PERBANDINGAN KINERJA ALGORITMA K-NEAREST NEIGHBORS (K-NN) DAN DECISION TREE DALAM DETEKSI PAKET MALIS PADA JARINGAN

Bib Nugraha Kasmara[1], Endah Tri Esti Handayani[2], Novi Dian Nathasia[3]
Informatika, Universitas Nasional[1,2,3]
bibnk27@gmail.com[1]

Submitted February 11, 2024; Revised March 13, 2024; Accepted March 13, 2024

Abstrak

Penelitian ini bertujuan untuk melakukan klasifikasi terhadap data paket malis dan membandingkan performa dua algoritma, yaitu K-Nearest Neighbor (K-NN) dan Decision Tree (DT). Dataset UNSW-NB15 yang digunakan untuk penelitian ini telah melalui tahap preprocessing, feature selection, dan data split. Tahap preprocessing termasuk transformasi data dan pemilihan fitur yang relevan untuk mendeteksi paket malis. Selanjutnya, eksperimen dilakukan untuk menguji variasi nilai K pada K-NN dan mengukur akurasi, recall, precision, dan F1-Score. Hasilnya menunjukkan bahwa K-NN memiliki akurasi 91.54%, sedangkan DT memiliki 92.41%. Kesimpulan dari penelitian ini menunjukkan bahwa algoritma Decision Tree (DT) memiliki kinerja yang sedikit lebih baik daripada K-Nearest Neighbor (K-NN) dalam mendeteksi paket malis. Oleh karena itu, dalam memilih algoritma untuk deteksi keamanan jaringan, penting untuk mempertimbangkan kebutuhan dan tujuan spesifik penelitian serta karakteristik data yang digunakan.
Kata Kunci*:* KNN, Decision Tree, Paket Malis

Abstract

This research aims to classify malicious packet data and compare the performance of two algorithms, namely K-Nearest Neighbor (K-NN) and Decision Tree (DT). The UNSW-NB15 dataset used in this study has undergone preprocessing, feature selection, and data split stages. The preprocessing stage includes data transformation and selection of relevant features to detect malicious packets. Subsequently, experiments were conducted to test various values of K in K-NN and measure accuracy, recall, precision, and F1-Score. The results show that K-NN has an accuracy of 91.54%, while DT has 92.41%. The conclusion of this research indicates that the Decision Tree (DT) algorithm performs slightly better than K-Nearest Neighbor (K-NN) in detecting malicious packets. Therefore, in selecting an algorithm for network security detection, it is important to consider the specific needs and goals of the research as well as the characteristics of the data used.
Keywords: *KNN, Decision Tree, Accuracy, Malicious Packets*

1. PENDAHULUAN

Tanpa disadari, masyarakat semakin terintegrasi dengan teknologi informasi [1]. Jaringan komputer dan sistem digital telah menjadi tulang punggung sebagian besar aktivitas, baik dalam skala pribadi maupun organisasi [2]. Seiring dengan perkembangan tersebut, kompleksitas ancaman terhadap keamanan siber juga semakin meningkat, termasuk serangan melalui paket berbahaya yang dapat

membahayakan integritas dan ketersediaan data [3]. Dalam upaya melindungi kelangsungan fungsi jaringan dan menjaga kepercayaan pengguna terhadap sistem digital, penting untuk mengembangkan solusi yang efektif dan responsive [4].

KNN adalah salah satu algoritma klasifikasi yang sederhana dan efisien [5], sementara DT adalah algoritma pembelajaran mesin yang digunakan untuk tugas regresi dan klasifikasi [6]. Decision tree terdiri dari akar (root), brach node, dan leaf node . Prosesnya dilakukan secara rekursif dari root hingga leaf node untuk memecah proses pengambilan keputusan menjadi lebih sederhana [7]. Studi sebelumnya telah menggunakan DT untuk sistem deteksi intrusi dengan tingkat akurasi mencapai 99,15% dan menggunakan data KDDCUP'99 sebagai datasetnya [8]. Dataset KDDCUP'99 memberikan pemahaman terbaik tentang berbagai serangan intrusi adalah dataset KDD. Dataset KDDCUP'99 merupakan salah satu dataset yang tersedia untuk Sistem Deteksi Intrusi jaringan, namun memiliki masalah utama [9].

