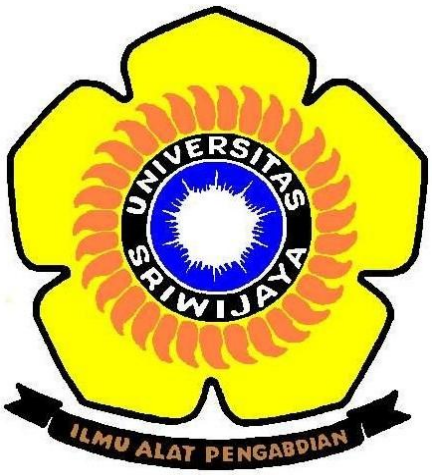
DETEKSI SERANGAN SQL INJECTION DENGAN APACHE SPARK MENGGUNAKAN METODE K-MEANS CLUSTERING

SKRIPSI



OLEH :

BIMA GUSTI SYAUQI

09011281823077

JURUSAN SISTEM KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS SRIWIJAYA

2024

LEMBAR PENGESAHAN

DETEKSI SERANGAN SQL INJECTION DENGAN APACHE SPARK MENGGUNAKAN METODE K-MEANS CLUSTERING

SKRIPSI

Program Studi Sistem Komputer
Jenjang S1

Oleh :

BIMA GUSTI SYAUQI
09011281823077

Indralaya, 26 Agustus 2024

Mengetahui

Ketua Jurusan Sistem Komputer	**Pembimbing Tugas Akhir,**
Dr. Ir. Sukemi, M.T.	**Ahmad Heryanto, S.Kom., M.T.**
NIP. 19661203200641001	**NIP. 198701222015041002**

HALAMAN PERSETUJUAN

Telah diuji dan lulus pada

Hari : Jum'at

Tanggal : 12 Juli 2024

Tim Penguji :

1. **Ketua : Dr. Ahmad Zarkasi, M.T.**
2. **Sekretaris : Muhammad Ali Buchari, M.T.**
3. **Penguji : Huda Ubaya, M.T.**
4. **Pembimbing I : Ahmad Heryanto, S.Kom., M.T.**

Mengetahui,
Ketua Jurusan Sistem Komputer

Dr. Ir. Sukemi, M.T.
NIP. 196612032006041001

HALAMAN PERNYATAAN

Yang bertanda tangan di bawah ini :

Nama : Bima Gusti Syauqi

NIM : 09011281823077

Judul : Deteksi Serangan SQL Injection Dengan Apache Spark Menggunakan Metode K-Means Clustering

Hasil Pengecekan Software *iThenticate/Turnitin* : 16%

Menyatakan bahwa laporan tugas akhir 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.

Indralaya, Agustus 2024

Bima Gusti Syauqi

NIM. 09011281823077

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh.

Puji dan syukur atas kehadiran Allah Subhanahu Wa ta'ala yang telah memberikan rahmat dan hidayah-Nya lah sehingga penulis dapat menyelesaikan penyusunan Tugas Akhir ini yang Berjudul "**Deteksi Serangan SQL Injection Dengan Apache Spark Menggunakan Metode K-means Clustering**".

Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak yang telah memberikan bantuan, dorongan, motivasi, semangat dan bimbingan dalam menyelesaikan penyusunan Tugas Akhir ini. Penulis mengucapkan terima kasih kepada :

1. Allah Subhanahu Wa ta'ala yang memberikan rahmat dan hidayah-Nya serta nikmat yang tak terhitung.
2. Kedua orangtua saya dan saudara yang telah membantu dan mendoakan.
3. Bapak Prof. Dr. Erwin, S.Si., M.Si. selaku Dekan Fakultas Ilmu Komputer Universitas Sriwijaya.
4. Bapak Dr. Ir. H. Sukemi, M.T. selaku Ketua Jurusan Sistem Komputer Fakultas Ilmu Komputer Universitas Sriwijaya.
5. Bapak Ahmad Heryanto, S.Kom., M.T. selaku Pembimbing Tugas Akhir yang telah berkenan meluangkan waktunya untuk membimbing. Memberikan saran dan motivasi serta bimbingan terbaik untuk penulisan dalam menyelesaikan Tugas Akhir ini.
6. Bapak Sutarno, S.T., M.T. selaku Dosen Pembimbing Akademik saya.
7. Mochammad Rafii Nanda Wicaksana, Dwi Lingga Hanayuda, Jepi Sujana, Daffa Bima Perdana dan Agung Al hafizin selaku rekan yang membantu menyusun dan menyelesaikan penulisan Tugas Akhir ini.
8. Kak Angga selaku admin Jurusan Sistem Komputer yang telah membantu mengurus seluruh berkas.
9. Teman-teman Sistem Komputer Angkatan 2018 Indralaya.

Dalam penyusunan Tugas Akhir ini penulis menyadari sepenuhnya masih jauh dari kata sempurna, oleh karena itu penulis mengharapkan saran dan kritik dari semua pihak yang berkenan agar menjadi bahan evaluasi yang lebih baik lagi.

