Madani: Jurnal Ilmiah Multidisiplin
Volume 2, Nomor 7, 2024, Halaman 266-275

E-ISSN: 2986-6340
DOI: https://doi.org/10.5281/zenodo.12562358

Optimalisasi Deteksi Anomali Untuk Pemfilteran Log dan Integrasi Dengan SIEM Menggunakan Machine Learning

Salsabila Amalia Harjanto[1], Mutiara Nurhaliza[2], Jody Hezekiah Tanasa Sagala[3]
[123]Department of Information Technology, Sepuluh Nopember Institute of Technology Surabaya, Indonesia
Email: harjantosalsabila@gmail.com[1], mutiaranurhaliza@gmail.com[2], hezesagala.hs@gmail.com[3]

Abstract
Cybersecurity has become a paramount concern in today's digital age, necessitating robust systems like Security Information and Event Management (SIEM) for effective threat detection through log analysis. Traditional methods often prove inadequate due to static rules prone to false positives. In this study, we propose a Machine Learning-based approach to optimize anomaly detection in Hadoop Distributed File System (HDFS) logs. Evaluating Decision Tree, Naive Bayes, Log Clustering, Support Vector Machine (SVM), and Logistic Regression, Log Clustering emerges with the highest accuracy at 98.19% and the highest recall at 56.05% among the models tested. These findings underscore Log Clustering's efficacy in enhancing cybersecurity in big data environments, particularly in its efficiency for integration with SIEM systems.
Keywords:Cyber Security, SIEM, Anomaly Detection, HDFS Log, Machine Learning

Article Info
Received date: 10 June 2024 Revised date: 18 June 2024 Accepted date: 23 June 2024

PENDAHULUAN

Dalam era digital yang semakin maju, keamanan siber menjadi salah satu aspek yang paling krusial bagi organisasi di seluruh dunia. Ancaman siber terus berkembang dengan cepat, baik dari segi kompleksitas maupun volume. Sistem Informasi dan Manajemen Keamanan (SIEM) telah menjadi tulang punggung dalam upaya keamanan siber, menyediakan kemampuan untuk mengumpulkan, menganalisis, dan merespons data log dari berbagai sumber. Namun, beberapa tantangan terbesar yang dihadapi SIEM saat ini adalah volume log yang sangat besar, sumber data yang heterogen, tingginya jumlah alarm false positive, serta kompleksitas log yang dapat membuat proses deteksi ancaman menjadi sulit dan memakan waktu (1).

Pemfilteran log dengan pendekatan tradisional seringkali terbatas pada aturan statis tidak dapat mengatasi tantangan tersebut. Akibatnya, banyak aktivitas mencurigakan yang tidak terdeteksi atau sebaliknya, aktivitas normal yang dianggap sebagai ancaman. Untuk mengatasi masalah ini, pendekatan berbasis pembelajaran mesin (Machine Learning) dapat menjadi solusi yang lebih dinamis dan adaptif. Dengan memanfaatkan algoritma pembelajaran mesin, sistem dapat belajar dari data historis dan terus memperbarui model deteksi ancamannya berdasarkan pola baru yang teridentifikasi.

Implementasi pembelajaran mesin dalam pemfilteran log dapat dioptimalkan dengan menerapkan beberapa algoritma pembelajaran mesin yang memiliki karakteristik dan pendekatan unik dalam menangani data log. Pada penelitian ini, akan digunakan lima algoritma pembelajaran mesin yaitu *Decision Tree*, *Naive Bayes*, *Log Clustering*, *Support Vector Machine* (SVM), dan *Logistic Regression*. Dengan melakukan evaluasi dan membandingkan kinerja dari kelima algoritma ini, akan dapat diidentifikasi model yang paling efektif dalam mendeteksi anomali dan membedakannya dari aktivitas normal dalam dataset log HDFS (Hadoop Distributed File System). Evaluasi ini akan mencakup pengukuran akurasi, presisi, recall, dan F1-score untuk menunjukkan gambaran menyeluruh tentang efektivitas masing-masing algoritma.

Dataset log HDFS terdiri dari log operasi yang mencatat berbagai aktivitas sistem, baik yang normal maupun yang berpotensi mencurigakan atau anomali. Dataset ini dipilih karena memiliki lingkungan informasi yang terdistribusi besar sehingga dapat mencerminkan tantangan nyata dalam lingkungan sistem informasi yang besar dan terdistribusi (2). Dengan menggunakan dataset ini, penelitian ini akan menyimulasikan situasi dunia nyata di mana sistem SIEM perlu beroperasi. Hasil dari penelitian ini diharapkan dapat memberikan rekomendasi praktis bagi para profesional keamanan

