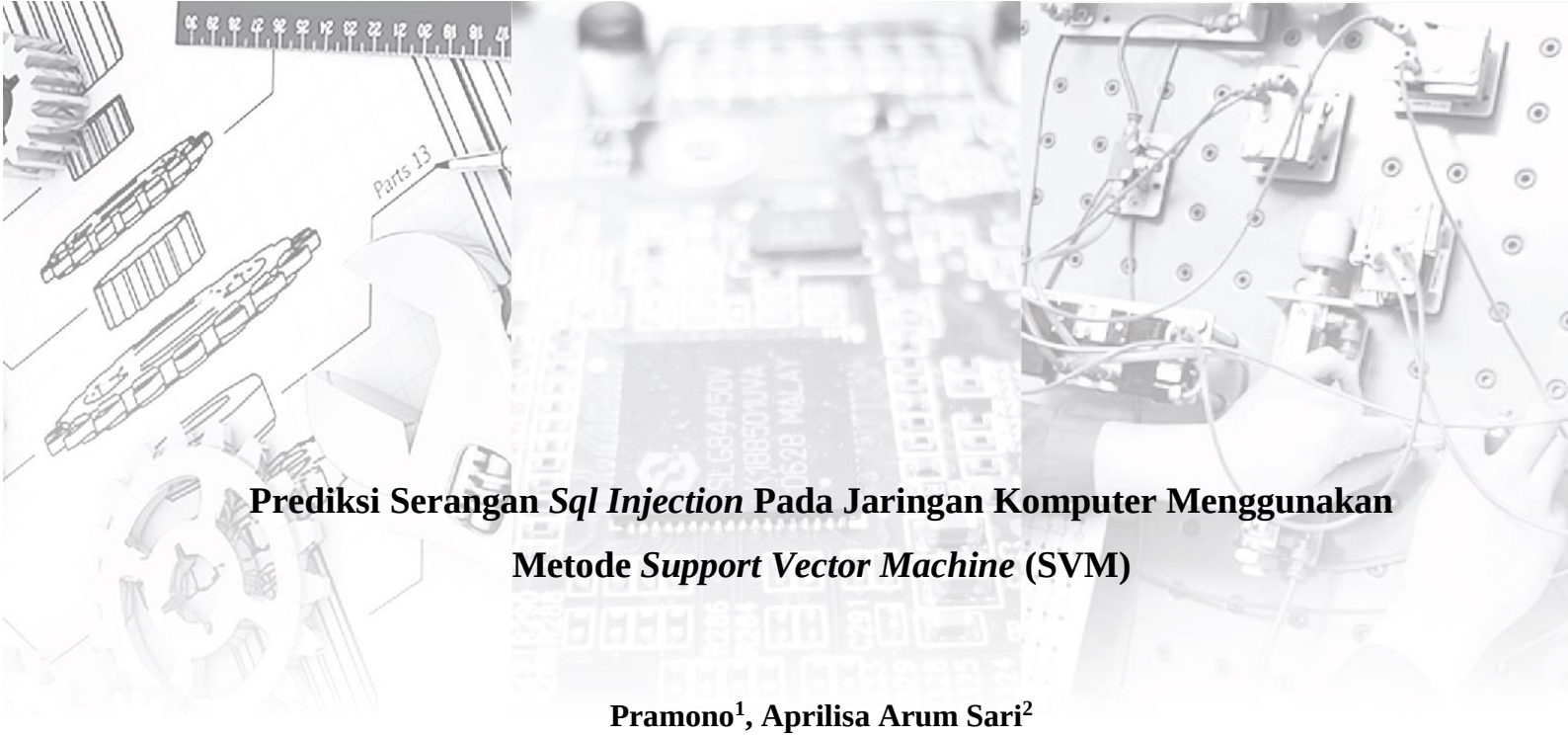


Prediksi Serangan *Sql Injection* Pada Jaringan Komputer Menggunakan Metode *Support Vector Machine* (SVM)

Pramono[1], Aprilisa Arum Sari[2]
Program Studi Teknik Informatika, Universitas Duta Bangsa
pramono@udb.ac.id[1], aprilisa_arumsari@udb.ac.id[2]