Akhir kata saya harap semoga Laporan Tugas Akhir ini dapat bermanfaat serta dapat menambah pengetahuan dan wawasan bagi yang membutuhkannya.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Indralaya, Agustus 2024
Penulis

Bima Gusti Syauqi
09011281823077

DETEKSI SERANGAN SQL INJECTION DENGAN APACHE SPARK MENGGUNAKAN METODE K-MEANS CLUSTERING

Bima Gusti Syauqi (09011281823077)
Jurusan Sistem Komputer
Fakultas Ilmu Komputer, Universitas Sriwijaya
Palembang, Indonesia
Email : syauqi849@gmail.com

ABSTRAK

Structured Query Language Injection atau biasa disebut SQL Injection merupakan sebuah teknik hacking untuk mendapatkan akses pada sistem database yang berbasis SQL. Structured Query Language yaitu bahasa yang digunakan untuk membuat serta mengolah dan memanipulasi database. Dikarenakan hal tersebut, deteksi serangan Sql iniection menjadi yang pertama dan yang paling penting untuk melawan serangan Sql Injection. Dasar untuk melakukan pendekatan deteksi yaitu menggunakan machine learning. K-means clustering adalah algoritma analisis clustering yang paling sederhana dan paling terkenal dalam memecahkan masalah clustering. Algoritma ini dikenal efisien untuk dataset yang besar. Pada jurnal ini mengusulkan deteksi serangan Sql injection dengan menggunakan salah satu metode unsupervised learning yaitu K-means clustering pada apache spark. Penelitian ini menggunakan dataset CIC-IDS2018 dari University of New Brunswick (UNB) untuk melatih dan melakukan percobaan pada sistem deteksi yang digunakan.

Kata Kunci : SQL Injection, K-means, Apache Spark, Keamanan Jaringan, Machine Learning

SQL INJECTION ATTACK DETECTION WITH APACHE SPARK USING K-MEANS CLUSTERING METHOD

Bima Gusti Syauqi (09011281823077)
Department of Computer System
Faculty of Computer Science, University of Sriwijaya
Palembang, Indonesia
Email : syauqi849@gmail.com

ABSTRACT

Structured Query Language Injection or commonly called SQL Injection is a hacking technique to gain access to a SQL-based database system. Structured Query Language is a language used to create, process and manipulate databases. Because of this, detecting SQL injection attacks is the first and most important thing to do to combat SQL Injection attacks. The basis for conducting a detection approach is using machine learning. K-means clustering is the simplest and most well-known clustering analysis algorithm in solving clustering problems. This algorithm is known to be efficient for large datasets. This journal proposes the detection of SQL injection attacks using one of the unsupervised learning methods, namely K-means clustering on Apache Spark. This study uses the CIC-IDS2018 dataset from the University of New Brunswick (UNB) to train and experiment with the detection system used.

Keyword *: SQL Injection, K-means, Apache Spark, Network Security, Machine Learning*

DAFTAR ISI

HALAMAN JUDUL..... **i**
LEMBAR PENGESAHAN……. **ii**
HALAMAN PERSETUJUAN..... **iii**
HALAMAN PERNYATAAN..... **iv**
KATA PENGANTAR.....**v**
DAFTAR ISI.....**ix**
DAFTAR GAMBAR.....**xii**
DAFTAR TABEL..... **xiv**
BAB 1 PENDAHULUAN..... **1**
1.1 Latar Belakang..... 1
1.2 Perumusan Masalah……... 4
1.3 Tujuan…… 4
1.4 Manfaat..... 4
1.5 Batasan Masalah..... 5
1.6 Sistematika Penulisan..... 5
BAB II TINJAUAN PUSTAKA..... **7**
2.1 Pendahuluan.....7
2.2 Structured Query Language Injection.....10
2.2.1 In-Band SQL Injection.....12
2.2.2 Inferential SQL Injection..... 13
2.2.3 Out of Bond SQL..... 13
2.3 Hadoop Ecosystem.....14
2.3.1 Data Storage Layer.....15
2.3.2 Data Processing Layer..... 16
2.3.3 Data Access Layer..... 16
2.3.4 Data Management Layer.....18
2.4 Apache Spark..... 18

2.4.1 Komponen Spark..... 19
2.4.2 Resilient Distributed Datasets (RDDs)..... 21
2.4.3 Cara Kerja Spark.....22
2.5 Machine Learning..... 23
2.5.1 Supervised Learning..... 23
2.5.2 Unsupervised Learning..... 24
2.5.3 Reinforcement Learning.....25
2.6 K-Means Clustering.....26
2.7 *Confusion Matrix*..... 30
2.7.1 *Accuracy*..... 31
2.7.2 Recall..... 31
2.7.3 Spesifisitas..... 32
2.7.4 Presisi.....32
2.7.5 F1 Score..... 32
BAB III METODOLOGI PENELITIAN.....33
3.1 Pendahuluan.....33
3.2 Kerangka Kerja..... 33
3.3 Kerangka Kerja Metodologi Penelitian..... 34
3.4 Kebutuhan Perangkat.....35
3.5 Persiapan Dataset.....36
3.6 Ekstraksi Data.....38
3.7 Apache Spark.....40
3.8 Preprocessing Data.....41
3.8.1 Seleksi Fitur..... 42
3.8.2 Normalisasi Data.....43
3.9 K-Means Clustering.....44
3.10 Skenario Percobaan.....48
3.11 Validasi Hasil.....49
BAB IV HASIL DAN ANALISA.....50
4.1 Pendahuluan.....50
4.2 Hasil Ekstraksi Dataset..... 50

