Jurnal Elektronik Ilmu Komputer Udayana
Volume 12, No 3. Februari 2024

p-ISSN: 2301-5373
e-ISSN: 2654-5101

# Klasifikasi Serangan Application Layer Denial of Service Menggunakan Support Vector Machine (SVM) dan Chi Square

Putu Agus Prawira Dharma Yuda[a1], Cokorda Pramartha[ab2], I Komang Ari Mogi[a3]

[a]Informatics Engineering, Faculty of Math and Science, University of Udayana
[b]Center for Interdisciplinary Research on the Humanities and Social Sciences, Udayana University
South Kuta, Badung, Bali, Indonesia
[1]agusprawira28@email.com
[2]cokorda@unud.ac.id
[3]arimogi@unud.ac.id

***Abstract***

*In an era marked by widespread computer usage, security emerges as a critical focal point demanding meticulous attention. The spectrum of potential threats encompasses various methods of attacking computer systems, with Denial of Service (DoS) attacks being a prominent concern. This study delves into the enhancement of cybersecurity by implementing a system capable of discerning between DoS attack data and normal data, employing the Support Vector Machine (SVM) algorithm. To optimize the efficacy of the classification system, a strategic feature selection process is imperative. This research advocates for the utilization of the Chi-square method for this purpose, aiming to eliminate irrelevant features and thereby enhance system performance. The Support Vector Machine algorithm, hinging on hyperplanes for classification, gains efficiency through judicious feature selection. The empirical findings of this research unveil that employing Chi-square feature selection significantly elevates the performance of the classification system when dealing with application layer attacks. Remarkably, this enhancement is achieved without compromising the accuracy of the system. Specifically, the classification of DoS application layer attacks using SVM in tandem with Chi-square yielded identical accuracy results compared to using SVM alone. The average accuracy reached an impressive 99.9995%, with a processing time of 6.08 minutes with chi-square selection feature. In contrast, the classification system without feature selection consumed a comparatively longer processing time of 6.85 minutes. This underscores the efficacy of Chi-square feature selection in optimizing the performance of cybersecurity systems, demonstrating a streamlined approach to safeguarding computer networks from malicious threats.*

***Keywords:*** *Denial of Service (DoS), Support Vector Machine (SVM), Chi-square, Classification, Feature Selection*

## 1. Pendahuluan

Di zaman sekarang, peran komputer tidak bisa terlepas dari kehidupan kita sehari-hari. Seperti contoh, kita menggunakan komputer untuk melakukan aktivitas seperti bertransaksi, membuat tugas, menjelajah internet, mengirim pesan, membuat konten, melestarikan budaya, dan masih banyak lagi [1, 2]. Dengan adanya komputer, kehidupan kita menjadi lebih mudah karena komputer mampu memproses tugas dengan cepat. Dengan begitu maka komputer memiliki peran yang sangat penting dalam kehidupan sehari-hari. Dengan banyaknya pengguna komputer maka keamanan menjadi aspek yang sangat perlu diperhatikan. Menurut Badan Siber dan Sandi Negara (BSSN) yang dilansir oleh situs berita Media Indonesia, selama periode Januari hingga Mei 2021 telah terjadi serangan siber di Indonesia sebanyak 448 juta kasus [3]. Ada berbagai cara untuk melakukan serangan ke komputer. salah satunya yaitu dengan melakukan serangan DoS (Denial of Service). DoS merupakan jenis serangan yang terjadi pada komputer maupun server pada jaringan internet yang dimana seorang intruder menyerang dengan menghabiskan resource dari komputer atau server hingga komputer atau server tersebut tidak dapat menjalankan fungsinya. Serangan tersebut sangat mempengaruhi kinerja dari layanan komputer atau server internet seperti contoh membuka webpage menjadi sangat lamban atau kinerja komputer menjadi sangat berat akibat dari serangan tersebut [4].