Abstrak

Serangan *sql injection* merupakan ancaman serius bagi keamanan jaringan komputer dan integritas data. Dalam upaya untuk mengatasi ancaman ini, penelitian telah dilakukan untuk mengembangkan metode deteksi yang efektif. Salah satu pendekatan yang menjanjikan adalah menggunakan metode *Support Vector Machine* (SVM) dalam *Machine Learning*. Dalam penelitian ini, kami memperkenalkan pendekatan prediksi serangan *SQL injection* pada jaringan komputer menggunakan SVM. Langkah-langkah prediksi meliputi, pengelompokan data, pelatihan model SVM, validasi, pengujian, dan evaluasi kinerja model. Diharapkan penelitian ini dapat memberikan kontribusi dalam pengembangan sistem deteksi yang dapat melindungi sistem komputer dari ancaman *sql injection*. Dataset yang akan digunakan dalam penelitian ini berasal dari sebuah website bernama *Kaggle*. Penelitian ini menganalisis metode yang dihasilkan dari proses klasifikasi berdasarkan Sekenario percobaan menghasilkan nilai akurasi *confusion matrix, precision, recall*, dan menghasilkan tingkat akurasi 96, 8412% pada sekenario kedua.

Kata Kunci: *Sql Injection; Machine Learning; Support Vector Machine*

ABSTRACT

Sql injection attacks pose a serious threat to the security of computer networks and data integrity. In an effort to address this threat, research has been conducted to develop effective detection methods. One promising approach is to use Support Vector Machine (SVM) methods in Machine Learning. In this study, we introduce a predictive approach to detecting sql injection attacks on computer networks using SVM. The prediction steps include data clustering, SVM model training, validation, testing, and model performance evaluation. It is hoped that this research will contribute to the development of detection systems that can protect computer systems from sql injection threats. The dataset used in this study is sourced from a website called Kaggle. This research analyzes the method resulting from the classification process based on experimental scenarios, producing accuracy values for confusion matrix, precision, recall, and producing an accuracy level of 96.8412% in the second scenario.

Keywords: *Sql Injection; Machine Learning; Support Vector Machine*

A. PENDAHULUAN

Di era digital yang terus berkembang, jaringan komputer memainkan peran sentral dalam mendukung berbagai aktivitas, mulai dari komunikasi bisnis hingga penyimpanan dan

pertukaran data sensitif. Namun seiring kemajuan teknologi, keamanan jaringan komputer menjadi semakin penting. Ancaman terhadap keamanan jaringan menjadi semakin kompleks, dan serangan siber berkembang dalam teknologi dan kecanggihan. Salah satu serangan yang paling umum dan berbahaya adalah serangan *sql injection* (Tahir, 2023).

Serangan *sql injection* adalah teknik yang digunakan oleh penyerang untuk mengeksploitasi kerentanan dalam aplikasi web. Dengan mengeksploitasi kerentanan dalam aplikasi, penyerang dapat memasukkan perintah sql berbahaya ke dalam kueri yang dijalankan oleh database. Dampak serangan ini bisa sangat buruk, mulai dari mencuri data sensitif hingga membahayakan integritas seluruh sistem. Ketika kita semakin bergantung pada sistem komputer, kebutuhan akan perlindungan yang kuat terhadap serangan *SQL injection* menjadi semakin mendesak (Perdana Putranto et al., 2022).

Mendeteksi serangan *sql injection* sejak dini adalah kunci untuk memitigasi kerugian yang terjadi. Namun serangan ini sulit dideteksi karena penyerang sering kali menyamarkan serangannya dengan baik. Dalam mengatasi tantangan ini, teknik pembelajaran mesin telah terbukti menjadi alat yang efektif untuk mendeteksi serangan siber seperti serangan injeksi sql. *Machine learning* memungkinkan komputer belajar dari data historis dan mengidentifikasi pola mencurigakan yang dapat digunakan untuk membedakan serangan dari aktivitas normal di lalu lintas jaringan (Triloka et al., 2022).

Pendekatan yang menjanjikan untuk mendeteksi serangan *SQL injection* adalah dengan menggunakan metode *support vector machine* (SVM). SVM adalah algoritma pembelajaran mesin yang kuat dan serbaguna yang dapat digunakan untuk klasifikasi dan regresi. SVM telah terbukti efektif dalam mendeteksi serangan siber seperti serangan injeksi sql karena kemampuannya memproses kumpulan data yang kompleks dan menghasilkan batasan keputusan yang kuat. Dengan menggunakan SVM, organisasi dapat mengembangkan sistem deteksi yang mendeteksi serangan *sql injection* dengan akurasi tinggi, memungkinkan mereka merespons ancaman yang muncul dengan cepat dan efektif (Geiß et al., 2023).

