Jurnal EL Sains *P-ISSN: 2527-6336*
Volume 5,Nomor 1, Juni 2023 *E-ISSN: 2656-7075*

Deteksi DDoS pada Unbalanced Dataset Menggunakan PCA dan Local Outlier Factor

Oky Rahmanto[1], Hendrik Setyo Utomo[2], Arif Supriyanto[3]
[1]*Prodi Teknologi Informasi, Politeknik Negeri Tanah Laut*
[2,3]*Prodi Teknologi Rekayasa Komputer dan Jaringan, Politeknik Negeri Tanah Laut*
Jl. A.Yani Km.06, Desa Panggung Tanah Laut
Telp. (0512) 2021065
[1] *oky.rahmanto@politala.ac.id*
[2]*hendrik_tomo@politala.ac.id*
[3]*arif@politala.ac.id*

ABSTRAKS
Dataset DDoS adalah salah satu data yang sering tersedia dalam bentuk tidak seimbang(Unbalanced) antara cluster data serangan dengan cluster data normal. beberapa teknik klasifikasi telah diterapkan untuk mengatasi permasalahan ini salah satunya adalah menggunakan teknik Local Outlier Factor. Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja teknik LOF dalam mendeteksi paket data yang merupakan serangan DDoS.. Sebelum dataset digunakan, dilakukan pembersihan data dan seleksi fitur menggunakan PCA. Penentuan hasil secara keseluruhan menggunakan metrik F1-Score. Nilai F1-Score terendah terdapat pada pengaturan Neighbour=12 dan Contamination=0,5 sebesar 0,619205. Nilai F1-Score tertinggi terdapat pada pengaturan Neighbour=19 dan Contamination=0,1 sebesar 0,957500 .

Kata Kunci: ddos, local outlier factor, unbalanced dataset

1. PENDAHULUAN

Serangan DDoS adalah varian dari serangan Denial-of-Service (DoS) yang memiliki karakteristik khusus. Dalam serangan ini, server korban dibanjiri dengan paket data yang dikirim oleh server atau mesin yang sebelumnya telah diretas[1]. Serangan ini melibatkan penggunaan sejumlah server atau mesin yang telah dikendalikan oleh penyerang, yang disebut juga sebagai "bot". Bot-bot ini secara bersamaan mengirimkan paket data ke target dalam periode waktu yang singkat, sehingga mengakibatkan ketidakmampuan target untuk memberikan layanan kepada pengguna yang sebenarnya.

Salah satu bentuk mesin yang dapat diretas dan digunakan sebagai bot dalam serangan DDoS adalah perangkat Internet of Things (IoT). Hal ini disebabkan oleh kemampuan perangkat IoT untuk terhubung ke jaringan dan mengirimkan paket data, serta kekurangan perlindungan dan pengendalian akses pada banyak perangkat IoT [2]. Selain itu, dengan peningkatan penggunaan perangkat IoT yang terus meningkat, ini secara tidak langsung akan menyebabkan peningkatan jenis dan volume bot yang dapat melakukan serangan.

Untuk menghadapi serangan DDoS, penting untuk mengambil langkah-langkah pencegahan dalam berbagai aspek keamanan agar layanan server tetap berjalan normal tanpa terpengaruh oleh serangan tersebut. Salah satu langkah yang diperlukan adalah menerapkan mekanisme deteksi yang dapat membedakan antara paket data normal dan paket data serangan. Salah satu pendekatan untuk memisahkan kedua jenis paket data ini adalah dengan menggunakan teknik Pattern Matching and Fingerprinting [1], yaitu metode pendeteksian yang berdasarkan pada kemiripan paket data di dalam jaringan[3]. Mekanisme ini bekerja dengan mempelajari pola serangan DDoS pada masa lampau dan menggunakan pola tersebut sebagai model untuk melakukan prediksi dan klasifikasi terhadap paket data yang baru.