4.3 Hasil Normalisasi Data dan Seleksi Fitur..... 52
4.4 Hasil Pengujian K-Means Clustering.....52
4.5 Validasi Hasil.....53
4.5.1 Hasil Validasi Data Latih 80% dan Data Uji 20%.....53
4.5.2 Hasil Validasi Data Latih 50% dan Data Uji 50%.....54
4.5.3 Hasil Validasi Data Latih 30% dan Data Uji 70%.....55
4.6 Korelasi Hasil Deteksi Terhadap Label..... 55

4.6.1 Korelasi Hasil Deteksi Data Latih 30% dan Data Uji 70%.....55
4.6.2 Korelasi Hasil Deteksi Data Latih 50% dan Data Uji 50%.....57
4.6.3 Korelasi Hasil Deteksi Data Latih 80% dan Data Uji 20%.....58
4.7 Hasil Validasi Terhadap Apache Spark..... 59
4.7.1 Hasil Data Latih 30% dan Data Uji 70% Terhadap Apache Spark....59
4.7.2 Hasil Data Latih 50% dan Data Uji 50% Terhadap Apache Spark....60
4.7.3 Hasil Data Latih 80% dan Data Uji 20% Terhadap Apache Spark....61
4.8 Analisis Hasil Validasi.....62
4.9 Perbandingan Berdasarkan Penelitian Terkait..... 63
BAB V KESIMPULAN DAN SARAN.....65
5.1 Kesimpulan..... 65
5.2 Saran..... 65
DAFTAR PUSTAKA.....67

DAFTAR GAMBAR

Gambar 2.1 Skema Diagram Serangan SQL Injection..... 11
Gambar 2.2 Jenis-jenis SQL Injection..... 12
Gambar 2.3 Hadoop Ecosystem.....14
Gambar 2.4 Elemen Tiap Lapisan pada Hadoop Ecosystem..... 15
Gambar 2.5 Komponen Apache Spark..... 19
Gambar 2.6 Interaksi Spark dan Cluster Manager.....22
Gambar 2.7 Cluster Spark dengan Tiga Executor.....23
Gambar 2.8 Alur Kerja Supervised Learning..... 24
Gambar 2.9 Unsupervised Learning..... 25
Gambar 2.10 Algoritma Umum K-Means..... 27
Gambar 2.11 Elbow Method.....29
Gambar 3.1 Kerangka Kerja Penelitian..... 34
Gambar 3.2 Kerangka Kerja Metodologi Penelitian.....35
Gambar 3.3 Model pada Penelitian.....36
Gambar 3.4 Arsitektur Pada Jaringan Dataset CSE-CIC-IDS2018..... 38
Gambar 3.5 Apache Spark..... 40
Gambar 3.6 Flowchart Spark..... 41
Gambar 3.7 Flowchart Preprocessing Data.....41
Gambar 3.8 Jumlah Komponen PCA.....43
Gambar 3.9 Nilai Kontribusi Komponen..... 43
Gambar 3.10 Flowchart K-Means.....47
Gambar 3.11 Grafik Nilai Silhouette..... 48
Gambar 3.12 Flowchart Validasi Data.....49
Gambar 4.1 Data pcap.....50
Gambar 4.2 Hasil Ekstraksi Data..... 51
Gambar 4.3 Proses Ekstraksi Data..... 51

Gambar 4.4 Data Normal dan Data Serangan.....51
Gambar 4.5 Hasil Normalisasi Data..... 52
Gambar 4.6 Hasil Seleksi Fitur PCA..... 52
Gambar 4.7 Hasil Clustering.....53
Gambar 4.8 Hasil Confusion Matrix Data Latih 80% dan Data Uji 20%..... 53

Gambar 4.9 Hasil Confusion Matrix Data Latih 50% dan Data Uji 50%.....54
Gambar 4.10 Hasil Confusion Matrix Data Latih 30% dan Data Uji 70%.....55
Gambar 4.11 Korelasi Keseluruhan Data Latih 30% dan Data Uji 70%.....56
Gambar 4.12 Korelasi *False Positive* Data Latih 30% dan Data Uji 70%..... 56
Gambar 4.13 Korelasi *False Negative* Data Latih 30% dan Data Uji 70%.....56
Gambar 4.14 Korelasi Keseluruhan Data Latih 50% dan Data Uji 50%.....57
Gambar 4.15 Korelasi *False Positive* Data Latih 50% dan Data Uji 50%..... 57
Gambar 4.16 Korelasi *False Negative* Data Latih 50% dan Data Uji 50%.....57
Gambar 4.17 Korelasi Keseluruhan Data Latih 80% dan Data Uji 20%.....58
Gambar 4.18 Korelasi *False Positive* Data Latih 80% dan Data Uji 20%..... 58
Gambar 4.19 Korelasi *False Negative* Data Latih 80% dan Data Uji 20%.....58
Gambar 4.20 Performa Spark Pada Data Latih 30% dan Data Uji 70%..... 59
Gambar 4.21 Performa Spark Pada Data Latih 50% dan Data Uji 50%..... 60
Gambar 4.22 Performa Spark Pada Data Latih 80% dan Data Uji 20%..... 61