Tujuan dari penelitian ini adalah mengembangkan metode untuk memprediksi serangan *sql injection* pada jaringan komputer dengan menggunakan teknik *support vector machine* (SVM). Tujuan dari penelitian ini adalah untuk mengumpulkan data lalu lintas jaringan yang representatif, mengembangkan metode pemrosesan data yang efisien, melatih model SVM, dan meningkatkan keandalan sistem deteksi yang dikembangkan dalam mencegah serangan injeksi sql terhadap jaringan komputer. Dengan mencapai tujuan tersebut,

penelitian ini diharapkan dapat memberikan kontribusi yang berharga bagi pengembangan sistem keamanan jaringan komputer yang lebih kuat dan andal terhadap ancaman *sql injection* yang semakin kompleks dan destruktif. Penelitian ini bertujuan untuk menemukan solusi efektif untuk melindungi jaringan komputer dari serangan *sql injection* dan meminimalkan risiko terkait terhadap organisasi dan individu.

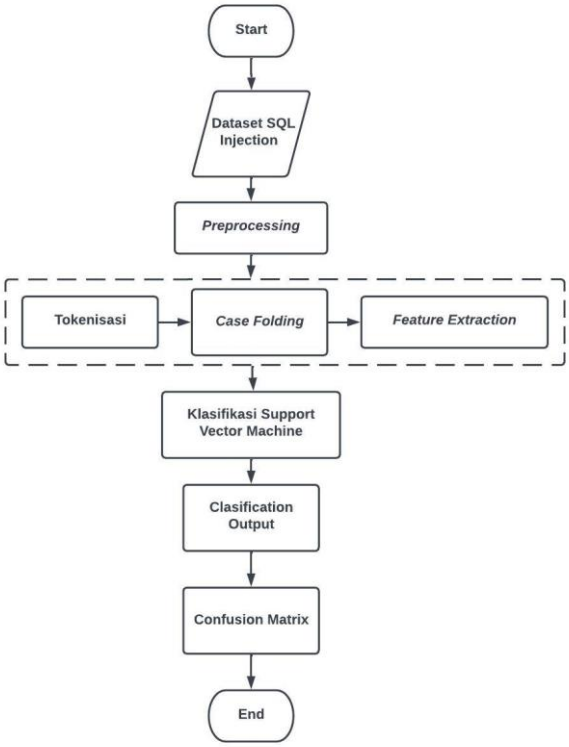
B. METODE

Pendekatan yang dilakukan dalam penelitian ini adalah kuantitatif dan berdasarkan jenis penelitian eksperimental yang menjalankan skenario pengujian menggunakan algoritma *suport vector machine* untuk mendeteksi serangan *sql injection* baik berupa serangan maupun non-serangan. Kemudian mengevaluasi skenario eksperimental untuk menilai keakuratan deteksi dan klasifikasi serangan dan non-serangan dari setiap skenario dan menetapkan fakta dari hasil penelitian penulis. Dalam hal ini penelitian yang dilakukan penulis bersifat deskriptif, data dijelaskan dengan gambar dan tabel.

Model ini menunjukkan alur bahwa suatu dataset yang dikumpulkan di website kaggel harus melalui beberapa tahapan. Yaitu mengelompokkan data melalui *preprocessing, tokenization, feature extraction* (Cahyani & Saraswati, 2023). Proses klasifikasi kemudian dilakukan pada setiap logaritma (dalam hal ini *support vector machine*). Model klasifikasi kemudian diuji dan menghasilkan keluaran *output confusion matrix*. Dari metode klasifikasi ini dibandingkan nilai *accuracy, precision,*dan *recall*. Hasil pengujian proses klasifikasi dataset menghasilkan keluaran berupa *confusion matrix* (Valero-Carreras et al., 2023). Gambar desain model ditunjukkan pada gambar 1.