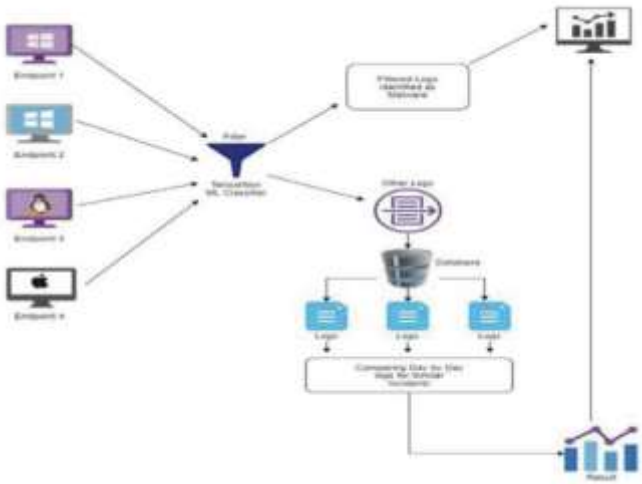
siber dalam memilih dan mengimplementasikan algoritma pembelajaran mesin yang paling sesuai untuk meningkatkan efektivitas deteksi ancaman dalam lingkungan mereka.

STUDI LITERATUR

Pemfilteran Log

Pemfilteran log merupakan suatu proses penting dalam analisis sistem yang mengandalkan log untuk memahami status dan kinerja sistem yang bersangkutan, terutama dalam konteks deteksi dan pencegahan ancaman keamanan yang kompleks. Log files menjadi sumber informasi utama untuk berbagai pertanyaan terkait sistem, bahkan seringkali menjadi satu-satunya cara untuk mendapatkan informasi tentang kesalahan yang terjadi dalam sistem yang terdistribusi. Pemfilteran log memungkinkan organisasi untuk memonitor dan menganalisis aktivitas jaringan dan sistem mereka dengan lebih efektif, dengan tujuan untuk mendeteksi indikasi potensial dari serangan cyber, perilaku mencurigakan, atau pelanggaran keamanan lainnya. Dengan menganalisis log secara efektif, peneliti dan praktisi dapat mengidentifikasi pola, anomali, dan masalah tersembunyi dalam sistem, yang pada gilirannya dapat membantu dalam pengambilan keputusan yang lebih baik (3)

Pemfilteran log pada SIEM (*Security Information and Event Management*) memiliki beberapa komponen penting yang perlu dipertimbangkan. Salah satunya adalah pemfilteran log berbasis endpoint yang merupakan aspek utama dalam perlindungan terhadap malware. Dalam konteks ini, fokusnya terbagi menjadi dua bagian utama: perlindungan berbasis endpoint terhadap malware dan perlindungan log yang difilter terhadap malware.

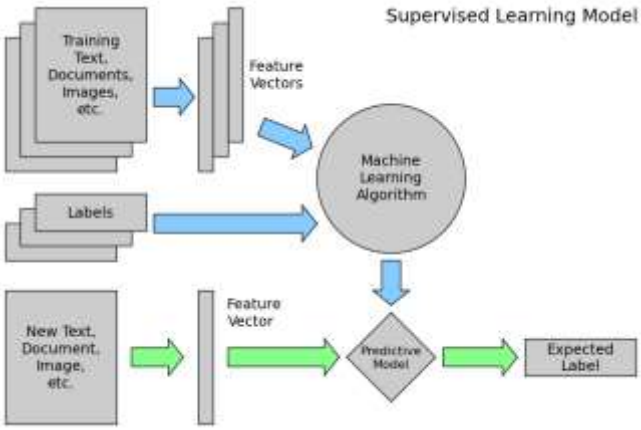


Gambar 1. Endpoint filtering logs

Terdapat beberapa pendekatan yang dapat dilakukan untuk penyaringan log meliputi: analisis log, otomatisasi proses pemfilteran log, identifikasi log malware dan pengarahannya ke SIEM, serta pengurangan alarm yang dihasilkan oleh SIEM. Metode pemfilteran menggunakan machine learning akan membantu mengotomatisasi proses identifikasi kasus melalui alarm oleh SOC analyst level 1. Hal ini akan mengarah pada peningkatan efisiensi operasional dan mengurangi beban kerja analis (4).

Supervised Learning

Supervised learning adalah metode yang paling umum digunakan dalam masalah klasifikasi, di mana tujuannya adalah mengajarkan mesin untuk memahami dan menerapkan sistem klasifikasi yang telah ditetapkan sebelumnya. Proses ini melibatkan pengumpulan dataset yang terdiri dari fitur-fitur dan label-label yang sesuai. Tujuan utamanya adalah mengembangkan model yang dapat memprediksi label dari suatu objek berdasarkan fitur-fitur yang dimilikinya. Algoritma pembelajaran menerima serangkaian fitur bersama dengan output yang sudah diketahui, dan belajar dengan membandingkan prediksi yang dihasilkan dengan output yang sebenarnya untuk mengidentifikasi dan memperbaiki kesalahan.