DAFTAR TABEL

Tabel 2.1 Penelitian Terkait Yang Dijadikan Landasan..... 7
Tabel 2.2 Matriks Konfusi.....31
Tabel 3.1 Spesifikasi Perangkat Keras..... 35
Tabel 3.2 Spesifikasi Perangkat Lunak..... 35
Tabel 3.3 Fitur Pada Dataset.....36
Tabel 3.4 Atribut Feature Extraction.....38
Tabel 3.5 Hasil Pengujian Berdasarkan Jumlah K Cluster.....48
Tabel 3.6 Jumlah Data yang Digunakan.....49
Tabel 3.7 Pembagian Data.....50
Tabel 4.1 Hasil Validasi Data Latih 80% dan Data Uji 20%..... 54
Tabel 4.2 Hasil Validasi Data Latih 50% dan Data Uji 50%..... 54
Tabel 4.3 Hasil Validasi Data Latih 30% dan Data Uji 70%..... 55
Tabel 4.4 Durasi Pemrosesan Data Latih 30% dan Data Uji 70%... 59
Tabel 4.5 Durasi Pemrosesan Data Latih 50% dan Data Uji 50%... 60
Tabel 4.6 Durasi Pemrosesan Data Latih 80% dan Data Uji 20%... 61
Tabel 4.7 Hasil Performa Validasi Keseluruhan..... 62
Tabel 4.8 Hasil Performa Spark..... 63
Tabel 4.9 Hasil Keseluruhan..... 63
Tabel 4.10 Perbandingan Penelitian Terkait.....64

BAB I
PENDAHULUAN

1.1 Latar Belakang

Structured Query Language Injection atau biasa disebut SQL Injection merupakan sebuah teknik *hacking* untuk mendapatkan akses pada sistem database yang berbasis SQL. Structured Query Language yaitu bahasa yang digunakan untuk membuat serta mengolah dan memanipulasi *database*[1]. Dalam proses melakukan teknik SQL injection, penyerang akan memanfaatkan celah keamanan pada web atau aplikasi. Penyerang akan memasukkan kode berbahaya ke dalam parameter permintaan, yang menyebabkan server mengeksekusi kueri ilegal, akibatnya kebocoran data dan kerusakan pada database. Seperti, penyerang dapat memperoleh nama dan kata sandi pengguna, selain itu data privasi pengguna pada situs web yang lainnya yang secara serius mengancam keamanan data. SQL injection ini dapat terjadi karena beberapa hal seperti kurangnya penanganan terhadap karakter-karakter seperti tanda petik satu atau karakter double minus yang dapat menyebabkan suatu aplikasi dapat disisipi peretas dengan perintah SQL[1].

SQL Injection sebagai salah satu metode serangan pertama kali yang dipublikasikan sebagai catatan tambahan untuk artikel eksploitasi layanan web Microsoft yang komprehensif. Artikel tersebut pertama kali muncul di artikel *Phrack* yang ke 54, sebuah majalah digital yang membahas topik peretasan. Artikel tersebut berjudul *"NT Web Technology Vulnerabilities"* artikel ini ditulis oleh Rainforest Puppy dari grup keamanan Wire Trip dan membahas tentang eksploitasi injeksi Microsoft SQL dan ASP[2]. Dalam beberapa tahun terakhir, banyak peneliti telah melakukan riset pada deteksi SQLi, tetapi cakupan deteksi biasanya terbatas pada beberapa sub set dari SQL Injection. Sangat diperlukan untuk menyediakan arsitektur deteksi SQLi yang komprehensif yang dapat mendeteksi semua jenis serangan SQLi dan memiliki fleksibilitas untuk

memperbarui jika jenis serangan baru telah terjadi[1].

Sedangkan menurut *open web application security project (OWASP) SQL injection* merupakan suatu teknik yang sering digunakan oleh attacker untuk menerobos ke suatu web secara ilegal. SQL injection digunakan oleh attacker untuk mengirimkan perintah - perintah SQL melalui URL yang nantinya dieksekusi oleh web server. Dari informasi tersebut injection termasuk dalam 10 risiko keamanan web paling kritis[1]. Dalam mengatasi serangan SQL injection, diketahui ada beberapa cara untuk mencegah serangan tersebut. Seperti yang telah dilakukan oleh peneliti sebelumnya guna mengatasi permasalahan serangan ini. Seperti halnya yang telah dilakukan oleh Xin Xie, dkk (2019), menggunakan metode CNN *(Convolutional Neural Network)*. Dijelaskan bahwa *Convolutional Neural Network (CNN)* ialah sejenis *deep feedforward neural network* yang mengikuti mekanisme penyusunan kognisi visual pada organisme. CNN mempunyai daya kerja yang benar-benar baik dalam visi komputer dan pemrosesan bunyi. Penelitian ini memperlihatkan cara deteksi injeksi SQL berdasarkan *Elastic-Pooling CNN (EP-CNN)* dan log website besar-besaran, dan CNN ditingkatkan dan dipakai pada deteksi injeksi SQL di aplikasi Laman. Hasil praktis memperlihatkan bahwa cara ini mempunyai efek yang baik dan kecermatan pengenalan yang tinggi.[3].