Untuk mengatasi masalah tersebut maka diperlukan sebuah sistem yang dapat mengklasifikasikan data serangan Denial of Service dan data normal. Dari sistem klasifikasi serangan *application layer DoS* yang dibangun bertujuan untuk mendeteksi adanya potensi serangan *application layer DoS* sehingga dapat memperkecil potensi kerusakan yang terjadi. Ada banyak metode yang dapat digunakan. Salah satunya yaitu dengan menggunakan metode Support Vector Machine. Support Vector Machine merupakan salah satu algoritma yang menggunakan hyperplane untuk melakukan klasifikasi. Dan untuk meningkatkan performa dari sistem klasifikasi maka perlu dilakukan seleksi pada fitur yang digunakan untuk menghilangkan fitur yang dianggap tidak berpengaruh pada sistem sehingga performa sistem dapat meningkat. Salah satu metode yang dapat digunakan untuk seleksi fitur yaitu metode Chi-square. Dengan menggunakan kedua metode tersebut, penulis akan melakukan klasifikasi serangan *application layer DoS* dengan menggunakan *dataset* yang telah ditentukan serta menganalisa tingkat efektifitas dari sistem yang dibangun melalui performa yang dihasilkan (akurasi, *precision*, *recall, f1-score,* dan waktu proses) dalam melakukan klasifikasi data serangan *application layer DoS*.

## 2. Metode Penelitian

### 2.1 Pengumpulan Data

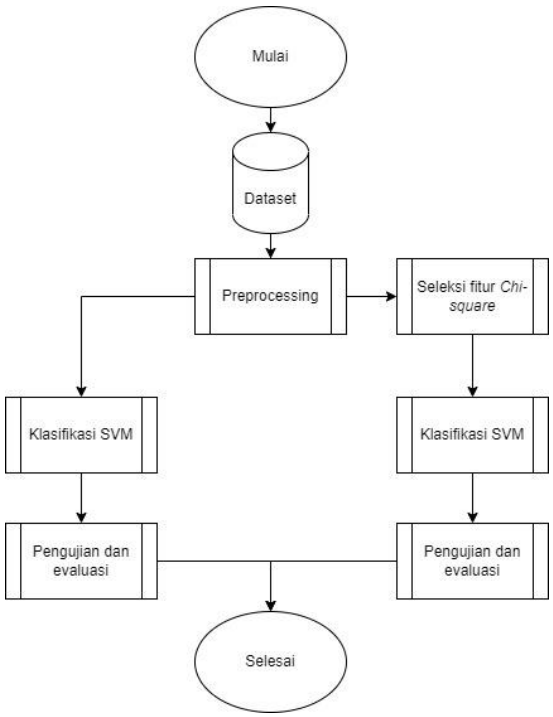
Dalam penelitian ini akan menggunakan dataset CSE-CIC-IDS2018 karena CSE-CIC-IDS2018 merupakan dataset serangan DoS paling terbaru sehingga hasil penelitian menjadi lebih relevan. Dataset CSE-CIC-IDS2018 memiliki jumlah fitur sebanyak 80 fitur yang dimana jumlah data serangan DoS yaitu sebanyak 601.802 data dari total keseluruhan data sebanyak 1.048.574. pada tabel di bawah ini merupakan rincian dari label data pada dataset.

*Tabel 1. Rincian Data Target*

| | **Normal/Benign** | **DoS** | |
|---|---|---|---|
| | | **Hulk** | **SlowHTTP** |
| | 446772 | 461912 | 139890 |
| **Total** | 446772 | 601802 | |

### 2.2 Alur Penelitian

Alur dari penelitian dari masing-masing seleksi fitur yang akan digunakan yang digambarkan melalui flowchart berikut.




*Gambar 1. Alur Penelitian*

p-ISSN: 2301-5373
e-ISSN: 2654-5101

Tahap pertama pada yang akan dilakukan pada penelitian ini adalah mengumpulkan data yang digunakan untuk penelitian. Pada penelitian ini menggunakan data dari CSE-CIC-IDS2018 yang kemudian dikonversi ke format *.csv. Setelah mengumpulkan data kemudian dilanjutkan dengan tahap preprocessing yang bertujuan untuk menghilangkan data yang tidak relevan dan menyetarakan format data pada masing-masing fitur sehingga mudah diproses oleh program. Selanjutnya data secara terpisah dilakukan seleksi fitur dengan metode Chi-square yang nanti akan dipilih fitur yang memiliki nilai $\alpha < 0{,}05$. Kemudian dilanjutkan dengan proses klasifikasi menggunakan SVM pada masing-masing data yang sudah di seleksi fitur. Dan yang terakhir dilakukan proses pengujian beserta dengan evaluasi atas hasil pengujian yang didapatkan.
Untuk klasifikasi tanpa menggunakan seleksi fitur, alur penelitian hampir sama dengan alur penelitian dengan menggunakan seleksi fitur. Perbedaannya pada alur penilitian tersebut, setelah melakukan preprocessing langsung dilakukan proses klasifikasi.
Dari evaluasi yang dilakukan akan didapatkan output berupa tingkat performa dari hasil masing-masing metode yang digunakan yaitu tingkat akurasi dan waktu proses pada masing-masing metode.