Gambar 2. Supervised Learning Model (5)

Pemodelan ini adalah teknik yang sering digunakan untuk melatih *Neural Networks* dan *Decision Trees*, keduanya mengandalkan informasi klasifikasi yang telah ditentukan sebelumnya. Metode ini juga diterapkan dalam berbagai aplikasi di mana data historis dimanfaatkan untuk meramalkan kejadian di masa depan. Sebagaimana disebutkan sebelumnya, tugas supervised learning dibagi menjadi dua kategori: klasifikasi dan regresi. Dalam klasifikasi, label bersifat diskrit, sementara dalam regresi, label bersifat kontinu (6).

Beberapa algoritma *supervised learning* yang umum digunakan adalah *Decision Trees*, *Naive Bayes*, *Random Forest*, dan *Support Vector Machine* (SVM). Setiap pemodelan memiliki metode pendekatan yang unik dalam klasifikasi dan regresi, sehingga memiliki nilai evaluasi yang berbeda-beda untuk setiap dataset.

Large Data Set 768		
Algorithm	**Precision of YES (Positive Diabetes)**	**Precision of NO (Negative Diabetes**
SVM	0.74	0.785
Naïve Bayes	0.678	0.802
JRip	0.659	0.78
Random Forest	0.653	0.791
Neural Networks (Perceptron)	0.653	0.799
Decision Tree (J48)	0.632	0.79
Decision Table	0.619	0.771

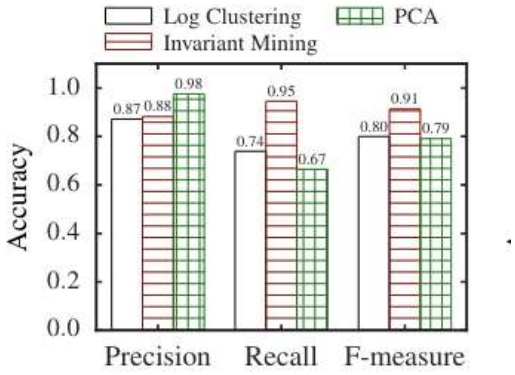
Gambar 3. Pemeringkatan Presisi Diabetes Positif dan Diabetes Negatif menggunakan algoritma yang berbeda

Pada gambar 3 dilakukan perbandingan beberapa algoritma supervised learning dengan menggunakan dataset diabetes. Dilakukan 2 pelabelan yaitu "YES" yang berarti positif diabetes atau "NO" yang berarti negatif diabetes. Berdasarkan hasil perbandingan presisi, algoritma SVM memiliki nilai presisi tertinggi untuk prediksi label "YES" dan algoritma Naive Bayes memiliki nilai presisi tertinggi untuk prediksi label "NO" (7).

2.3 *Unsupervised Learning*

Unsupervised learning merupakan bagian integral dari bidang *machine learning* yang menemukan pola dalam kumpulan data besar atau mengelompokkannya ke dalam kategori tanpa dilatih secara eksplisit menggunakan data berlabel (8). *Unsupervised learning* mempelajari bagaimana sistem dapat belajar mewakili pola-pola input dengan cara yang mencerminkan struktur statistik dari seluruh koleksi pola input tanpa memerlukan output target yang jelas (9). Pendekatan ini memberikan keunggulan khususnya dalam situasi lingkungan produksi di mana data berlabel terbatas karena tidak memerlukan data pelatihan yang disertai label secara eksplisit. Beberapa metode umum dalam

pembelajaran tanpa pengawasan meliputi metode log clustering, invariant mining, dan analisis reduksi dimensi seperti *Principal Component Analysis* (PCA).



Gambar 4. Akurasi Metode Unsupervised Learning pada data

Grafik pada gambar 4 menunjukan bahwa dari tiga metode yang dibandingkan, invariants mining menunjukkan kinerja unggul (dengan *F-measure* 0.91) dibandingkan dengan metode-metode deteksi anomali lainnya pada dataset. *Invariants mining* secara otomatis membentuk pola korelasi linear untuk mendeteksi anomali. Kinerja rendah dari log clustering disebabkan oleh sifat matriks perhitungan kejadian yang berdimensi tinggi dan jarang. Hal ini membuat log clustering kesulitan dalam memisahkan antara anomali dan instansi normal dengan efektif, sehingga sering kali menghasilkan banyak positif palsu. Secara umum, metode-metode *unsupervised learning* cenderung memiliki kinerja lebih rendah dibandingkan dengan metode supervised learning. Namun, invariants mining menonjol sebagai pendekatan yang menjanjikan dengan kinerja yang stabil dan tinggi. Sebagian besar metode deteksi anomali berskala linier dengan ukuran log, meskipun *Log Clustering* dan *Invariants Mining* masih memerlukan optimisasi lebih lanjut untuk meningkatkan kecepatan (10).