Pada penelitian[1] melakukan pendeteksian serangan *Sql Injection* dengan berbasis metode *Adaptive Deep Forest.* Dari penelitian ini dijelaskan bahwa mendeteksi serangan *sql injection* di ruang lingkup jaringan yang kompleks ialah tempat yang tepat di bidang keamanan jaringan. Sulit untuk memproses banyaknya fitur yang berlebihan menggunakan metode berbasis mesin klasik. Sedangkan metode berbasis *Deep Learning* memiliki parameter tinggi dan mudah menyebabkan *over-fitting*. Dengan menggunakan *Adaptive Deep Forest (ADF)*, parameter pada model dapat disesuaikan secara otomatis selama proses *training*, yang meningkatkan akurasi deteksi. Namun, penggunaan fitur tersebut dapat mempengaruhi proses training data.

Pada penelitian[3] mendeteksi *sql injection* untuk *web application*

berbasis *Elastic-Pooling Convolutional Neural Network(CNN)*. Pada penelitian ini pengenalan sql injection berdasarkan EP-CNN secara otomatis mengekstrak fitur umum tersembunyi dari injeksi sql dan mengidentifikasi lalu lintas serangan. Menggunakan data set real log web. Hasil percobaan akurasi, presisi, f1, dan AUC lebih tinggi 0,999, menunjukkan model sangat baik bahkan garis ROC pada grafik mendekati persegi panjang sempurna.

Pada penelitian[4] menyajikan metode berbasis *long short-term memory (LSTM)* untuk sistem Intelligent Transportasi. Hasil penelitian berdasarkan pada validasi silang 10 kali lipat yang membagi dataset menjadi 10 subset yang tidak tumpang tindih dan mengambil 9 di antaranya sebagai data training secara bergantian, sisa 1 subset sebagai data uji. Pada penelitian didapatkan akurasi 93,47% lebih baik dari hasil deteksi menggunakan Bow untuk menyisipkan kata. Model pada penelitian ini menggunakan vektor fitur berdasarkan Word2vec.

K-Means Clustering adalah algoritma analisis clustering yang mengelompokkan objek berdasarkan nilai fiturnya ke dalam K cluster yang terpisah. Objek yang diklasifikasikan ke dalam cluster yang sama memiliki nilai fitur yang serupa. K adalah nilai positif yang menentukan jumlah cluster dan harus diberikan terlebih dahulu[5]. K-means merupakan algoritma yang sangat sederhana dan terkenal dalam memecahkan masalah clustering. Algoritma tersebut diketahui sangat efisien pada dataset yang besar.

Pada [6] melakukan sebuah penelitian analisa kinerja apache spark menggunakan K-means. Hasil yang didapat pada penelitian tersebut bahwa spark sangat efisien dan jauh lebih cepat ketika setiap ukuran dari dataset menghasilkan penurunan waktu dua kali lipat dibandingkan Map Reduce yang berarti bahwa Spark dapat digunakan pada pemrosesan big data.

Berdasarkan berbagai penelitian terkait di atas dapat dijadikan landasan penulis untuk meningkatkan performa model yang telah dibuat maka dapat di usulkan pada penelitian ini akan mengangkat metode K-means Clustering pada apache spark untuk mendeteksi serangan SQL Injection.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang dijelaskan, maka perumusan masalah yang akan dibahas yaitu sebagai berikut :

1) Bagaimana mengimplementasikan simulasi program untuk mengenali dan mendeteksi pola terhadap serangan dari SQL Injection pada apache spark?

2) Bagaimana cara model tersebut dapat menghitung akurasi deteksi serangan SQL Injection pada apache spark menggunakan metode K-means clustering?

3) Bagaimana kinerja dari deteksi serangan SQL Injection dengan metode K-means clustering terhadap akurasi, presisi, spesifisitas, dan F1-Score?

1.3 Tujuan

Adapun tujuan dari penulisan Tugas Akhir ini antara lain :

1) Mengembangkan sistem deteksi serangan SQL Injection pada apache spark.

2) Mendapatkan nilai optimal dari akurasi, presisi, recall, spesifitas, dan F1-Score dari deteksi serangan SQL Injection pada apache spark menggunakan metode K-means clustering.

3) Penerapan metode K-means clustering yang digunakan untuk mendeteksi pada serangan SQL Injection.

1.4 Manfaat

Manfaat dari penulisan Tugas Akhir ini, antara lain :

1) Dapat membantu dalam mendeteksi serangan SQL Injection pada lalu lintas jaringan.

2) Dapat menerangkan proses terjadinya penyerangan yang dilakukan oleh pelaku pada sistem korban.

3) Dapat membantu mengurangi waktu dan biaya yang diperlukan dalam mendeteksi serangan SQL Injection dengan menggunakan apache spark.