### 2.3 Appilcation Layer Denial of Service

*Denial of Service (DoS)* merupakan sebuah tipe serangan pada komputer maupun server yang dimana penyerang menyiapkan komputer yang disebut dengan *botnet* yang dikendalikan oleh komputer server untuk membantu penyerangan terhadap komputer target yang menyebabkan komputer target menjadi lambat hingga sistem komputer menjadi *error* [5]. Ada berbagai tipe DoS salah satunya yaitu *Application Layer Denial of Service*. *Application Layer Denial of Service* adalah sebuah tipe serangan DoS yang mengeksploitasi semantik spesifik aplikasi dan pengetahuan domain untuk meluncurkan serangan DoS yang dimana sangat sulit bagi filter DoS untuk membedakan rangkaian permintaan serangan dengan rangkaian permintaan yang sah. Ada dua karakteristik yang membuat serangan *Application Layer Denial of Service* sangat merusak. Pertama, serangan *Application Layer Denial of Service* meniru permintaan dan karakteristik lalu lintas jaringan yang sama dengan permintaan klien yang sah, sehingga membuatnya sulit untuk dideteksi. Kedua, penyerang dapat menargetkan sumber daya yang lebih tinggi pada *server* seperti *socket, bandwidth disk, bandwidth database*, dan proses kerja [6].

### 2.4 Normalisasi Data

Normalisasi data adalah proses transformasi data yang dimana sebuah atribut diskalakan ke dalam rentang -1.0 hingga 1.0 atau dari 0.0 hingga 1.0. Normalisasi data memiliki berbagai jenis metode yang dapat digunakan, salah satunya dengan menggunakan *Min-max Normalization. Min-max Normalization* adalah metode normalisasi yang memetakan nilai v dari atribut A menjadi v' ke dalam rentang [$new\_min_A$, $new\_max_A$] berdasarkan rumus [7].

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A \quad (1)$$

### 2.5 Seleksi Fitur Chi-square

*Chi-square* adalah salah satu seleksi fitur yang bertipe *supervised* yang dimana dapat menghilangkan fitur-fitur yang tidak relevan tanpa mengurangi tingkat akurasi [8]. Chi-square dapat dihitung dengan menggunakan persamaan.

$$x^2 = \sum_{i=1}^{r} \frac{(o_i - e_i)^2}{e_i} \quad (2)$$

Dimana

$$e_i = \frac{\Sigma\, baris . \Sigma kolom}{n} \quad (3)$$

Keterangan:
a. $r$ = Jumlah baris
b. $c$ = jumlah kolom
c. $o_i$ = frekuensi observasi ke-i
d. $e_i$ = frekuensi harapan ke-i
e. $n$ = jumlah sampel

Adapun langkah-langkah dalam menggunakan metode *Chi-square* adalah sebagai berikut [9].
a. Merumuskan hipotesis H0 dan H1 yang dimana
    1. H0 = Kedua variabel tidak memiliki pengaruh yang signifikan
    2. H1 = Kedua variabel memiliki pengaruh yang signifikan
b. Tentukan nilai $\alpha$ (contoh $\alpha = 0.05$)
c. Hitung nilai df (*degree of freedom*) dengan persamaan.

$$df = (baris - 1)(kolom - 1) \quad (4)$$

d. Hitung nilai frekuensi harapan dengan persamaan (3)
e. Hitung nilai distribusi *chi-square* dengan persamaan (2)
f. Tentukan kriteria pengujian
   1. Jika nilai $x^2$ hitung ≤ $x^2$ tabel maka H0 diterima
   2. Jika nilai $x^2$ hitung > $x^2$ tabel maka H0 ditolak
   3. Jika nilai $\alpha \geq 0.05$ tabel maka H0 diterima
   4. Jika nilai $\alpha < 0.05$ tabel maka H0 ditolak