Penelitian ini berfokus pada perbandingan dua metode klasifikasi yaitu algoritma K-Nearest Neighbor (KNN) dan Decision Tree dalam mendeteksi paket malis menggunakan dataset UNSW-NB15. UNS-NB15 adalah sebuah dataset yang dikembangkan oleh University of New South Wales (UNSW), Australia, untuk penelitian keamanan jaringan. Dataset ini berisi berbagai jenis serangan siber yang direkam dalam lingkungan jaringan terkontrol, seperti serangan DoS, DDoS, probing, dan brute-force[10]. Kedua algoritma tersebut merupakan pendekatan yang umum digunakan dalam analisis keamanan jaringan [11]. Oleh karena itu, penelitian ini berupaya mengidentifikasi metode terbaik dalam mengenali ancaman[12]. Hasil penelitian ini diharapkan dapat memberikan panduan bagi para profesional dan peneliti keamanan siber untuk memilih strategi deteksi yang paling efektif dan selaras dengan kebutuhan spesifik lingkungan mereka [13].

Dalam penelitian Okki Setyawan, Angge Firizkiansah, Ahmad Nuryanto, pada penelitian yang berjudul "*Klasifikasi Tingkat Keparahan Serangan Jaringan Komputer Dengan Metode Machine Learning"*, Jaringan komputer berkembang pesat dan keamanannya penting. Penelitian ini menggunakan rekaman data perusahaan untuk mengevaluasi keparahan serangan yang dideteksi oleh firewall. Metode machine learning yang digunakan adalah K-Nearest Neighbours dan Decision Tree. Dataset terdiri dari 5999 entri log firewall dengan 23 fitur. Hasil penelitian menunjukkan akurasi 100% untuk kedua metode. Dengan demikian, machine learning dapat digunakan untuk mengklasifikasikan tingkat keparahan serangan jaringan computer [14].

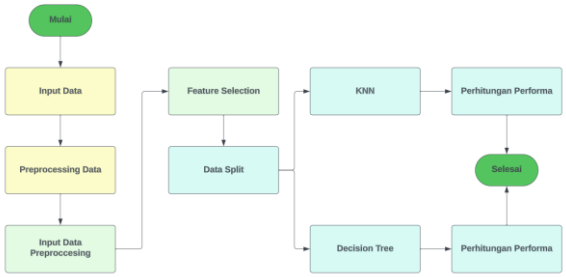
Selain itu, dalam Penelitian yang diterbitkan oleh Rasi Nuraeni, Aso Sudiarjo, Randi Rizal berjudul "*Perbandingan Algoritma Naïve Bayes Classifier Dan Algoritma Decision Tree Untuk Analisa Sistem Klasifikasi Judul Skripsi*". Studi ini membandingkan algoritma Naive Bayes Classifier dan Decision Tree dalam mengklasifikasikan judul tesis di Program Studi Teknik Informatika Universitas Perjuangan Tasikmalaya. Data diperoleh melalui studi literatur dan dianalisis menggunakan rapidminer. Hasilnya menunjukkan akurasi Naive Bayes sebesar 80,33% dan Decision Tree sebesar 60,33% dari 55 judul tesis dengan 3 kategori [15].

Tujuan dari penelitian ini dibuat adalah untuk menguji akurasi yang digunakan untuk Menilai kinerja algoritma K-Nearest Neighbor (K-NN) dan Decision Tree berdasarkan metrik evaluasi seperti akurasi, recall, precision, dan F1-score dan

menentukan algoritma mana yang memiliki akurasi lebih baik dari kedua algoritma tersebut dalam mendeteksi paket malis.