1.5 Batasan Masalah

Batasan Masalah pada Tugas Akhir ini, antara lain :

1) Penelitian ini menggunakan data dari *University of New Brunswick* (UNB).

2) Penelitian ini didasari dengan metode K-Means clustering menggunakan apache spark

3) Hasil penelitian ini berupa nilai akurasi, presisi, recall, dan F1-Score yang digunakan sebagai acuan untuk melihat tingkat kecocokan author dengan label.

1.6 Sistematika Penulisan

Sistematika yang akan digunakan dalam penulisan tugas akhir adalah sebagai berikut :

BAB I PENDAHULUAN

Bab pertama akan memaparkan sistematis mengenai latar belakang, tujuan penelitian, rumusan masalah, serta bentuk sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab kedua akan menjelaskan teori-teori dasar yang akan menjadi landasan dari penelitian ini.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan proses dan rangkaian kegiatan dalam penelitian.

BAB IV HASIL DAN ANALISIS

Bab ini akan memaparkan hasil pengujian yang diperoleh dan menjelaskan analisa terhadap hasil penelitian sementara yang telah dilakukan.

BAB V KESIMPULAN

Bab ini akan memaparkan kesimpulan sementara dari hasil yang telah didapat dari penelitian.

DAFTAR PUSTAKA

[1] Q. Li, W. Li, J. Wang, and M. Cheng, "A SQL Injection Detection Method Based on Adaptive Deep Forest," *IEEE Access*, vol. 7, pp. 145385–145394, 2019, doi: 10.1109/ACCESS.2019.2944951.

[2] A. A. Sarhan, S. A. Farhan, and F. M. Al-Harby, "Understanding and discovering sql injection vulnerabilities," *Adv. Intell. Syst. Comput.*, vol. 593, pp. 45–51, 2018, doi: 10.1007/978-3-319-60585-2_5.

[3] X. Xie, C. Ren, Y. Fu, J. Xu, and J. Guo, "SQL Injection Detection for Web Applications Based on Elastic-Pooling CNN," *IEEE Access*, vol. 7, pp. 151475–151481, 2019, doi: 10.1109/ACCESS.2019.2947527.

[4] Q. Li, F. Wang, J. Wang, and W. Li, "LSTM-Based SQL Injection Detection Method for Intelligent Transportation System," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4182–4191, 2019, doi: 10.1109/TVT.2019.2893675.

[5] L. Wan, G. Zhang, H. Li, and C. Li, "A Novel Bearing Fault Diagnosis Method Using Spark-Based Parallel ACO-K-Means Clustering Algorithm," *IEEE Access*, vol. 9, pp. 28753–28768, 2021, doi: 10.1109/ACCESS.2021.3059221.

[6] A. Alsirhani, S. Sampalli, and P. Bodorik, "DDoS Detection System: Using a Set of Classification Algorithms Controlled by Fuzzy Logic System in Apache Spark," *IEEE Trans. Netw. Serv. Manag.*, vol. PP, no. c, p. 1, 2019, doi: 10.1109/TNSM.2019.2929425.

[7] A. Joshi and V. Geetha, "SQL Injection detection using machine learning," *2014 Int. Conf. Control. Instrumentation, Commun. Comput. Technol. ICCICCT 2014*, no. 2, pp. 1111–1115, 2014, doi: 10.1109/ICCICCT.2014.6993127.

[8] N. Bharill, A. Tiwari, and A. Malviya, "Fuzzy Based Scalable Clustering Algorithms for Handling Big Data Using Apache Spark," *IEEE Trans. Big Data*, vol. 2, no. 4, pp. 339–352, 2016, doi: 10.1109/tbdata.2016.2622288.

[9] C. She, W. Wen, K. Zheng, and Y. Lyu, "Application-Layer DDoS

Detection by K-means Algorithm," vol. 50, no. Iceeecs, pp. 75–78, 2016, doi: 10.2991/iceeecs-16.2016.16.

[10] M. Assefi, E. Behravesh, G. Liu, and A. P. Tafti, "Big data machine learning using apache spark MLlib," *Proc. - 2017 IEEE Int. Conf. Big Data, Big Data 2017*, vol. 2018-Janua, pp. 3492–3498, 2017, doi: 10.1109/BigData.2017.8258338.

[11] L. Chen, Y. Zhang, Q. Zhao, G. Geng, and Z. Yan, "Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark," *Procedia Comput. Sci.*, vol. 134, pp. 310–315, 2018, doi: 10.1016/j.procs.2018.07.177.

[12] M. Belouch, S. El Hadaj, and M. Idlianmiad, "Performance evaluation of intrusion detection based on machine learning using apache spark," *Procedia Comput. Sci.*, vol. 127, pp. 1–6, 2018, doi: 10.1016/j.procs.2018.01.091.

[13] H. Zhang, S. Dai, Y. Li, and W. Zhang, "Real-time Distributed-Random-Forest-Based Network Intrusion Detection System Using Apache Spark," *2018 IEEE 37th Int. Perform. Comput. Commun. Conf. IPCCC 2018*, pp. 1–7, 2018, doi: 10.1109/PCCC.2018.8711068.