Model klasifikasi ini dapat membedakan antara paket data normal dan paket data serangan DDoS dengan baik jika data paket masa lampau yang digunakan dalam pembentukan pola pada saat pelatihan model cukup dan seimbang. Namun, dalam kenyataannya, seringkali jumlah data yang tersedia untuk pelatihan tidak seimbang, di mana jumlah paket serangan DDoS jauh lebih banyak daripada paket data normal. Keadaan ini disebut sebagai *Unbalance Dataset.*

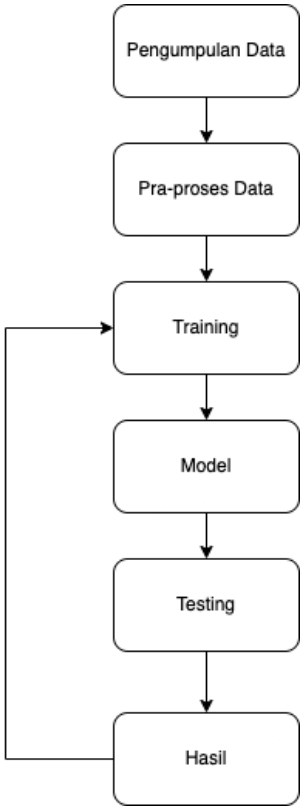
Karakteristik utama dari *Unbalance Dataset* adalah ketidakseimbangan dataset, yang disebabkan oleh fenomena bahwa beberapa jenis data yang tidak memenuhi kondisi untuk digunakan akan diabaikan[4] saat dijadikan bahan untuk pelatihan sehingga jumlah data dalam kelas Positif akan berbeda dengan jumlah data dalam kelas Negatif yang akan mengakibatkan kurangnya representasi data dalam setiap kelas. Dalam kasus dataset DDoS terjadi hal demikian pula.

P-ISSN: 2527-6336
E-ISSN: 2656-7075

Salah satu teknik pengklasifikasi yang dapat mengatasi ketidakseimbangan dataset adalah *Local Outlier Factor*.

2. METODE PENELITIAN

Berikut adalah penjabaran dari langkah-langkah penelitian yang dilakukan



Gambar 1. Alur Proses Penelitian

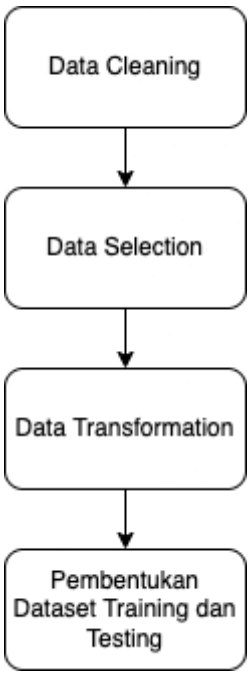
2.1 Pengumpulan data

Pada tahap ini, peneliti mendapatkan data berasal dari sumber terbuka *DDoS Botnet Attack on IOT Devices* tersedia pada https://www.kaggle.com/datasets/siddharthm1698/ddos-botnet-attack-on-iot-devices. Dataset ini berisi 47 kolom yang didalamnya beberapa propertis seperrti *source port, source address, destination port*, *destination address* serta jumlah paket data yang merupakan bagian komponen penting dalam pendeteksian serangan DDoS[5]

Pada proses training Local Outlier Factor, hanya data paket DDoS yang diperlukan untuk, tetapi data paket normal juga diperlukan untuk fase evaluasi (pengujian) dan dapat lebih sedikit dibandingkan data untuk training.

2.2 Pra-Pemrosesan data

Pra-pemrosesan data dilakukan untuk memastikan data mentah telah siap untuk dimasukkan dalam training untuk Local Outlier Factor. Tahapan yang dilalui berupa Data Cleaning, Data Integration, Data Transformation serta pembentukan dataset training dan testing



Gambar 2. Langkah pra-pemrosesan data

A. *Data Cleaning*

Proses cleaning dilakukan dengan cara melakukan pengecekan terhadap data yang memiliki instance kosong serta menghapus instance tersebut.