2.4 *Hadoop Distributed File System*

HDFS (*Hadoop Distributed File System*) adalah bagian dari Apache Hadoop dan menyimpan set data besar secara konsisten, dengan menggandakan data ke berbagai node untuk kemudahan akses dan pengenalan kesalahan (11). HDFS secara dinamis menggandakan file data berdasarkan analisis prediktif, meningkatkan ketersediaan dan keandalan dalam pengelolaan Big Data (12).

```
LineId,Date,Time,Pid,Level,Component,Content,EventId,EventTemplate
1,081109,203518,143,INFO,dfs.DataNode$DataXceiver,Receiving block blk_-1608999687919862906 src:
/10.250.19.102:54106 dest: /10.250.19.102:50010,E5,Receiving block <*> src: /<*> dest: /<*>
2,081109,203518,35,INFO,dfs.FSNamesystem,BLOCK* NameSystem.allocateBlock:
/mnt/hadoop/mapred/system/job_200811092030_0001/job.jar. blk_-1608999687919862906,E22,BLOCK*
NameSystem.allocateBlock:<*>
3,081109,203519,143,INFO,dfs.DataNode$DataXceiver,Receiving block blk_-1608999687919862906 src:
/10.250.10.6:40524 dest: /10.250.10.6:50010,E5,Receiving block <*> src: /<*> dest: /<*>
4,081109,203519,145,INFO,dfs.DataNode$DataXceiver,Receiving block blk_-1608999687919862906 src:
/10.250.14.224:42420 dest: /10.250.14.224:50010,E5,Receiving block <*> src: /<*> dest: /<*>
5,081109,203519,145,INFO,dfs.DataNode$PacketResponder,PacketResponder 1 for block blk_-1608999687919862906
terminating,E11,PacketResponder <*> for block <*> terminating
```

Gambar 5. Contoh log dataset HDFS

HDFS menyimpan dan mengalirkan set data besar ke aplikasi pengguna dengan efisien, memungkinkan pengelolaan data yang efisien dan ekonomis untuk perusahaan besar seperti Yahoo! (13). Sistem ini adalah teknologi yang handal, toleran terhadap kesalahan, dan dapat diskalakan untuk memproses jumlah data tak terstruktur dalam skala besar, menyediakan toleransi kesalahan, akses informasi yang cepat, dan dapat diskalakan (14). Hadoop mengelola jumlah besar data tak terstruktur dengan menggunakan teknik seperti pengumpulan data, pra-pemrosesan, dan algoritma Fragmentasi untuk meningkatkan kinerja dan mengurangi waktu respons (15). Hadoop menggunakan pendekatan map-reduce, membagi aplikasi menjadi fragmen kecil dan menjalankannya pada node apa pun di dalam kluster, untuk menyimpan dan memproses data tak terstruktur dengan efisien (16).

METODE

Sumber Data

Dataset yang digunakan dalam penelitian ini adalah dataset log dari *Hadoop Distributed File System* (HDFS). Dataset ini berisi informasi mengenai aktivitas sistem seperti operasi file, penggunaan sumber daya, dan konfigurasi sistem yang tercatat dalam bentuk entri log.

Prepocessing Data

Data yang diterima dalam *preprocessing* log mungkin mengandung *noise* atau kekotoran yang perlu dibersihkan sebelum dilakukan deteksi anomali. Data kotor bisa tidak akurat, tidak konsisten, dan tidak lengkap karena kesalahan dalam kumpulan data [17]. Proses pembersihan data akan fokus pada penghapusan entri log tidak relevan, penanganan nilai yang tidak wajar seperti timestamp tidak konsisten, dan memastikan format entri log konsisten untuk analisis lebih lanjut. Entri log yang menunjukkan kesalahan atau aktivitas tidak biasa akan diidentifikasi sebagai anomali dan diproses terpisah dalam analisis deteksi anomali log.

Algoritma Model

Untuk mendeteksi anomali dalam dataset log, kami menguji lima algoritma utama: *Decision Tree*, *Support Vector Machine* (SVM), *Logistic Regression*, *Naive Bayes*, dan *Log Clustering*. Setiap algoritma memberikan pendekatan yang berbeda dalam mengidentifikasi entri log yang menyimpang dari pola normal.

Decision Tree

Decision Tree adalah diagram struktur pohon yang menggunakan cabang untuk mengilustrasikan keadaan yang diprediksi untuk setiap contoh. *Decision Tree* dibangun secara *top-down* menggunakan data pelatihan. Setiap node pohon dibuat dengan menggunakan atribut "terbaik" saat ini, yang dipilih berdasarkan gain informasi atribut [18].

Support Vector Machine (SVM)

SVM adalah pengklasifikasi vektor berdasarkan teori pembelajaran statistik, dengan kinerja yang ditingkatkan saat menggunakan fungsi kernel dalam tugas klasifikasi [19]. SVM menunjukkan kemampuan generalisasi yang tinggi dan mengungguli algoritma pembelajaran klasik dalam masalah klasifikasi dua kelompok [20].