### 2.6 Support Vector Machine

*Support Vector Machine* (SVM) merupakan sebuah algoritma yang bertipe *supervised learning* yang digunakan untuk klasifikasi maupun regresi. SVM dikembangkan pertama kali oleh Boser, Guyon, dan Vapnik pada tahun 1992. Konsep SVM yaitu mencari *hyperplane* berukuran x-1 untuk memisahkan kedua buah kelas pada input space. Proses menemukan *hyperplane* dilakukan dengan memaksimalkan jarak antar kelas pada data yang digunakan [10]. SVM memiliki berbagai jenis kernel. Dalam penelitian ini kernel SVM yang digunakan yaitu SVM kernel linear. Adapun persamaan fungsi dari kernel SVM linear yaitu.

$$K\left(x_i,x_j\right) = x_i^T x_j \quad (5)$$

Adapun langkah-langkah dalam menggunakan metode SVM adalah sebagai berikut [11].

a. Lakukan *training* data hasil pembobotan dengan *sequential training* SVM.
b. Inisiasi parameter yang digunakan seperti $\lambda$, $\varepsilon$, $\gamma$, $\alpha$i, dan C.
c. Hitunglah matriks Hessian dengan persamaan
   $$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (6)$$
d. Lakukan iterasi perhitungan dari data ke-1 hingga data ke-n dengan persamaan berikut
   $$E_i = \sum_{j=1}^{n} a_j D_{ij} \quad (7)$$
   $$\delta ai = \min\{\max[\gamma(1 - E_i), -a_i], C - a_i\} \quad (8)$$
   $$\alpha i = \alpha_i + \delta\alpha_i \quad (9)$$
e. Dari perhitungan yang dilakukan sebelumnya, cari nilai $\alpha$i yang terbesar dan dilanjutkan dengan melakukan perhitungan untuk menentukan bias dengan persamaan *lagrange* berikut.
   $$b = -\frac{1}{2}[(\sum_{i=1}^{n} a_i y_i K(x_i, x^-)) + (\sum_{i=1}^{n} a_i y_i K(x_i, x^+))] \quad (10)$$
f. Untuk mengetahui hasil klasifikasi, dilakukan testing dengan melakukan perhitungan fungsi f(x) yang didapatkan dengan persamaan berikut.
   $$f(x) = \sum_{i=0}^{n} a_i y_i K(x_i, x^-) + b \quad (11)$$

Keterangan:

a. $\alpha$i = alfa, digunakan untuk mencari support vector
b. $\gamma$ = gamma, untuk mengatur kecepatan *learning rate*
c. $C$ = variabel slack
d. $\varepsilon$ = epsilon, untuk pencari nilai error
e. $D_{ij}$ = matriks *Hessian*
f. $x_i$ = data ke-i
g. $x_j$ = data ke-j
h. $y_i$ = kelas data ke i
i. $K(x_i x_j)$ = fungsi kernel

### 2.7 K-Fold Cross Validation

*K-Fold Cross Validation* merupakan metode yang digunakan untuk menguji tingkat keberhasilan suatu sistem dengan melakukan pengujian berulang dengan mengacak atribut masukan [12]. *K-Fold Cross Validation* membagi data menjadi subdata yang saling bebas secara acak. Pada himpunan data yang dibangun masing-masing data memiliki (k-1) *fold* data sebagai data latih dan 1 *fold* data sebagai data uji [13]. Berikut merupakan contoh dari pembagian dataset ke dalam proses *5-Fold Cross Validation.*

*Tabel 2. K-Fold Cross Validation*

| 100 data | | | | | |
|---|---|---|---|---|---|
| | A = 20 data | B = 20 data | C = 20 data | D = 20 data | E = 20 data |
| Percobaan 1 | A (Data testing) | B | C | D | E |
| Percobaan 2 | A | B (Data testing) | C | D | E |

p-ISSN: 2301-5373
e-ISSN: 2654-5101

| | | | | | |
|---|---|---|---|---|---|
| Percobaan 3 | A | B | C (Data testing) | D | E |
| Percobaan 4 | A | B | C | D (Data Testing) | E |
| Percobaan 5 | A | B | C | D | E (Data Testing) |

Adapun langkah-langkah *K-Fold Cross Validation* yaitu.

a. Data dibagi menjadi k bagian
b. Pada *fold* pertama, data bagian ke-1 menjadi data uji dan sisanya menjadi data latih yang kemudian menghitung akurasi dengan persamaan

$$Akurasi = \frac{\sum data\ uji\ yang\ benar\ diklasifikasi}{\sum total\ data} . 100\% \quad (12)$$

c. Lakukan hal yang serupa pada iterasi selanjutnya hingga mencapai fold ke-k. Lalu hitung rata-rata akurasi.