B. *Data Selection*

Proses ini dilakukan untuk memilih propertis dari data yang akan digunakan. Proses ini meilibatkan adanya *exploratory analysis* serta studi referensi dalam hal karakteristik paket data berjenis DDoS. Analisis dalam penelitian ini bertujuan untuk mengenal paket data yang berjenis serangan. dalam hal ini fokus pada penyerang sehingga propertis data utama yang perlu diperhatikan berupa *protocol, source ip , source port, count_pkts, CountPktsStateDestIP, CountPktsStateSrcIP[5]*

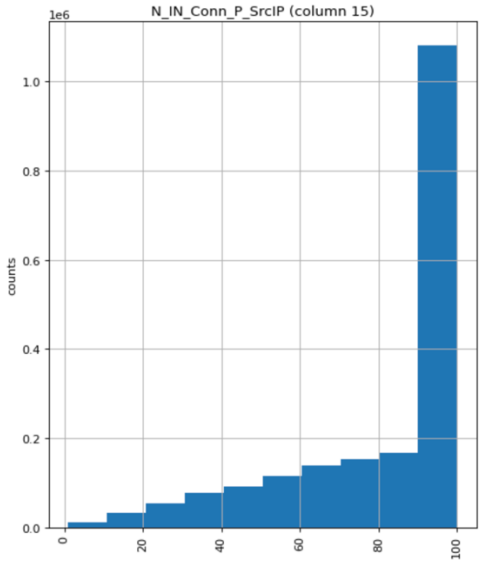
C. *Data Transformation*

Setiap properti dataset biasanya akan diubah menjadi nilai MinMax selama proses transformasi data [0,1]. Namun, tidak semua properti dapat diubah menjadi nilai MinMax dalam kasus DDoS[0,1]. Properti tertentu memiliki sifat kategorikal yang memerlukan analisis tambahan sebelum diubah menjadi nilai MinMax[0,1].

SourceIP dan sourcePort adalah contoh data yang harus dianalisis terlebih dahulu. Kedua properti ini harus mengalami proses peringkat berdasarkan frekuensi data yang muncul. Analisis eksplanatori distribusi kolom menunjukkan bahwa ada lima jenis sumber IP dengan frekuensi yang cukup besar.

P-ISSN: 2527-6336
E-ISSN: 2656-7075

Selain itu juga terlihat jumlah koneksi per IP address terlihat sangat signifikan untuk 1 buah IP address.



Gambar 2. *Exploratory Analysis* menggunakan *Column Distribution* untuk Jumlah Koneksi Per IP Address

Selanjutnya dilakukan normalisasi MinMax[0,1] untuk setiap propertis data. Untuk data kategori protocol, normaliasi dilakukan dengan memberikan label [0,1] dikarenakan hanya terdapat 2 buah data yaitu [TCP,UDP]. Pada tahap ini juga dilakukan pengurangan dimensi dari dataset. Teknik PCA di terapkan pada dataset untuk mengurangi dimensi serta ketergantungan antara propertis data.

D. Pembentukan dataset training dan testing

Dataset yang diperlukan dalam One Class Classification harus tidak seimbang, dalam hal ini untuk training menggunakan 41700 data DDoS serta 0 data normal. Untuk testing digunakan 417 data DDoS dan 417 data normal

Tabel 1. Distribusi data training dan testing

Jenis Data	*Training*	*Testing*
DDoS	41700	417
Normal	0	417

2.3 Principal Component Analysis

Dalam bidang pengenalan pola dan visi komputer (seperti pengenalan wajah), analisis komponen utama (Principal Component Analysis/PCA) adalah teknik ekstraksi fitur klasik dan representasi data. Ini juga dikenal sebagai ekspansi Karhunen-Loeve[6]. PCA digunakan untuk mengekstrak informasi penting dari pengamatan tersebut dan mengurangi noise. Untuk mencapai tujuan ini, PCA menghitung seperangkat variabel baru yang saling ortogonal yang disebut komponen utama, yang diperoleh sebagai kombinasi linear dari variabel asli. Nilai variabel baru ini untuk pengamatan disebut skor faktor. Skor faktor ini dapat diinterpretasikan sebagai proyeksi pengamatan ke komponen utama. PCA juga dapat diformulasikan sebagai proyeksi linear yang meminimalkan biaya proyeksi rata-rata yang didefinisikan sebagai jarak kuadrat rata-rata antara titik data dan proyeksinya.[7].