Logistic Regression

Logistic Regression adalah metode statistik untuk menganalisis pengaruh variabel independen terhadap variabel dependen biner, menghasilkan estimasi probabilitas antara 0 dan 1 [21]. *Logistic Regression* dapat menyesuaikan dengan beberapa prediktor, mengurangi bias potensial dan meningkatkan hasil [22].

Naive Bayes

Naive Bayes adalah skema pembelajaran yang menggunakan Teorema Bayes untuk langsung mengarahkan pencariannya, menghasilkan model yang lebih akurat daripada pembelajaran sederhana dan dapat bersaing dengan sistem ILP yang lebih canggih [23].

Log Clustering

Log Clustering adalah proses mempartisi log sistem menjadi berbagai klaster berdasarkan perilaku pengguna, mengidentifikasi anomali, dan menghapus kejadian normal [24]. *Log clustering* adalah metode efisien untuk menyimpan, mengelola, dan menganalisis log web skala besar menggunakan kerangka komputasi terdistribusi *Map/Reduce* [25].

Metrik Evaluasi

Untuk mengevaluasi efektivitas algoritma yang digunakan dalam mendeteksi anomali pada dataset log, berbagai metrik evaluasi diterapkan. Metrik ini membantu mengukur kinerja model dalam aspek ketepatan, sensitivitas, akurasi, dan kemampuan keseluruhan dalam mengidentifikasi anomali dengan benar. Berikut adalah metrik evaluasi yang kami gunakan dalam penelitian ini:

Akurasi (*Accuracy*)

Akurasi dapat mengukur persentase dari prediksi yang benar terhadap keseluruhan prediksi yang dibuat oleh model. Akurasi memberikan gambaran umum tentang seberapa baik model dapat memprediksi kelas yang benar. Metrik ini dihitung dengan rumus:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN}$$

dimana TP (*True Positive*) adalah jumlah anomali yang terdeteksi dengan benar sebagai anomali, TN (*True Negative*) adalah jumlah data normal yang terdeteksi dengan benar sebagai normal, FP (*False*

Positive) adalah jumlah data normal yang salah terdeteksi sebagai anomali, dan FN (*False Negative*) adalah jumlah anomali yang salah terdeteksi sebagai normal.

Precision

Precision mengukur proporsi dari prediksi positif yang benar terhadap semua prediksi positif yang dibuat oleh model. Metrik ini penting dalam konteks di mana biaya *False Positive* tinggi, seperti dalam sistem deteksi anomali yang kritis. *Precision* dihitung dengan rumus:

$$\text{Precision} = \frac{TP}{TP + FP}$$

Precision yang tinggi menunjukkan bahwa model memiliki tingkat kesalahan yang rendah dalam mendeteksi anomali.

***Recall* (Sensitivitas atau *True Positive Rate*)**

Recall mengukur kemampuan model dalam menemukan semua entri yang benar-benar merupakan anomali. Ini dihitung dengan rumus:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Recall yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar anomali yang ada dalam dataset, sehingga meminimalkan jumlah *False Negative*.

F1-Score

F1-Score adalah rata-rata harmonik dari *precision* dan *recall*, memberikan keseimbangan antara kedua metrik tersebut. *F1-Score* digunakan untuk memberikan penilaian yang lebih holistik tentang kinerja model, terutama ketika ada ketidakseimbangan antara kelas normal dan anomali. *F1-Score* dihitung dengan rumus:

$$F1\text{-}Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Nilai *F1-Score* yang tinggi menunjukkan bahwa model memiliki *precision* dan *recall* yang seimbang, yang penting dalam konteks deteksi anomali.

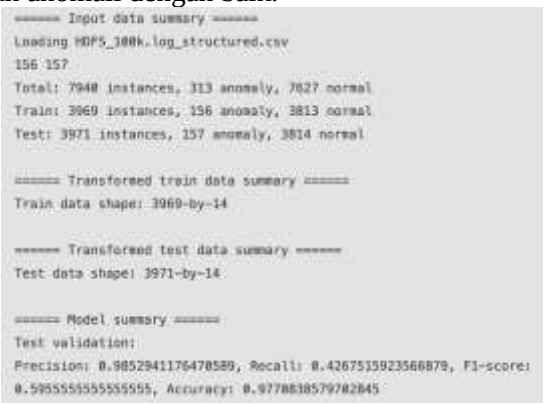
HASIL DAN PEMBAHASAN

Pada bab ini, akan dibahas evaluasi hasil dari penerapan berbagai model deteksi anomali yang telah dijelaskan dalam metodologi. Model-model ini dievaluasi berdasarkan kinerja mereka dalam mendeteksi anomali pada dataset HDFS menggunakan metrik evaluasi seperti *Accuracy*, *Precision*, *Recall*, dan *F1-Score*. Evaluasi ini bertujuan untuk menilai efektivitas masing-masing model dalam mengidentifikasi anomali serta untuk menentukan model yang paling optimal.