### 2.8 Confusion Matriks

*Confusion Matriks* merupakan metode yang digunakan untuk menguji tingkat akurasi sebuah sistem klasifikasi dengan menggunakan tabel matriks untuk menghitung tingkat akurasi [14].

*Tabel 3. Confusion Matriks*

| | | Sebenarnya | |
|---|---|---|---|
| | | Positif | Negatif |
| Prediksi | Positif | TP | FN |
| | Negatif | FP | TN |

Selain menghitung tingkat akurasi, *confusion matriks* juga dapat digunakan untuk menghitung nilai *precision, recall, dan F1-score*. Berikut merupakan rumus-rumus untuk menghitung akurasi, *precision, recall, dan F1-score*.

$$Akurasi = \frac{TP+TN}{\sum jumlah\ data} . 100\% \quad (13)$$

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Recall = \frac{TP}{TP+FN} \quad (15)$$

$$Fmeasure = \frac{2}{\frac{1}{precision}+\frac{1}{recall}} \quad (16)$$

## 3. Hasil dan Pembahasan

Berikut merupakan hasil klasifikasi serangan *application layer* menggunakan SVM baik dengan seleksi fitur *chi-square* maupun tanpa seleksi fitur.

*Tabel 4. Hasil Klasifikasi SVM Full Fitur dengan Gamma = 0.01*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **7.17** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 10.04 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 16.43 |

*Tabel 5. Hasil Klasifikasi SVM Full Fitur dengan Gamma = 0.1*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **6.89** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 9.46 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 16.6 |

*Tabel 6. Hasil Klasifikasi SVM Full Fitur dengan Gamma = 1*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **6.85** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 10.31 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 19 |

*Tabel 7. Hasil Klasifikasi SVM Chi-square dengan Gamma = 0.01*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **7.04** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 8.39 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 14.98 |

*Tabel 8. Hasil Klasifikasi SVM Chi-square dengan Gamma = 0.1*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **6.03** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 8.51 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 15.54 |

*Tabel 9. Hasil Klasifikasi SVM Chi-square dengan Gamma = 1*

| Lambda | Rata-rata | Precision | Recall | F1-scores | Waktu proses (menit) |
|---|---|---|---|---|---|
| **0.01** | **0.999995232** | **0.999991692** | **1** | **0.999995846** | **6.76** |
| 0.1 | 0.999992847 | 0.999987538 | 1 | 0.999993769 | 9.91 |
| 1 | 0.999985695 | 0.999975077 | 1 | 0.999987538 | 15.45 |

Dari hasil pengujian didapatkan hasil klasifikasi yang dimana tingkat akurasi yang dihasilkan selalu sama mengikuti parameter lambda yang digunakan baik perbedaan parameter gamma maupun perbedaan pada dataset yang digunakan. Sebagai contoh hasil akurasi untuk klasifikasi data serangan DDoS dengan SVM berparameter lambda 0.01 akan selalu menghasilkan tingkat akurasi 0.999995232 atau 99.9995% meskipun parameter gamma yang digunakan berbeda dan itu berlaku pada kedua jenis dataset baik dataset yang full fitur maupun dataset hasil seleksi chi-square.
Kemudian untuk pengaruh parameter terhadap performa sistem klasifikasi serangan DoS didapatkan kesimpulan bahwa parameter gamma mempengaruhi performa system dalam hal waktu proses yang dimana semakin besar nilai gamma memiliki kecenderungan semakin lama waktu proses yang dilakukan pada penggunaan parameter lambda yang sama. Sebagai contoh pada klasifikasi serangan DoS dengan seleksi fitur pada parameter $\lambda = 0.1$ didapatkan waktu proses yaitu pada $\gamma = 0.01$ memerlukan waktu 8.39 menit, $\gamma = 0.1$ memerlukan waktu 8.51 menit, dan $\gamma = 0.01$ memerlukan waktu 9.91 menit. Sedangkan parameter lambda mempengaruhi performa system dalam hal hasil klasifikasi