2. METODE PENELITIAN

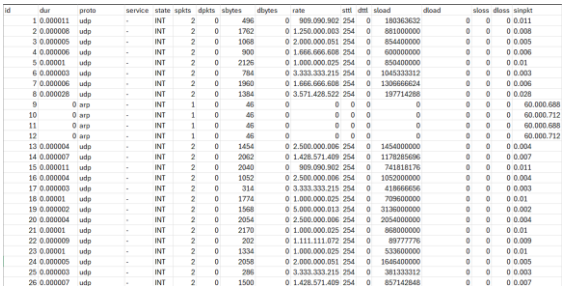
Penelitian ini menggunakan metodologi yang terdiri dari beberapa tahapan seperti, Pada gambar alur metode penelitian ini pada Gambar berikut.



Gambar 1. Metode Penelitian

Input Data

Dataset yang digunakan pada penelitian ini yaitu dataset UNSW_NB15, sebuah dataset yang menyajikan data normal dan data yang mengandung serangan jaringan. Dataset ini diperoleh dari Kaggle dataset library dengan judul UNSW_NB15.



id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	dload	sloss	dloss	sinpkt
1	0.000011	udp	-	INT	2	0	496	0	909.090.902	254	0	180363632		0	0	0.011
2	0.000008	udp	-	INT	2	0	1762	0	1.250.000.003	254	0	881000000		0	0	0.008
3	0.000005	udp	-	INT	2	0	1068	0	2.000.000.051	254	0	854400000		0	0	0.005
4	0.000006	udp	-	INT	2	0	900	0	1.666.666.608	254	0	600000000		0	0	0.006
5	0.00001	udp	-	INT	2	0	2126	0	1.000.000.025	254	0	850400000		0	0	0.01
6	0.000003	udp	-	INT	2	0	784	0	3.333.333.215	254	0	1045333312		0	0	0.003
7	0.000006	udp	-	INT	2	0	1960	0	1.666.666.608	254	0	1306666624		0	0	0.006
8	0.000028	udp	-	INT	2	0	1384	0	3.571.428.522	254	0	197714288		0	0	0.028
9	0	arp	-	INT	1	0	46	0	0	0	0	0		0	0	0	60.000.688
10	0	arp	-	INT	1	0	46	0	0	0	0	0		0	0	0	60.000.712
11	0	arp	-	INT	1	0	46	0	0	0	0	0		0	0	0	60.000.688
12	0	arp	-	INT	1	0	46	0	0	0	0	0		0	0	0	60.000.712
13	0.000004	udp	-	INT	2	0	1454	0	2.500.000.006	254	0	1454000000		0	0	0.004
14	0.000007	udp	-	INT	2	0	2062	0	1.428.571.409	254	0	1178285696		0	0	0.007
15	0.000011	udp	-	INT	2	0	2040	0	909.090.902	254	0	741618176		0	0	0.011
16	0.000004	udp	-	INT	2	0	1052	0	2.500.000.006	254	0	1052000000		0	0	0.004
17	0.000003	udp	-	INT	2	0	314	0	3.333.333.215	254	0	418666656		0	0	0.003
18	0.00001	udp	-	INT	2	0	1774	0	1.000.000.025	254	0	709600000		0	0	0.01
19	0.000002	udp	-	INT	2	0	1568	0	5.000.000.013	254	0	3136000000		0	0	0.002
20	0.000004	udp	-	INT	2	0	2054	0	2.500.000.006	254	0	2054000000		0	0	0.004
21	0.00001	udp	-	INT	2	0	2170	0	1.000.000.025	254	0	868000000		0	0	0.01
22	0.000009	udp	-	INT	2	0	202	0	1.111.111.072	254	0	89777776		0	0	0.009
23	0.00001	udp	-	INT	2	0	1334	0	1.000.000.025	254	0	533600000		0	0	0.01
24	0.000005	udp	-	INT	2	0	2058	0	2.000.000.051	254	0	1646400000		0	0	0.005
25	0.000003	udp	-	INT	2	0	286	0	3.333.333.215	254	0	381333312		0	0	0.003
26	0.000007	udp	-	INT	2	0	1500	0	1.428.571.409	254	0	857142848		0	0	0.007

Gambar 2. Dataset UNSW_NB15

Data tersebut memiki 45 atribut. Berikut beberapa atribut-atribut yang ada pada dataset UNSW_NB15:

Tabel 1. Atribut Dataset UNSW_NB15

No.	Nama Atribut	Deskripsi
1	id	Nomor identifikasi unik untuk setiap rekaman.
2	dur	Durasi total rekaman.
3	proto	Protokol transaksi.
4	service	Layanan yang terkait dengan transaksi (contoh: http, ftp, smtp).
5	state	Menunjukkan keadaan dan protokol terkait (misalnya, ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST).
6	spkts	Jumlah paket dari sumber ke tujuan.
7	dpkts	Jumlah paket dari tujuan ke sumber.
8	sbytes	Byte transaksi dari sumber ke tujuan.
9	dbytes	Byte transaksi dari tujuan ke sumber.