[14] T. Y. Chang and C. J. Hsieh, "Detection and analysis of distributed denial-of-service in internet of things-employing artificial neural network and apache spark platform," *Sensors Mater.*, vol. 30, no. 4, pp. 857–867, 2018, doi: 10.18494/SAM.2018.1789.

[15] M. Hasan, Z. Balbahaith, and M. Tarique, "Detection of SQL Injection Attacks: A Machine Learning Approach," *2019 Int. Conf. Electr. Comput. Technol. Appl. ICECTA 2019*, 2019, doi: 10.1109/ICECTA48151.2019.8959617.

[16] P. H. Pwint and T. Shwe, "Network Traffic Anomaly Detection based on Apache Spark," *2019 Int. Conf. Adv. Inf. Technol. ICAIT 2019*, pp. 222–226, 2019, doi: 10.1109/AITC.2019.8920897.

[17] A. Alsirhani, S. Sampalli, and P. Bodorik, "DDoS Detection System: Utilizing Gradient Boosting Algorithm and Apache Spark," *Can. Conf.*

Electr. Comput. Eng., vol. 2018-May, pp. 1–6, 2018, doi: 10.1109/CCECE.2018.8447671.

[18] Y. Gu, K. Li, Z. Guo, and Y. Wang, "Semi-supervised k-means ddos detection method using hybrid feature selection algorithm," *IEEE Access*, vol. 7, pp. 64351–64365, 2019, doi: 10.1109/ACCESS.2019.2917532.

[19] 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. Dl, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[20] S. V. Sivareddy and S. Saravanan, "Performance evaluation of classification algorithms in the Design of apache spark based intrusion detection system," *Proc. 5th Int. Conf. Commun. Electron. Syst. ICCES 2020*, no. Icces 2020, pp. 443–447, 2020, doi: 10.1109/ICCES48766.2020.09138066.

[21] S. Gumaste, D. G. Narayan, S. Shinde, and K. Amit, "Detection of DDoS attacks in openstack-based private cloud using apache spark," *J. Telecommun. Inf. Technol.*, vol. 2020, no. 4, pp. 62–71, 2020, doi: 10.26636/JTIT.2020.146120.

[22] I. S. Crespo-Martínez, A. Campazas-Vega, Á. M. Guerrero-Higueras, V. Riego-DelCastillo, C. Álvarez-Aparicio, and C. Fernández-Llamas, "SQL injection attack detection in network flow data," *Comput. Secur.*, vol. 127, 2023, doi: 10.1016/j.cose.2023.103093.

[23] M. A. Prabakar, M. Karthikeyan, and K. Marimuthu, "An efficient technique for preventing SQL injection attack using pattern matching algorithm," *2013 IEEE Int. Conf. Emerg. Trends Comput. Commun. Nanotechnology, ICE-CCN 2013*, no. Iceccn, pp. 503–506, 2013, doi: 10.1109/ICE-CCN.2013.6528551.

[24] A. Rai, M. M. I. Miraz, D. Das, H. Kaur, and Swati, "SQL Injection: Classification and Prevention," *Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021*, pp. 367–372, 2021, doi: 10.1109/ICIEM51511.2021.9445347.

[25] P. R. McWhirter, K. Kifayat, Q. Shi, and B. Askwith, “SQL Injection Attack classification through the feature extraction of SQL query strings using a Gap-Weighted String Subsequence Kernel,” *J. Inf. Secur. Appl.*, vol. 40, pp. 199–216, 2018, doi: 10.1016/j.jisa.2018.04.001.

[26] K. Zhang, “A machine learning based approach to identify SQL injection vulnerabilities,” *Proc. - 2019 34th IEEE/ACM Int. Conf. Autom. Softw. Eng. ASE 2019*, pp. 1286–1288, 2019, doi: 10.1109/ASE.2019.00164.

[27] A. S. and R. M., “A Review of Hadoop Ecosystem for BigData,” *Int. J. Comput. Appl.*, vol. 180, no. 14, pp. 35–40, 2018, doi: 10.5120/ijca2018916273.

[28] A. Raj and R. D’Souza, “A Review on Hadoop Eco System for Big Data,” *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, no. February 2019, pp. 343–348, 2019, doi: 10.32628/cseit195172.

[29] @theprogrammedwords, “Hadoop Ecosystem,” *GeekfsorGeeks*, 2021. https://www.geeksforgeeks.org/hadoop-ecosystem/.

[30] K. Hildebrandt, F. Panse, N. Wilcke, and N. Ritter, “Large-Scale Data Pollution with Apache Spark,” *IEEE Trans. Big Data*, vol. 6, no. 2, pp. 396–411, 2020, doi: 10.1109/TBDATA.2016.2637378.

[31] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, “Big data analytics on Apache Spark,” *Int. J. Data Sci. Anal.*, vol. 1, no. 3–4, pp. 145–164, 2016, doi: 10.1007/s41060-016-0027-9.

[32] S. Shah, Y. Amannejad, and Di. Krishnamurthy, “Diaspore: Diagnosing Performance Interference in Apache Spark,” *IEEE Access*, vol. 9, pp. 103230–103243, 2021, doi: 10.1109/ACCESS.2021.3098426.