p-ISSN: 2301-5373
e-ISSN: 2654-5101

dan waktu proses yang dimana semakin besar lambda maka semakin kecil tingkat akurasi yang dihasilkan serta waktu proses yang diperlukan juga semakin melambat.
Untuk pemilihan model parameter yang akan digunakan untuk perbandingan klasifikasi dipilih berdasarkan hasil klasifikasi yang dihasilkan (nilai akurasi rata-rata, precision, recall, f1-scores) serta tingkat efisiensi pada waktu proses sistem. Pada klasifikasi SVM dengan dataset full fitur menggunakan model pengujian dengan parameter γ (gamma) sebesar 1 dan λ (lambda) sebesar 0.01. Hali itu dikarenakan model tersebut menghasilkan tingkat akurasi terbaik yaitu 99.9995% dengan waktu yang paling efisien yaitu 6.85 menit. Sementara itu Pada klasifikasi SVM dengan dataset hasil seleksi fitur menggunakan model pengujian dengan parameter γ (gamma) sebesar 0.1 dan λ (lambda) sebesar 0.01. Hali itu dikarenakan model tersebut menghasilkan tingkat akurasi terbaik yaitu 99.9995% dengan waktu yang paling efisien yaitu 6.03 menit. Berikut merupakan tabel dari hasil perbandingan klasifikasi SVM pada data serangan application layers DDoS baik menggunakan full fitur maupun hasil seleksi fitur menggunakan chi-square.

*Tabel 10. Perbandingan Hasil Klasifikasi Serangan DoS dengan full fitur dan chi-square*

| Tipe Klasifikasi | Rata-rata | Precision | Recall | F1-scores | Waktu (menit) |
|---|---|---|---|---|---|
| SVM Full Fitur | 0.999995232 | 0.999991692 | 1 | 0.999995846 | 6.85 |
| SVM Chi Square | 0.999995232 | 0.999991692 | 1 | 0.999995846 | 6.03 |

Dari tabel diatas, dapat disimpulkan bahwa klasifikasi SVM dengan seleksi fitur Chi-square mampu meningkatkan tingkat efisiensi waktu proses pada sistem klasifikasi data serangan application layers DoS tanpa mengurangi tingkat akurasi, precision, recall, dan f1-scores. Hal itu dapat dilihat dengan hasil klasifikasi yang sama yaitu nilai akurasi rata-rata sebesar 0.999995232 atau 99.9995%, precision sebesar 0.999991692, recall sebesar 1, serta nilai f1-scores sebesar 0.999995846. Untuk waktu proses, metode klasifikasi SVM dengan seleksi fitur chi-square menjalankan proses klasifikasi lebih cepat dari Klasifikasi SVM dengan full fitur yang dimana waktu proses SVM dengan seleksi fitur chi-square memerlukan waktu 6.03 menit sedangkan untuk SVM dengan full fitur menggunakan waktu 6.85 menit.

## 4. Kesimpulan

Dari hasil penelitian yang telah dilakukan dapat dimuat kesimpulan yaitu.

a. Perbandingan performa yang dihasilkan antara kedua tipe klasifikasi serangan *DoS* baik dengan SVM full fitur maupun SVM dengan seleksi *chi-square* pada pengaruh parameter memiliki kesamaan yaitu parameter gamma mempengaruhi performa system dalam hal waktu proses yang dimana semakin besar nilai *gamma* memiliki kecenderungan semakin lama waktu proses yang dilakukan pada penggunaan parameter *lambda* yang sama. Sedangkan parameter lambda mempengaruhi performa system dalam hal hasil klasifikasi dan waktu proses yang dimana semakin besar nilai lambda yang digunakan maka semakin kecil tingkat akurasi yang dihasilkan serta waktu proses yang diperlukan juga semakin melambat.
b. Penggunaan metode seleksi fitur chi-square pada klasifikasi serangan application layer DoS tidak mempengaruhi hasil klasifikasi pada sistem. Dimana hasil klasifikasi serangan application layer DoS menggunakan SVM dan chi-square memiliki hasil akurasi yang sama dengan hasil klasifikasi serangan application layer DoS dengan hanya menggunakan SVM yaitu nilai akurasi rata-rata sebesar 0.999995232 atau 99.9995%, precision sebesar 0.999991692, recall sebesar 1, serta nilai f1-scores sebesar 0.999995846.
c. Klasifikasi serangan application layer DoS menggunakan SVM dan chi-square dapat meningkatkan tingkat efisiensi pada waktu proses pada system yang dimana waktu proses SVM dengan seleksi fitur chi-square memerlukan waktu 6.03 menit sedangkan untuk SVM dengan full fitur menggunakan waktu 6.85 menit.