Evaluasi Hasil Pengujian

1. Decision Tree

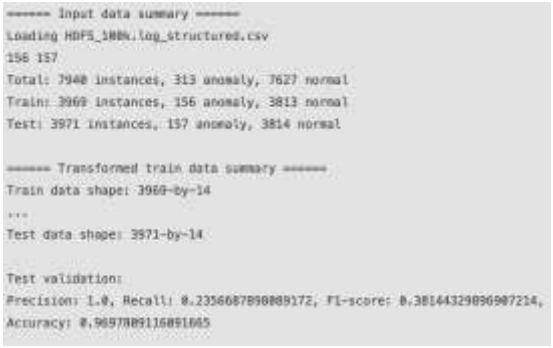
Model *Decision Tree* menunjukkan performa yang baik dengan precision 98.5%, *recall* 42.7%, *F1-score* 59.6%, dan akurasi 97.71%. Model ini efektif dalam mengenali sebagian besar instance normal dan sejumlah anomali dengan baik.



Gambar 6. Hasil dari model *Decision Tree*

2. *Support Vector Machine* (SVM)

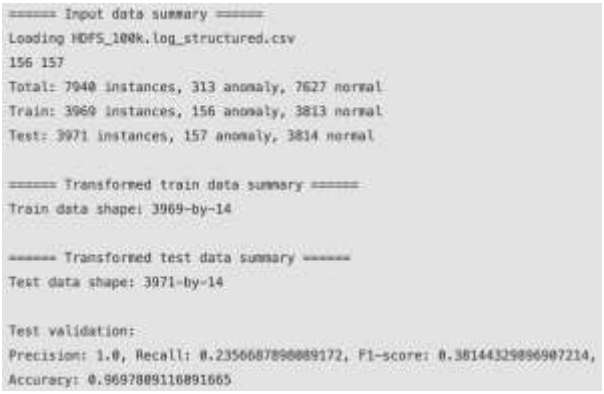
Support Vector Machine (SVM) mencapai *precision* 100%, *recall* 23.6%, F1-score 38.1%, dan akurasi 96.98%. *Precision* yang tinggi menunjukkan kemampuan model dalam mengidentifikasi dengan tepat anomali yang ada, tetapi recall yang rendah menandakan bahwa model hanya mengenali sebagian kecil dari keseluruhan anomali.



Gambar 7. Hasil dari model *Support Vector Machine*

3. *Logistic Regression*

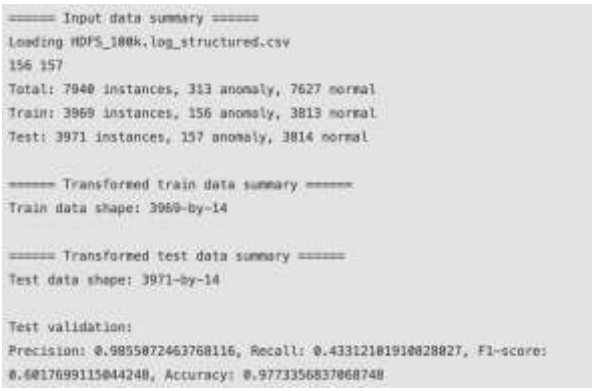
Logistic Regression juga menunjukkan *precision* 100%, *recall* 23.6%, *F1-score* 38.1%, dan akurasi 96.98%. Seperti SVM, model ini memiliki *precision* yang tinggi namun *recall* yang rendah, menunjukkan kecenderungan untuk mengenali lebih baik *instance* normal daripada mendeteksi semua anomali.



Gambar 8. Hasil dari model *Logistic Regression*

4. *Naive Bayes*

Naive Bayes menunjukkan *precision* 98.55%, *recall* 43.31%, *F1-score* 60.18%, dan akurasi 97.73%. Model ini efektif dalam mengenali sebagian besar instance normal dan sejumlah anomali dengan baik.