1. Dataset

Pada penelitian ini dataset yang akan gunakan berasal dari website bernama kaggle yang merupakan situs dan platform untuk menganalisa dan memprediksi suatu dataset dan dilakukan pengunduhan secara manual melalui link (https://www.kaggle.com/datasets/syedsaqlainhussain/sql-injection-dataset). Dataset terdiri dari kalimat teks biasa yang merupakan Non serangan atau disebut sql yang berisi 3060 baris. Kemudian yang kedua yaitu dataset sqli atau dataset yang merupakan serangan dataset ini berisi sekitar 1127 baris. Dataset disimpan di dalam *google drive*, karena proses klasifikasinya menggunakan *google colab*. Contoh dataset dapat di lihat pada gambar 2.



Gambar 1 Perancangan Model

992	bfilename	1
993	having 1 = 1--	1
994	) or benchmark (10000000,MD5 (1)) #	1
995	or username like char (37) ;	1
996	;waitfor delay '0:0:__TIME__'--	1
997	" or 1 = 1--	1
998	x' AND userid IS NULL; --	1

Gambar 2. Dataset Dari Kaggle

2. *Preprocessing*

Pada tahap ini, proses diawali dengan melakukan load data dari *google drive* ke dalam *google colab* (Maulana et al., 2023). Tahap ini memiliki tujuan untuk mengolah data yang belum siap menjadi data yang suda siap untuk masuk kedalam pembentukan model. Kemudian proses *preprocessing* pada penelitian ini memiliki tiga proses diantaranya *case folding, stopwords removal*, dan *tokenisasi.*

Case folding adalah proses untuk menyetarakan seluruh kata menjadi sama atau setiap huruf besar atau *uppercase* menjadi huruf kecil atau *lowercase* (Liang, 2021). Hasil dari *case folding* dapat dilihat pada tabel 1.

Stopword removal adalah proses menyingkirkan sebuah kata dan yang dianggap tidak memiliki keterkaitan pada suatu kalimat atau kurang penting (Liang, 2021). Hasil dari *stopword removal* dapat di lihat pada tabel 2.

Tabel 1. ***Case Folding***

Sebelum	Sesudah
org/?option = com_k2 <a href = "http://corfopym	org/?option = com_k2 <a href = "http://corfopym
com/?option = com_k2 <act> <![CDATA[procMemb...	com/?option = com_k2 <act> <![cdata[procmemb...

Tabel 2. ***Stopword Removal***

Sebelum	Sesudah
org/?option = com_k2 <a href = "	org/?option = com_k <a href = "
com/?option = com_k2 <act> <![cdata[procmember...	com/?option = com_k <act> <![cdata[procmemberi...

Tokenizing atau bisa juga disebut *parsing*. *Tokenizing* adalah proses pemotongan dokumen menjadi bagian-bagian kata yang disebut *token*. Spasi digunakan untuk memisahkan antar kata tersebut. Sedangkan kata-kata yang tidak dibutuhkan akan dihilangkan melalui proses *filtering* dari hasil *tokenizing* (Liang, 2021). Hasil dari *Tokenizing* dapat dilihat pada tabel 3.

Tabel 3. ***Tokenizing***

Sebelum	Sesudah
orgoption comk href	[orgoption, comk, href]
comoption comk act cdataprocmemberinsert act a...	[comoption, comk, act, cdataprocmemberinsert, ...

3. *Feature Extraction*

Dalam hal ini *feature extraction* membantu dalam mendeteksi karekteristik dalam query pada dataset dan menjadikan sebagai fitur. Dalam penggunaan *machine learning*, *feature extraction* dianggap penting karena fitur digunakan untuk mencapai performa yang baik (Hakim, 2021). Pada penelitian ini dilakukan *feature extraction* menggunakan TF-IDF (Hibattullah et al., n.d.). Hasil dari TF IDF dapat dilihat pada tabel 4,kemudian data dan label disimpan kedalam variabel dalam bentuk *array* (Astiko & Achmad Khodar, 2020).Hasil dapat di lihat pada tabel 5.