Untuk penelitian ini, PCA membantu proses pelatihan dapat berjalan lebih cepat dan tepat dengan mengurangi dimensi dari dataset serta ketergantungan antar propertis dari dataset tersebut.

2.4 Local Outlier Factor

Deteksi outlier adalah prosedur statistik yang bertujuan untuk menemukan kejadian atau data yang mencurigakan dan berbeda dari bentuk normal suatu dataset. Hal ini menarik minat yang cukup besar dalam bidang data mining dan machine learning. Deteksi outlier penting karena digunakan untuk deteksi penipuan dalam transaksi kartu kredit dan deteksi intrusi jaringan. Terdapat dua jenis umum dari deteksi outlier: global dan lokal. Outlier global berada di luar rentang normal untuk keseluruhan dataset, sedangkan outlier local berada dalam rentang normal untuk keseluruhan dataset, tetapi di luar rentang normal untuk titik data sekitarnya[8].

Local Outlier Factor (LOF) adalah algoritma deteksi outlier berbasis kepadatan yang menemukan outlier dengan menghitung deviasi lokal dari suatu titik data yang diberikan, yang cocok untuk deteksi outlier pada dataset dengan distribusi yang tidak merata. Penentuan outlier dinilai berdasarkan kepadatan antara setiap titik data dan titik tetangganya. Semakin rendah kepadatan titik tersebut, semakin mungkin ia diidentifikasi sebagai outlier[9].

3. HASIL DAN PEMBAHASAN

Metrik kinerja sangat membantu dalam membandingkan kualitas prediksi dari teknik-teknik pengklasifikasian. *accuracy*, *precision*, *recall*, *F1 score*, dan indeks Jaccard adalah beberapa metrik yang umum digunakan untuk klasifikasi biner.[10]

Untuk evaluasi dari penggunakan teknik LOF dalam melakukan pelatihan maupun pengujian digunakankanlah metrik *F1-Score*. *F1-Score* adalah ukuran yang menggabungkan *precision* dan recall untuk memberikan evaluasi yang lebih holistik tentang performa model klasifikasi. F1-Score diperoleh dengan menghitung *harmonic mean* (rata-rata harmonik) antara *precision*, dan *recall*, yang memberikan bobot yang seimbang antara keduanya. *Precision* mengukur sejauh mana model memberikan hasil positif yang benar, sedangkan recall mengukur sejauh mana model dapat mendeteksi secara akurat instance positif yang ada dalam dataset. Nilai F1-Score berkisar antara 0

hingga 1, dimana 1 menunjukkan kinerja model yang sempurna dengan *precision,* dan *recall* yang tinggi, sementara 0 menunjukkan kinerja yang sangat buruk. Pada teknik LOF, terdapat parameter yang digunakan untuk menentukan hasil model dari pelatihan berupa *neighbors* dan *contamination*. Nilai *neighbors* yang diujicoba untuk pelatihan ini berkisar dari 1 sampai 20 dan nilai *contamination* berkisar dari 0,0001 sampai 0,5. Dari hasil percobaan, Nilai F1-Score terendah terdapat pada pengaturan *neighbors*=12 dan *contamination* =0,5 sebesar 0,619205. Nilai F1-Score tertinggi terdapat pada pengaturan *neighbors* =19 dan *contamination* =0,1 sebesar 0,957500 .