Gambar 9. Hasil dari model *Naive Bayes*

5. *Log Clustering*

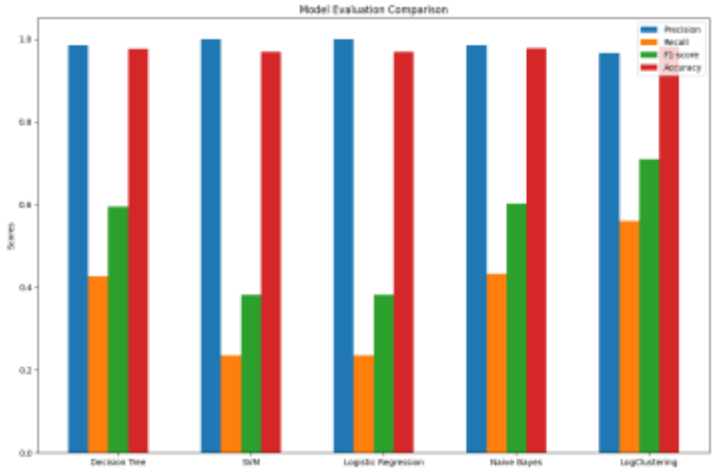
Model *Log Clustering* mencapai *precision* 96.70%, *recall* 56.05%, *F1-score* 70.97%, dan akurasi 98.19%. *Precision* tinggi menunjukkan bahwa model mampu mengidentifikasi dengan tepat sebagian besar anomali yang ada, sementara recall yang lebih tinggi dari model lain menunjukkan kemampuannya dalam mendeteksi lebih banyak anomali.

```
====== Input data summary ======
Loading HDFS_100k.log_structured.csv
156 157
Total: 7940 instances, 313 anomaly, 7627 normal
Train: 3969 instances, 156 anomaly, 3813 normal
Test: 3971 instances, 157 anomaly, 3814 normal

====== Transformed train data summary ======
Train data shape: 3969-by-14

====== Transformed test data summary ======
Test data shape: 3971-by-14

====== Model summary ======
Starting offline clustering...
Processed 1000 instances.
Found 3 clusters offline.

Starting online clustering...
Processed 2000 instances.
Processed 3813 instances.
Found 3 clusters online.

Test validation:
Precision: 0.967032967032967, Recall: 0.5605095541401274, F1-score:
0.7096774193548387, Accuracy: 0.9818685469654999
```

Gambar 10. Hasil dari model *Log Clustering*

Hasil Perbandingan

Pada bagian ini, akan dibahas perbandingan hasil dari lima algoritma yang telah dijelaskan sebelumnya: *Decision Tree*, *Support Vector Machine* (SVM), *Logistic Regression*, *Naive Bayes*, dan *Log Clustering*, sebagaimana ditunjukkan dalam Gambar 11. *Decision Tree* mencapai akurasi sebesar 97,71%, *precision* 98,5%, *recall* 42,7%, dan *F1-score* 59,6%. SVM mencapai akurasi 96,98%, *precision* 100%, *recall* 23,6%, dan *F1-score* 38,1%. *Logistic Regression* mencapai akurasi 96,98%, *precision* 100%, *recall* 23,6%, dan *F1-score* 38,1%. *Naive Bayes* menunjukkan performa dengan akurasi 97,73%, *precision* 98,55%, *recall* 43,31%, dan *F1-score* 60,18%. Sedangkan *Log Clustering* mencapai akurasi 98,19%, *precision* 96,70%, *recall* 56,05%, dan *F1-score* 70,97%. Dengan demikian, hasil evaluasi menunjukkan bahwa *Log Clustering* memberikan hasil terbaik pada tingkat akurasi 98,19% dan *recall* yang paling tinggi diantara model lain dengan tingkat *recall* 56,05%.



Gambar 11. Perbandingan antara *Decision Tree*, SVM, *Logistic Regression*, dan *Log Clustering*

Secara keseluruhan, kelima model yang disebutkan memberikan hasil yang relatif serupa dalam hal akurasi dan *precision*. Namun, dalam hal *recall* dan *F1-score*, *Log Clustering* lebih unggul dibandingkan dengan *Decision Tree*, SVM, *Logistic Regression*, dan *Log Clustering*. Hal ini menunjukkan keunggulan pendekatan clustering untuk deteksi anomali, di mana Log Clustering mampu mengenali lebih banyak kasus anomali secara efektif dibandingkan dengan model lain yang menggunakan pendekatan berbasis aturan atau probabilistik.

SIMPULAN

Berdasarkan hasil pengujian algoritma model untuk memfilter log dengan metode deteksi anomali yang telah diteliti dapat dibuat kesimpulan sebagai berikut.

1. Log Clustering mencapai akurasi sebesar 98,19%, yang merupakan yang tertinggi dibandingkan dengan algoritma lain yang Anda evaluasi. Selain itu, Log Clustering juga memiliki recall yang lebih tinggi (56,05%) dibandingkan dengan SVM dan Logistic Regression, yang memiliki recall 23,6%.
2. Hasil dari evaluasi juga menunjukkan bahwa meskipun Decision Tree, SVM, Logistic Regression, dan Naive Bayes memberikan kinerja yang baik dalam beberapa metrik. Namun kurang mampu dalam mendeteksi sejumlah besar anomali seperti yang dilakukan oleh Log Clustering.
3. Dengan demikian, untuk meningkatkan efektivitas deteksi ancaman dalam lingkungan SIEM, penting untuk mempertimbangkan penggunaan algoritma seperti Log Clustering yang mampu menangani volume data besar dan kompleksitas log yang tinggi dengan lebih baik.