[33] X. Meng *et al.*, “MLlib: Machine learning in Apache Spark,” *J. Mach. Learn. Res.*, vol. 17, pp. 1–7, 2016.

[34] S. Jonnalagadda, P. Srikanth, K. Thumati,] Sri, H. Nallamala, and A. Professors, “A Review Study of Apache Spark in Big Data Processing,” *Int. J. Comput. Sci. Trends Technol.*, vol. 4, no. 3, pp. 93–98, 2013, [Online]. Available: www.ijcstjournal.org.

[35] I. Mavridis and H. Karatza, “Performance evaluation of cloud-based log

file analysis with Apache Hadoop and Apache Spark," *J. Syst. Softw.*, vol. 125, pp. 133–151, 2017, doi: 10.1016/j.jss.2016.11.037.

[36] E. E. Drakonaki and G. M. Allen, "Magnetic resonance imaging, ultrasound and real-time ultrasound elastography of the thigh muscles in congenital muscle dystrophy," *Skeletal Radiol.*, vol. 39, no. 4, pp. 391–396, 2010, doi: 10.1007/s00256-009-0861-0.

[37] H. Luu, "Beginning Apache Spark 2," *Begin. Apache Spark 2*, pp. 1–13, 2018, doi: 10.1007/978-1-4842-3579-9.

[38] M. Ribeiro, K. Grolinger, and M. A. M. Capretz, "MLaaS: Machine learning as a service," *Proc. - 2015 IEEE 14th Int. Conf. Mach. Learn. Appl. ICMLA 2015*, no. c, pp. 896–902, 2016, doi: 10.1109/ICMLA.2015.152.

[39] M. Batta, "Machine Learning Algorithms - A Review," *Int. J. Sci. Res.*, vol. 18, no. 8, pp. 381–386, 2018, doi: 10.21275/ART20203995.

[40] C. Janiesch, P. Zschech, and K. Heinrich, "Machine learning and deep learning," *Electron. Mark.*, vol. 31, no. 3, pp. 685–695, 2021, doi: 10.1007/s12525-021-00475-2.

[41] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, "Machine Learning for Fluid Mechanics," *Annu. Rev. Fluid Mech.*, vol. 52, pp. 477–508, 2020, doi: 10.1146/annurev-fluid-010719-060214.

[42] T. M. Kodinariya and P. R. Makwana, "Review on determining of cluster in K-means," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 1, no. 6, pp. 90–95, 2013, [Online]. Available: https://www.researchgate.net/publication/313554124.

[43] S. Shukla, "A Review ON K-means DATA Clustering APPROACH," *Int. J. Inf. Comput. Technol.*, vol. 4, no. 17, pp. 1847–1860, 2014, [Online]. Available: http://www.irphouse.com.

[44] C. Zhang and S. Xia, "K-means clustering algorithm with improved initial center," *Proc. - 2009 2nd Int. Work. Knowl. Discov. Data Mining, WKKD 2009*, pp. 790–792, 2009, doi: 10.1109/WKDD.2009.210.

[45] S. Dwivedi and L. K. P.Bhaiya, "A Systematic Review on K-Means

Clustering Techniques," *Int. J. Sci. Res. Eng. Trends*, vol. 5, no. 3, pp. 750–752, 2019.

[46] A. Singh, A. Yadav, and A. Rana, "K-means with Three different Distance Metrics," *Int. J. Comput. Appl.*, vol. 67, no. 10, pp. 13–17, 2013, doi: 10.5120/11430-6785.

[47] G. K. Armah, G. Luo, and K. Qin, "A Deep Analysis of the Precision Formula for Imbalanced Class Distribution," *Int. J. Mach. Learn. Comput.*, vol. 4, no. 5, pp. 417–422, 2014, doi: 10.7763/ijmlc.2014.v4.447.

[48] M. Navin and Pankaja, "Performance Analysis of Text Classification Algorithm using Confusion Matrix," *Int. J. Eng. Tech. Res.*, vol. 6, no. 4, pp. 75–78, 2016.

[49] I. Technology and I. Technology, "Hossin, M. 1 and Sulaiman, M.N. 2 1," vol. 5, no. 2, pp. 1–11, 2015.

[50] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00382-x.

[51] Q. Zhou and D. Pezaros, "Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection -- An Analysis on CIC-AWS-2018 dataset," no. July, 2019, [Online]. Available: http://arxiv.org/abs/1905.03685.

[52] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSP 2018 - Proc. 4th Int. Conf. Inf. Syst. Secur. Priv.*, vol. 2018-Janua, no. Cic, pp. 108–116, 2018, doi: 10.5220/0006639801080116.

[53] A. Naveen and T. Velmurugan, "Identification of calcification in MRI brain images by k-means algorithm," *Indian J. Sci. Technol.*, vol. 8, no. 29, 2015, doi: 10.17485/ijst/2015/v8i29/83379.

[54] C. Zhu, C. U. Idemudia, and W. Feng, "Improved logistic regression model for diabetes prediction by integrating PCA and K-means techniques," *Informatics Med. Unlocked*, vol. 17, no. March, p. 100179, 2019, doi: 10.1016/j.imu.2019.100179.