Tabel 2. Hasil Pengujian *F1-Score*

n \ *c*	*0,0001*	*0,001*	*0.01*	*0.1*	*0.5*
1	0.674230	0.668273	0.695359	0.760649	0.789326
2	0.667734	0.693267	0.679835	0.711703	0.699844
3	0.676399	0.671498	0.681555	0.784730	0.649600
4	0.668269	0.670427	0.693467	0.873432	0.659164
5	0.667734	0.679184	0.732568	0.898113	0.623762
6	0.680261	0.685855	0.784091	0.931990	0.712963
7	0.677498	0.711604	0.810970	0.909765	0.648298
8	0.679707	0.699748	0.789675	0.940000	0.637255
9	0.685855	0.696234	0.830000	0.934866	0.657005
10	0.690969	0.704392	0.848610	0.915935	0.654839
11	0.686985	0.702020	0.848671	0.948814	0.628289
12	0.680261	0.703297	0.857438	0.948686	**0.619205**
13	0.684167	0.715266	0.836879	0.948686	0.652666
14	0.690397	0.705584	0.789724	0.943110	0.654839
15	0.692116	0.703204	0.826479	0.946970	0.628289
16	0.687552	0.718966	0.813425	0.945638	0.661316
17	0.688119	0.702020	0.840404	0.933842	0.671975
18	0.679153	0.696742	0.836548	0.948298	0.650485
19	0.688687	0.696234	0.844490	**0.957500**	0.654839
20	0.680294	0.708581	0.831325	0.950943	0.676190

4. KESIMPULAN

Secara umum, teknik Local Outlier Factor dapat digunakan untuk mendeteksi serangan DDoS, dengan menggunakan pola dari paket data serangan DDoS sebelumnya. Beberapa penelitian sebelumnya menyebutkan bahwa teknik Local Outlier Factor memiliki *time complexity* yang tinggi sehingga Penggunaan PCA dapat membantu agar teknik Local Outlier Factor yang memiliki *time complexity* yang tinggi agar menjadi lebih maksimal.

PUSTAKA

[1] N. Agrawal dan S. Tapaswi, "Defense Mechanisms Against DDoS Attacks in a Cloud Computing Environment: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, hlm. 3769–3795, 2019, doi: 10.1109/COMST.2019.2934468.

[2] X. Zhang, O. Upton, N. L. Beebe, dan K.-K. R. Choo, "IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers," *Forensic Science International: Digital Investigation*, vol. 32, hlm. 300926, Apr 2020, doi: 10.1016/j.fsidi.2020.300926.

[3] A. Chonka, J. Singh, dan W. Zhou, "Chaos theory based detection against network mimicking DDoS attacks," *IEEE Communications Letters*, vol. 13, no. 9, hlm. 717–719, Sep 2009, doi: 10.1109/LCOMM.2009.090615.

[4] L. Wang, M. Han, X. Li, N. Zhang, dan H. Cheng, "Review of Classification Methods on Unbalanced Data Sets," *IEEE Access*, vol. 9, hlm. 64606–64628, 2021, doi: 10.1109/ACCESS.2021.3074243.

[5] O. Rahmanto, "Deteksi DDoS dengan One Class Classification," *JI-Tech*, vol. 18, no. 1, hlm. 32–37, 2022.

[6] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, dan A. Hooman, "An Overview of Principal Component

P-ISSN: 2527-6336
E-ISSN: 2656-7075

Analysis," *JSIP*, vol. 04, no. 03, hlm. 173–175, 2013, doi: 10.4236/jsip.2013.43B031.

[7] T. Kurita, "Principal Component Analysis (PCA)," dalam *Computer Vision: A Reference Guide*, Cham: Springer International Publishing, 2019, hlm. 1–4. doi: 10.1007/978-3-030-03243-2_649-1.

[8] O. Alghushairy, R. Alsini, T. Soule, dan X. Ma, "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams," *Big Data and Cognitive Computing*, vol. 5, no. 1, Art. no. 1, Mar 2021, doi: 10.3390/bdcc5010001.

[9] Z. Cheng, C. Zou, dan J. Dong, "Outlier detection using isolation forest and local outlier factor," dalam *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, Chongqing China: ACM, Sep 2019, hlm. 161–168. doi: 10.1145/3338840.3355641.

[10] Z. C. Lipton, C. Elkan, dan B. Narayanaswamy, "Thresholding Classifiers to Maximize F1 Score." arXiv, 13 Mei 2014. doi: 10.48550/arXiv.1402.1892.

P-ISSN: 2527-6336
E-ISSN: 2656-7075

Halaman ini sengaja dikosongkan