REFERENSI

1. Sapegin, A., Jaeger, D., Cheng, F., & Meinel, C. (2017). Towards a system for complex analysis of security events in large-scale networks. *Comput. Secur.*, 67, 16-34. https://doi.org/10.1016/j.cose.2017.02.001.
2. Asim, M., McKinnel, D., Dehghantanha, A., Parizi, R., Hammoudeh, M., & Epiphaniou, G. (2019). Big Data Forensics: Hadoop Distributed File Systems as a Case Study. , 179-210. https://doi.org/10.1007/978-3-030-10543-3_8.
3. Zwietasch, T. (2014). *Detecting anomalies in system log files using machine learning techniques* (Bachelor's thesis).
4. Perera, A., Rathnayaka, S., Perera, N. D., Madushanka, W. W., & Senarathne, A. N. (2021, April). The next gen security operation center. In *2021 6th International Conference for Convergence in Technology (I2CT)* (pp. 1-9). IEEE.
5. https://radimrehurek.com/data_science_python/
6. Nasteski, V. (2017). An overview of the supervised machine learning methods. *Horizons. b*, *4*(51-62), 56.
7. Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*(3), 128-138.
8. Wang, L. (2016). Discovering phase transitions with unsupervised learning. *Physical Review B*, 94, 195105. https://doi.org/10.1103/PhysRevB.94.195105.
9. Hinton, G., & Sejnowski, T. (2018). Unsupervised Learning. , 1009. https://doi.org/10.1007/978-3-319-17885-1_101437.
10. He, S., Zhu, J., He, P., & Lyu, M. R. (2016, October). Experience report: System log analysis for anomaly detection. In *2016 IEEE 27th international symposium on software reliability engineering (ISSRE)* (pp. 207-218). IEEE.
11. Veeraiah, D., & Rao, J. (2020). An Efficient Data Duplication System based on Hadoop Distributed File System. *2020 International Conference on Inventive Computation Technologies (ICICT)*, 197-200. https://doi.org/10.1109/ICICT48043.2020.9112567.
12. Bui, D., Hussain, S., Huh, E., & Lee, S. (2016). Adaptive Replication Management in HDFS Based on Supervised Learning. *IEEE Transactions on Knowledge and Data Engineering*, 28, 1369-1382. https://doi.org/10.1109/TKDE.2016.2523510.
13. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, 1-10. https://doi.org/10.1109/MSST.2010.5496972.
14. Dwivedi, K., & Dubey, S. (2014). Analytical review on Hadoop Distributed file system. *2014 5th International Conference - Confluence The Next Generation Information Technology Summit (Confluence)*, 174-181. https://doi.org/10.1109/CONFLUENCE.2014.6949336.
15. Khalil, M., & Hamad, M. (2021). Big Data Management Using Hadoop. *Journal of Physics: Conference Series*, 1804. https://doi.org/10.1088/1742-6596/1804/1/012109.

16. Kousalya, K., & Parvez, S. (2018). Effective processing of unstructured data using python in Hadoop map reduce. *International journal of engineering and technology*, 7, 417. https://doi.org/10.14419/IJET.V7I2.21.12456.
17. Ridzuan, F., & Zainon, W. M. N. W. (2019). A review on data cleansing methods for big data. Procedia Computer Science, 161, 731-738.
18. J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques*. Elsevier, 2011.
19. Su, C., & Yang, C. (2008). Feature selection for the SVM: An application to hypertension diagnosis. *Expert Syst. Appl.*, 34, 754-763. https://doi.org/10.1016/j.eswa.2006.10.010.
20. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20, 273-297. https://doi.org/10.1023/A:1022627411411.
21. Fleiss, J., Williams, J., & Dubro, A. (1986). The logistic regression analysis of psychiatric data.. *Journal of psychiatric research*, 20 3, 195-209 . https://doi.org/10.1016/0022-3956(86)90003-8.
22. Larget, B. (2008). Logistic regression. *Exploring Concepts of Child Well-being*. https://doi.org/10.1007/978-0-387-79054-1_13.
23. Landwehr, N., Kersting, K., & Raedt, L. (2007). Integrating Naïve Bayes and FOIL. *J. Mach. Learn. Res.*, 8, 481-507. https://doi.org/10.5555/1314498.1314516.
24. Liu, Z., Qin, T., Guan, X., Jiang, H., & Wang, C. (2018). An Integrated Method for Anomaly Detection From Massive System Logs. *IEEE Access*, 6, 30602-30611. https://doi.org/10.1109/ACCESS.2018.2843336.
25. Huo, J., Weng, J., & Qu, H. (2019). A parallel clustering algorithm for logs data based on Hadoop platform. *Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications*. https://doi.org/10.1145/3318265.3318281.