Berdasarkan fitur-fitur yang ada pada data tersebut, seseorang dapat diklasifikasikan menjadi dua kelas target "*classification*", yaitu data normal dan data yang mengandung serangan jaringan.

Preprocessing Data

Tahap Pada tahapain ini, dataset dilakukan serangkaian langkah untuk menyiapkan data yang diperlukan sebelum digunakan dalam pemodelan menggunakan algoritma K-Nearest Neighbors (K-NN) dan Decision Tree untuk deteksi paket malis pada jaringan. Pertama memisahkan data *features*, *non-numeric*, *numeric_feature* dan *non log*. Lalu melakukan transformasi log pada data yang sudah dipisahkan untuk menormalkan distribusi data. Setelah itu, fitur-fitur tersebut digabungkan kembali dengan fitur non-numerik yang telah diidentifikasi sebelumnya.

Selanjutnya, dilakukan perhitungan skor Mutual Information (MI) untuk masing-masing fitur terhadap variabel target, yaitu label deteksi paket. Fitur-fitur yang memiliki MI *score* di atas suatu ambang

batas (misalnya 0.2) dipilih untuk digunakan dalam pemodelan. Data yang telah melalui tahap pemilihan fitur tersebut disimpan untuk digunakan pada tahap selanjutnya. Berikut adalah atribut-atribut yang relevan setelah dilakukannya preprocessing data:

Tabel 2. Atribut Relevan

Atribut Relevan	
sbytes	dttl
smean	dinpkt
sload	sttl
dbytes	dload
ct_state_ttl	dpkts
rate	tcprtt
dur	synack
dmean	label

Feature Selection

Pada tahap feature selection, dilakukan metode cross-validation K-Fold diterapkan sepuluh kali, di mana data pelatihan dipecah menjadi bagian training dan testing. Pada fase pelatihan, model K-Nearest Neighbors (K-NN) dilatih, dan metrik akurasi digunakan untuk mengevaluasi kinerja model, dengan hasilnya dicatat. Proses ini diulangi untuk berbagai nilai K guna memperoleh nilai K yang optimal.

Data Split

Sebelum tahap analisis algoritma K-Nearest Neighbors (K-NN) dan Decision Tree (DT), pembagian data dilakukan untuk membagi dataset menjadi dua set utama: set pelatihan (train) dan set pengujian. Proses ini penting untuk menguji dan memvalidasi kinerja algoritma pada data yang tidak pernah dilihat sebelumnya, sehingga memastikan generalisasi yang baik. Set pelatihan digunakan untuk melatih model dan menyesuaikannya dengan pola yang ada dalam data, sehingga Penelitian dapat memberikan evaluasi yang obyektif terhadap kemampuan K-NN dan DT dalam deteksi paket malis pada jaringan dengan melakukan split data sebelum analisis algoritma.

KNN

Analisis KNN dilakukan dengan menerapkan metode cross-validation K-Fold sebanyak sepuluh kali. Selanjutnya, dilakukan standarisasi data menggunakan kode program "*StandardScaler*". Setelah itu, model dilatih menggunakan neighbor yang telah terpilih dari tahap *feature selection*.

Tabel 3. Parameter KNN

Parameter	
Nama Parameter	**Nilai**
Neighbor	7

Decision Tree

Konfigurasi parameter yang digunakan pada algoritma Decision Tree adalah criterion dan max_depth. Tabel 6 menunjukan parameter Decisio Tree yang digunakan.