## Referensi

[1] C. Pramartha, I. Koten, I. G. N. A. C. Putra, I. W. Supriana, and I. W. Arka, “Pengembangan Sistem Dokumentasi Melalui Pendekatan Ontologi untuk Praktek Budaya Bali,” *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 3, pp. 259–268, Dec. 2022, doi: 10.23887/janapati.v11i3.53939.

[2] C. Pramartha, I. G. N. A. C. Putra, I. P. G. H. Suputra, and I. W. Arka, "Adopsi dan Pelatihan Penggunaan Perangkat Digital Papan Tombol Aksara Bali," *Jurnal Widya Laksmi*, vol. 3, no. 1, pp. 14–20, 2023.

[3] P. Ananda, "Serangan Siber di RI Terus Meningkat, Capai 448 Juta Kasus." Accessed: Dec. 16, 2021. [Online]. Available: https://mediaindonesia.com/politik-dan-hukum/414225/serangan-siber-di-ri-terus-meningkat-capai-448-juta-kasus

[4] Y. W. Pradipta, "Implementasi Intrusion Prevention System (Ips) Menggunakan Snort Dan Ip Tables Berbasis Linux," *Jurnal Manajemen Informatika*, vol. 7, no. 1, 2017.

[5] I. B. G. Amartya, I. M. Widiartha, I. G. A. G. A. Kadnyanan, I. G. N. A. C. Putra, I. P. G. H. Suputra, C. Pramartha, "Implementasi Algoritma Naive Bayes Classifier (NBC) Dan Information Gain Untuk Mendeteksi DDoS," *Jurnal Elektronik Ilmu Komputer Udayana*, vol. 11, no. 2, pp. 273–282, 2022, [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

[6] M. Srivatsa, A. Iyengar, J. Yin, and L. Liu, "Mitigating application-level denial of service attacks on Web servers: A client-transparent approach," *ACM Transactions on the Web*, vol. 2, no. 3, Jul. 2008, doi: 10.1145/1377488.1377489.

[7] H. Junaedi, H. Budianto, I. Maryati, dan Y. Melani, "Data Transformation pada Data Mining," *Prosiding Konferensi Nasional Inovasi dalam Desain dan Teknologi-IDeaTech*, vol. 7, pp. 93–99, 2011.

[8] L. Luthfiana, J. C. Young, dan A. Rusli, "Implementasi Algoritma Support Vector Machine dan Chi Square untuk Analisis Sentimen User Feedback Aplikasi," *Ultimatics : Jurnal Teknik Informatika*, vol. 12, no. 2, pp. 125–126, 2020, doi: 10.31937/ti.v12i2.1828.

[9] I. C. Negara and A. Prabowo, "Penggunaan Uji Chi–Square untuk Mengetahui Pengaruh Tingkat Pendidikan dan Umur terhadap Pengetahuan Penasun Mengenai HIV–AIDS di Provinsi DKI Jakarta," *Prosiding Seminar Nasional Matematika dan Terapannya 2018*, pp. 1–8, 2018.

[10] H. E. Wahanani, B. Nugroho, dan G. I. Prakoso, "Analisa Serangan Smurf Dan Ping of Death Dengan Metode Support Vector Machine ( Svm )," *Jurnal Teknologi Informasi dan Komunikas*, vol. 11, pp. 71–76, 2016.

[11] C. C. Aggarwal, "Mining Text Data," *Data Mining*, pp. 429–455, 2015, doi: 10.1007/978-3-319-14142-8_13.

[12] M. A. Banjarsari, H. I. Budiman, dan Farmadi, "Penerapan K-Optimal Pada Algoritma Knn Untuk Prediksi Kelulusan Tepat Waktu Mahasiswa Program Studi Ilmu Komputer Fmipa Unlam Berdasarkan Ip Sampai Dengan Semester 4," *Klik - Kumpulan Jurnal Ilmu Komputer*, vol. 2, no. 2, pp. 159–173, 2015.

[13] Suyanto, "Data Mining untuk Klasifikasi dan Klusterisasi Data," Informatika Bandung. Accessed: Dec. 10, 2021. [Online]. Available: https://scholar.google.co.id/scholar?hl=id&as_sdt=0%2C5&q=data+mining+untuk+klasifikasi+dan+klusterisasi+data+suyanto&btnG=

[14] A. Indriani, "Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier," *Seminar Nasional Aplikasi Teknologi Informasi (SNATI) Yogyakarta*, pp. 5–10, 2014, [Online]. Available: www.bluefame.com,