a convolutional neural network,” *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1595–1599, 2019, doi:

10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00234.

[33] A. Javadpour, S. Kazemi Abharian, and G. Wang, “Feature selection and intrusion detection in cloud environment based on machine learning algorithms,” *Proc. - 15th IEEE Int. Symp. Parallel Distrib. Process. with Appl. 16th IEEE Int. Conf. Ubiquitous Comput. Commun. ISPA/IUCC 2017*, pp. 1417–1421, 2018, doi: 10.1109/ISPA/IUCC.2017.00215.

[34] M. Rakhimov, D. Mamadjanov, and A. Mukhiddinov, “A High-Performance Parallel Approach to Image Processing in Distributed Computing,” *14th IEEE Int. Conf. Appl. Inf. Commun. Technol. AICT 2020 - Proc.*, 2020, doi: 10.1109/AICT50176.2020.9368840.

[35] C. M. Rao and K. Shyamala, “Analysis and implementation of a parallel computing cluster for solving computational problems in data analytics,” *Proc. 2020 Int. Conf. Comput. Commun. Secur. ICCCS 2020*, 2020, doi: 10.1109/ICCCS49678.2020.9277362.

[36] M. Arai, F. Akagi, S. Yamaguchi, and K. Yoshida, “Improving performance of transposition algorithm of 3-D data array for parallelization using message passing interface,” *Proc. - 2018 6th Int. Symp. Comput. Netw. Work. CANDARW 2018*, no. 1, pp. 486–490, 2018, doi: 10.1109/CANDARW.2018.00094.

[37] X. Wang *et al.*, “A New Parallel Scheduling Algorithm Based on MPI,” *2018 15th Int. Comput. Conf. Wavelet Act. Media Technol. Inf. Process. ICCWAMTIP 2018*, pp. 228–231, 2018, doi: 10.1109/ICCWAMTIP.2018.8632603.

[38] B. Klenk, H. Froening, H. Eberle, and L. Dennison, “Relaxations for High-Performance Message Passing on Massively Parallel SIMT Processors,” *Proc. - 2017 IEEE 31st Int. Parallel Distrib. Process. Symp. IPDPS 2017*, pp. 855–865, 2017, doi: 10.1109/IPDPS.2017.94.

[39] J. Seiler and A. Kaup, “Distributed parallel image signal extrapolation framework using message passing interface,” *Eur. Signal Process. Conf.*, vol. 2016-November, pp. 81–85, 2016, doi:

10.1109/EUSIPCO.2016.7760214.

[40] V. Sinha and T. Raj, “Implementation of message passing interface in MANETs in processing of big data,” *Proc. - 2015 Int. Conf. Comput. Commun. Syst. ICCCS 2015*, pp. 113–117, 2016, doi: 10.1109/CCOMS.2015.7562883.

[41] S. Karim, Z. Omar, and H. Ibrahim, “Efficient parallel algorithm for listing permutation with Message Passing Interface (MPI),” *2015 Int. Symp. Math. Sci. Comput. Res. iSMSC 2015 - Proc.*, pp. 343–348, 2016, doi: 10.1109/ISMSC.2015.7594077.

[42] D. Ibanez, I. Dunn, and M. S. Shephard, “Hybrid MPI-thread parallelization of adaptive mesh operations,” *Parallel Comput.*, vol. 52, pp. 133–143, 2016, doi: 10.1016/j.parco.2016.01.003.

[43] O. Vega-Gisbert, J. E. Roman, and J. M. Squyres, “Design and implementation of Java bindings in Open MPI,” *Parallel Comput.*, vol. 59, pp. 1–20, 2016, doi: 10.1016/j.parco.2016.08.004.

[44] H. Fujita, C. Cao, S. Sur, C. Archer, E. Paulson, and M. Garzaran, “Efficient implementation of MPI-3 RMA over openFabrics interfaces,” *Parallel Comput.*, vol. 87, pp. 1–10, 2019, doi: 10.1016/j.parco.2019.04.008.

[45] R. Titos-Gil, O. Palomar, O. Unsal, and A. Cristal, “Architectural support for efficient message passing on shared memory multi-cores,” *J. Parallel Distrib. Comput.*, vol. 95, pp. 92–106, 2016, doi: 10.1016/j.jpdc.2016.02.005.

[46] D. Jaeger, A. Sapegin, M. Ussath, F. Cheng, and C. Meinel, “Parallel and distributed normalization of security events for instant attack analysis,” *2015 IEEE 34th Int. Perform. Comput. Commun. Conf. IPCCC 2015*, 2016, doi: 10.1109/PCCC.2015.7410270.

[47] X. Zhou, R. Li, S. Li, Y. Liu, and Y. Dou, “An efficient parallel successive cancellation list polar decoder based on gpus,” *Proc. - 2019 IEEE Intl Conf*

Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019, pp. 1378–1385, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00198.

[48] P. Harikrishna and A. Amuthan, "Rival-Model Penalized Self-Organizing Map enforced DDoS attack prevention mechanism for software defined network-based cloud computing environment," *J. Parallel Distrib. Comput.*, vol. 154, pp. 142–152, 2021, doi: 10.1016/j.jpdc.2021.03.005.

[49] N. T. Karonis, B. Toonen, and I. Foster, "MPICH-G2: A grid-enabled implementation of the Message Passing Interface," *J. Parallel Distrib. Comput.*, vol. 63, no. 5, pp. 551–563, 2003, doi: 10.1016/S0743-7315(03)00002-9.

[50] R. Thakur and W. Gropp, "Test suite for evaluating performance of multithreaded MPI communication," *Parallel Comput.*, vol. 35, no. 12, pp. 608–617, 2009, doi: 10.1016/j.parco.2008.12.013.

[51] T. Newhall, K. C. Webb, V. Chaganti, and A. Danner, "An introductory-level undergraduate CS course that introduces parallel computing," *J. Parallel Distrib. Comput.*, vol. 199, no. January, p. 105044, 2025, doi: 10.1016/j.jpdc.2025.105044.