Tabel 4. Hasil Proses TF-IDF

	term	TF	TF-IDF
	and	0.11764705882352941	0.4208587340401235
	utlinaddrgethostaddress	0.058823529411764705	0.2567042634065776
	select	0.11764705882352941	0.2300516243421638
	distinct	0.11764705882352941	0.5205408352621474
	tablename	0.11764705882352941	0.6938709220985616
	from	0.11764705882352941	0.23978006150041242
	rownum	0.058823529411764705	0.23560815702869137
	as	0.058823529411764705	0.23022185152719193
	limit	0.11764705882352941	0.5307774678491628
	sysalltables	0.058823529411764705	0.3525419422142411
	where	0.058823529411764705	0.12277694614938328

Tabel 5. Variabel Array

Variabel *x*
array([[1.91907116, 0. , 0. , ..., 0. , 0. , 0.], [0. , 0. , 0. , ..., 0. , 0. , 0.], [1.91907116, 0. , 0. , ..., 0. , 0. , 0.], ..., [0. , 0. , 0. , ..., 0. , 0. , 0.], [0. , 0. , 0. , ..., 0. , 0. , 0.], [0. , 0. , 0. , ..., 0. , 0. , 0.]])
Variable *y*
188 1 634 1 3990 0 2040 0 439 1 .. 3798 0 1347 0 482 1 1248 0 4180 0 Name: Label, Length: 2004, dtype: int64

C. HASIL DAN PEMBAHASAN

Untuk mendapatkan model klasifikasi yang diharapkan, maka skenario percobaan yang dibuat adalah dengan membandingkan data latih dan data uji perbandingan yang digunakan adalah 50% data latih: 50% data uji, 70% data latih: 30% data uji, 90% data latih*:* 10% data uji, hal ini bertujuan untuk mengetahui akurasi, jika data uji lebih besar dibanding dengan data latih maka yang terjadi nilai akurasi justru semakin kecil. Sebaliknya semakin besar data latih dibanding dengan data uji maka prosentase hasil yang didapat pada akurasi semakin tinggi.

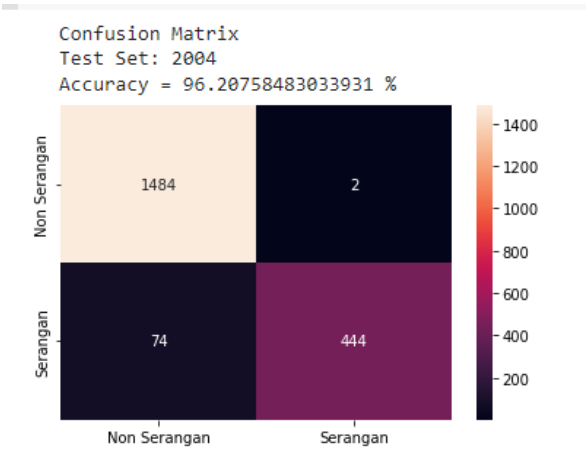
Tabel 6. Skenario Percobaan

No	Skenario	Jumlah Data training	Jumlah Data test	Akurasi	Precision	Recall
1.	S1	2003	2004	0,9575	0,97130	0,91811
2.	S2	2804	1203	0,9576	0,97139	0,92057
3.	S3	3606	401	0,9625	0,97572	0,92990

Pada tabel 6 memperlihatkan S1 dengan data latih lebih kecil dengan data uji nilai akurasi hanya 0,9575, sedangkan pada S2 dengan data uji yang lebih kecil nilai akurasinya jauh lebih besar yaitu dengan nilai 0,9576. Maka selanjutnya dilakukan proses skenario ke

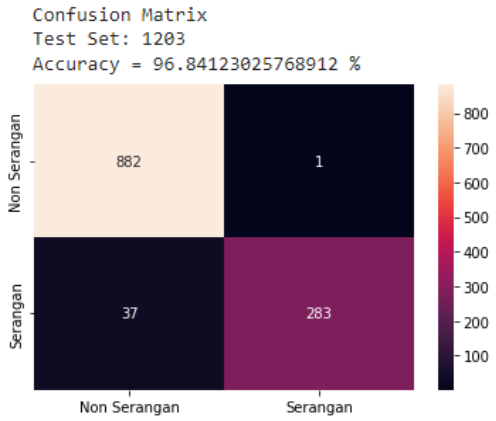
tiga untuk lebih melihat sejauh mana jika data uji semakin kecil seperti yang terlihat pada S3 dengan data uji lebih kecil hasil yang di dapat adalah 0,9625 hanya terpaut beberapa angka di belakang koma namun tidak tidak mengurangi akurasi yang jauh lebih tingi jika menggunakan data uji lebih kecil.

Pada gambar 3 menunjukkan hasil dari pengujian 2004 data uji terhadap 2003 data latih. Berdasarkan hasil pada gambar 3, tingkat akurasi pada pengujian dengan menggunakan 2004 data uji sebesar 96, 1077%.



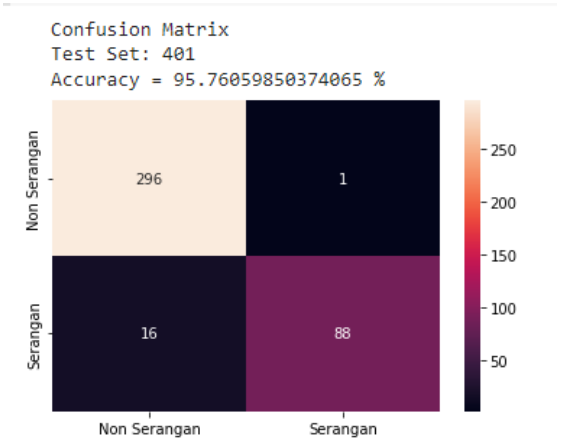
Gambar 3. Skenario Pengujian Satu

Pada gambar 4 menunjukkan hasil dari pengujian 1203 data uji terhadap 2804 data latih. Berdasarkan hasil pada gambar 4 tingkat akurasi pada pengujian dengan menggunakan 1203 Data uji sebesar 96, 8412%.



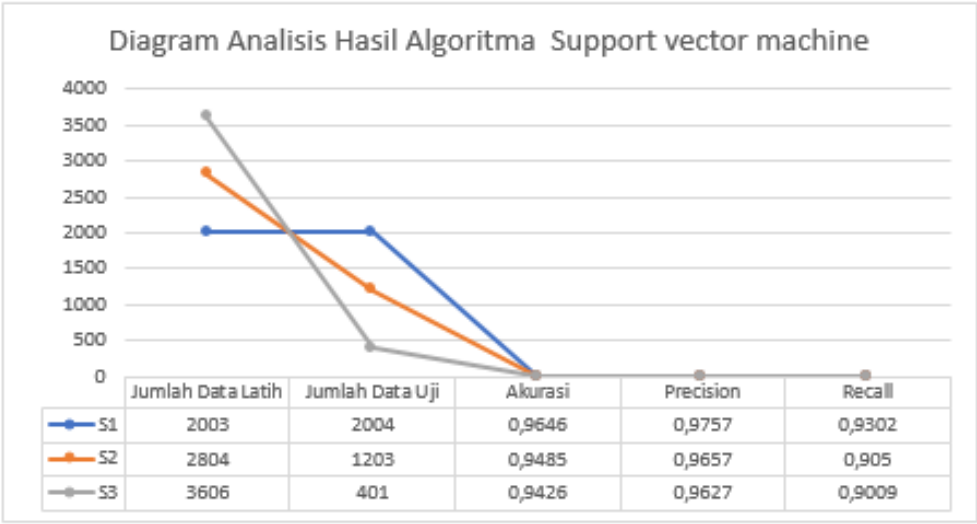
Gambar 4. Skenario Pengujian Kedua

Pada gambar 5 menunjukkan hasil dari pengujian 401 data uji terhadap 3606 data latih. Berdasarkan hasil pada gambar 5 tingkat akurasi pada pengujian dengan menggunakan 401 Data uji sebesar 95, 7605%.



Gambar 5. Skenario Pengujian Ketiga

Pada gambar 6 memperlihatkan nilai persentasi dari *accuracy,preccission* dan *recall* antara ketiga skenario. Diketahui nilai *accuracy* dari algoritma *support vector machine* cukup besar dengan nilai 0,9646 pada skenario 1. *Preccission* nilai tertinggi terjadi pada skenario 1 dengan nilai 0,9757. Nilai *recall* pun menunjukan angka yang tertinggi pada skenario ke 1 dengan nilai 0,9302.



	Jumlah Data Latih	Jumlah Data Uji	Akurasi	Precision	Recall
S1	2003	2004	0,9646	0,9757	0,9302
S2	2804	1203	0,9485	0,9657	0,905
S3	3606	401	0,9426	0,9627	0,9009

Gambar 6: Diagram Analisis Algoritma SVM

D. PENUTUP

Simpulan dan Saran

Berdasarkan hasil pengujian yang sudah dilakukan dalam penelitian ini, maka bisa disimpulkan bahwa motode *support vector machine* mampu diterapkan sebagai model dalam memprediksi serangan pada *sql injection* dengan skenario ke 1 dari jumlah data uji 2004 dan menghasilkan nilai akurasi 96, 1077%. Kemudian dapat di simpulkan bahwa metode *machine learning* dengan menggunakan model *support vector machine* dapat digunakan untuk memprediksi serangan *sql injection* pada jaringan komputer. Penelitian ini hanya terbatas pada dataset yang di unduh di web kaggel sebaiknya pada penelitian selanjutnya dapat menggunakan dataset lain untuk mendapatkan hasil yang berbeda. Kemudian teknik mengubah jumlah rasio data latih dan uji untuk menaikkan akurasi tidak bisa dilakukan sebagai bukti kenaikan akurasi. Sebaiknya yang harus dilakukan adalah merubah proses pada algoritmanya agar supaya akurasi yang dihasilkan pun bisa lebih maksimal.

DAFTAR PUSTAKA

Astiko, F., & Achmad Khodar. (2020). The Sentiment Analysis Reviewing Indosat Services from Twitter Using the Naive Bayes Classifier. *Journal of Applied Computer Science and Technology*, *1*(2), 61–66. https://doi.org/10.52158/jacost.v1i2.79

Cahyani, S. N., & Saraswati, G. W. (2023). Implementation Of Support Vector Machine Method In Classifying School Library Books With Combination Of TF-IDF And Word2vec. *Jurnal Teknik Informatika (Jutif)*, *4*(6), 1555–1566. https://doi.org/10.52436/1.jutif.2023.4.6.1536

Geiß, C., Aravena Pelizari, P., Tunçbilek, O., & Taubenböck, H. (2023). Semi-supervised learning with constrained virtual support vector machines for classification of remote sensing image data. *International Journal of Applied Earth Observation and Geoinformation*, *125*. https://doi.org/10.1016/j.jag.2023.103571

Hakim, B. (2021). Analisa Sentimen Data Text Preprocessing Pada Data Mining Dengan Menggunakan Machine Learning. *JBASE - Journal of Business and Audit Information Systems*, *4*(2). https://doi.org/10.30813/jbase.v4i2.3000

Hibattullah, N. M., Al Faraby, S., & Purbolaksono, M. D. (n.d.). *Analisis Sentimen terhadap Ulasan Film Berbahasa Inggris Menggunakan Metode Support Vector Machine dengan Feature Selection Information Gain*.

Liang, S. (2021). Comparative Analysis of SVM, XGBoost and Neural Network on Hate Speech Classification. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *5*(5), 896–903. https://doi.org/10.29207/resti.v5i5.3506

Maulana, M., Luthfi, A., & Wibowo, D. K. (2023). Network Attacks Classification for Network Forensics Investigation: Literature Reviews. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(5), 1132–1139. https://doi.org/10.29207/resti.v7i5.5153

Perdana Putranto, D., Hananto, B., Ilmu Komputer, F., Pembangunan Nasional Veteran Jakarta, U., Fatmawati Raya, J. R., & Labu, P. (2022). Analisis Keamanan Website Leads UPNVJ Terhadap Serangan SQL Injection & Sniffing Attack. *JURNAL INFORMATIK Edisi Ke 2022*, *18*, 2022.

Tahir, R. (2023). Transformasi Bisnis di Era Digital(Teknologi Informasi dalam Mendukung Transformasi Bisnisdi Era Digital). https://www.researchgate.net/publication/373161091

Triloka, J., Hartono, H., & Sutedi, S. (2022). Detection of SQL Injection Attack Using Machine Learning Based On Natural Language Processing. *International Journal of Artificial Intelligence Research*, *6*(2). https://doi.org/10.29099/ijair.v6i2.355

Valero-Carreras, D., Alcaraz, J., & Landete, M. (2023). Comparing two SVM models through different metrics based on the confusion matrix. *Computers and Operations Research*, *152*. https://doi.org/10.1016/j.cor.2022.106131