Tabel 4. Parameter Decision Tree

Parameter	
Nama Parameter	**Nilai**
Criterion	entropy
max_depth	4

Perhitungan Peforma

Pada tahapan ini dilakukan proses perhitungan performa dari tahapan testing di setiap algoritma. Metric performa yang digunakan adalah accuracy, precision, recall, f1 score, dan waktu eksekusi proses training dan testing. Setiap algoritma dengan parameternya masing-masing dihitung performanya di setiap cross validation dan kemudian dihitung rata-ratanya. Parameter disebuah algoritma yang memiliki performa terbaik akan dipilih dan kemudian dibandingkan dengan algoritma

yang lain yang memiliki parameter dengan performa yang terbaik.

Algoritma Penelitian

Berikut adalah langkah-langkah dalam algoritma KNN dan Decision Tree yaitu:

1. Pengumpulan data dilakukan dengan mengambil data dari kaggle.
2. Preprocessing data dilakukan untuk menyiapkan atribut yang relevan sebelum pelatihan model. Ini melibatkan pemisahan fitur data, transformasi log untuk normalisasi distribusi, dan penggabungan kembali fitur-fitur yang telah diidentifikasi.
3. Melakukan preprocessing terhadap data, seperti pemisahan data features, transformasi log, dan penggabungan kembali fitur-fitur. Selanjutnya, dilakukan perhitungan skor Mutual Information (MI) untuk pemilihan fitur dengan MI score di atas ambang batas (misalnya 0.2). Data hasil seleksi fitur disimpan untuk penggunaan tahap berikutnya.
4. Menginput data yang sudah dipreprocessing.
5. Melakukan feature selection dengan cara dataset dibagi menjadi train (80%) dan test (20%). Dilakukan metode cross-validation K-Fold sebanyak sepuluh kali, di mana data pelatihan dibagi lagi menjadi training dan testing. Model K-Nearest Neighbors (K-NN) dilatih pada bagian pelatihan. Evaluasi kinerja model menggunakan metrik akurasi, dicatat hasilnya. Proses ini diulangi untuk berbagai nilai K.
6. Melakukan split data menjadi menjadi train (80%) dan test (20%).
7. Melakukan analisis pada algoritma knn dengan cara menerapkan cross-validation K-Fold sepuluh kali, dilanjutkan dengan standarisasi data menggunakan "*StandardScaler*", dan model dilatih dengan neighbor yang terpilih dari tahap *feature selection*.
8. Melakukan analisis pada algoritma decision tree dengan cara menerapkan cross-validation K-Fold sepuluh kali, dilanjutkan dengan memakai parameter criterion.
9. Melakukan prediksi menggunakan model terlatih.
10. Membuat laporan klasifikasi yang berisi accuracy, precision, recall, f1-score dan heatmap confusion matriks.
11. Membuat visualisasi grafik yang menunjukkan nilai entropy untuk setiap atribut dalam dataset.
12. Membuat 4 node terbaik algoritma decision tree menggunakan parameter criterion dan max_depth.

Bahasa pemgrograman yang digunakan adalah Python, menggunakan software Google Collaboratory atau Google Colab. Serta lokasi penelitian ini dilakukan di daerah kediaman peneliti.

3. HASIL DAN PEMBAHASAN

Penelitian ini memulai langkah analisis dengan menjelaskan karakteristik dataset yang digunakan. UNSW-NB15 adalah dataset yang menjadi subjek penelitian karena berisi informasi tentang paket jaringan, termasuk klasifikasi apakah paket tersebut termasuk dalam kategori malis atau tidak.

Data Setelah Preprocessing

Pada tahap ini, dataset UNSW-NB15 melalui berbagai proses preprocessing untuk memastikan kualitas data dan integritasnya. Nilai yang hilang, transformasi fitur, penghapusan data tidak relevan, dan penanganan outliers dilakukan. Pertama, nilai yang hilang diatasi menggunakan strategi pengisian nilai yang sesuai. Kemudian, transformasi fitur diterapkan untuk beberapa variabel untuk

memenuhi kebutuhan analisis. Didasarkan pada standar tertentu, data tidak relevan dihapus, dan outliers ditemukan dan ditangani untuk mengurangi efeknya. Oleh karena itu, setelah preprocessing, dataset yang dihasilkan menjadi dasar yang bersih dan terstruktur, siap untuk digunakan dalam analisis lebih lanjut yang berkaitan dengan deteksi paket malis. Berikut adalah script yang digunakan untuk transformasi data:

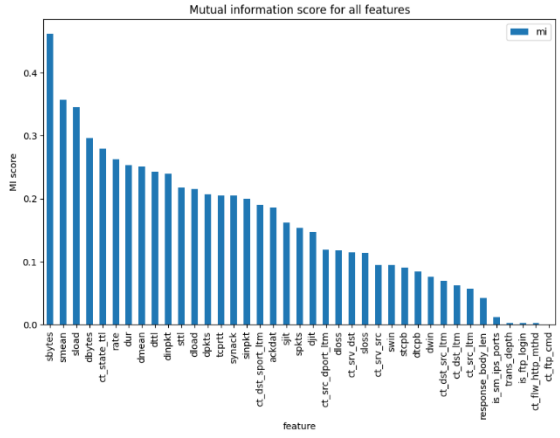
```
features = ['dur', 'spkts', 'dpkts', 'sbytes', 'dbytes', 'rate', 'sttl', 'dttl', 'sload',
    'dload', 'sloss', 'dloss', 'sinpkt', 'dinpkt', 'sjit', 'djit', 'swin', 'stcpb', 'dtcpb',
    'dwin', 'tcprtt', 'synack', 'ackdat', 'smean', 'dmean', 'trans_depth', 'response_body_len',
    'ct_srv_src', 'ct_state_ttl', 'ct_dst_ltm', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_dst_src_ltm',
    'is_ftp_login', 'ct_ftp_cmd', 'ct_flw_http_mthd', 'ct_src_ltm', 'ct_srv_dst', 'is_sm_ips_ports']


#DATA YANG TIDAK COCOK UNTUK KNN / DATA TIDAK COCOK KARENA METRIK JARAK TIDAK TERDEFINISI DENGAN BAIK
non_numeric = ['is_sm_ips_ports', 'is_ftp_login']

#numeric_features = features \ non_numeric
numeric_features = list(set(features) - set(non_numeric))

#non log: fitur numerik yang tidak perlu transformasi log
non_log = ['sttl', 'dttl', 'swin', 'dwin', 'trans_depth', 'ct_state_ttl', 'ct_flw_http_mthd']
```

```
# melakukan transformasi log dari semua fitur numerik yang diidentifikasi sebelumnya membutuhkan transformasi
df_logs = np.log10(df[list(set(fitur_numerik) - set(non_log))] + 1)

# menggabungkan fitur numerik yang telah di-log dan fitur numerik yang tidak di-log menjadi satu dataframe untuk menangani pencilan (outliers)
df_numerik = pd.concat([df_logs, df[non_log]], axis=1)

# menggabungkan semua fitur menjadi satu dataframe yang telah diubah
df_transformed = pd.concat([df_numerik, df[non_numeric]], axis=1)[fitur]
```

```
mi_cutoff = 0.2
(pd
    .concat([df_transformed[df_mi.feature[df_mi.mi > mi_cutoff]], df.label], axis=1)
    .to_csv('/content/drive/MyDrive/SKRIP-MALIS/preprocessed.csv', index=False))
```

Gambar 3. Preprocessing Data

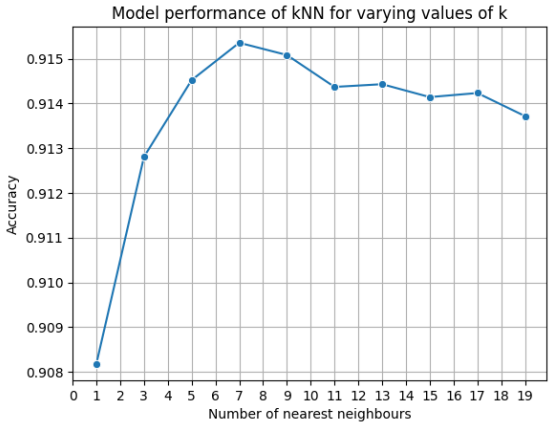
Berdasarkan script tersebut dilakukan pemisahan data feature, non numeric dan non log. Selanjutnya melakukan transformasi log dan menggabungkan semua feature menjadi satu dataframe yang diubah. Kemudian membuang nilai mutual informasion yang dibawah 0.2. Data disimpan menjadi data baru yang relevan untuk melanjutkan Langkah berikutnya. Berikut adalah data hasil dari preprocessing:



Gambar 4. Data Hasil Preprocessing

Hasil Feature Selection

Setelah proses preprocessing, tahap feature selection dilakukan untuk menentukan subset fitur yang paling relevan dalam konteks deteksi paket malis. Langkah awal melibatkan perhitungan nilai Mutual Information (MI) untuk setiap fitur terhadap label kelas pada dataset yang telah di-preprocess. Fitur-fitur yang memiliki nilai MI di atas ambang batas tertentu dipilih sebagai subset fitur yang signifikan. Selanjutnya, dilakukan penentuan nilai terbaik untuk parameter k dalam algoritma K-Nearest Neighbors (KNN). Eksperimen dilakukan dengan menggunakan variasi nilai k, yaitu 1, 3, 5, 7, 9, 11, 13, 15, 17, 19. Hasil eksperimen menunjukkan pada Gambar 3.3 bahwa kinerja model KNN mencapai akurasi tertinggi adalah nilai k=7 dengan akurasi 91.55%. Oleh karena itu, nilai k=7 dipilih sebagai parameter optimal untuk implementasi selanjutnya pada algoritma KNN.



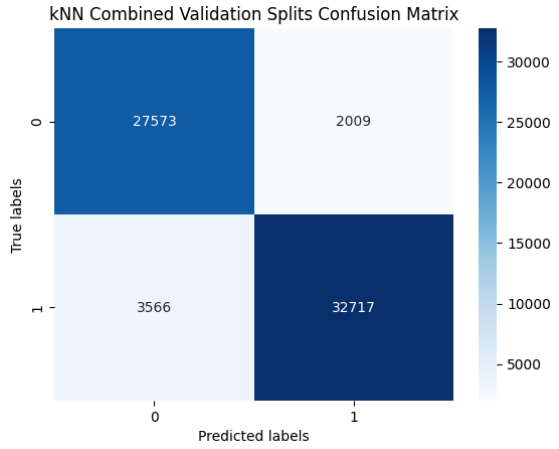
Gambar 5. Hasil Feature Selection

Analisis K-Nearest Neighbors

Hasil evaluasi klasifikasi K-NN dengan nilai k terbaik (k=7) menunjukkan tingkat akurasi sebesar 91.54%. Tingkat akurasi ini menunjukkan seberapa baik model mampu memprediksi dengan benar dataset pengujian secara keseluruhan. Nilai recall sebesar 90.17% dan nilai ketepatan sebesar 94.21% menunjukkan kemampuan model untuk menemukan paket malis sebanyak mungkin.

p-ISSN: 2527 - 9661
e-ISSN: 2549 - 2837

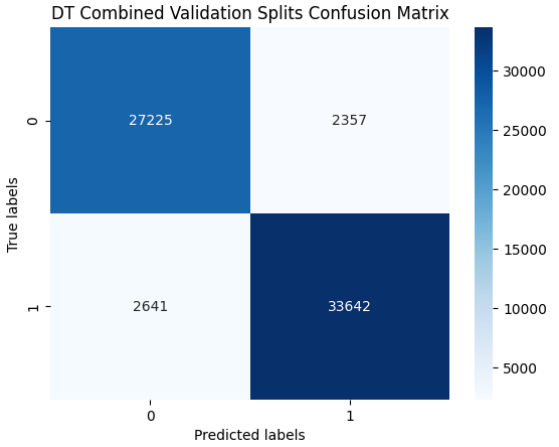
F1-Score, yang menyatukan informasi recall dan precision, memiliki nilai 92.15%. Ini adalah metrik yang relevan untuk mengukur seberapa baik sebuah model berhasil menghadapi trade-off antara recall dan precision.



Gambar 6. Confusion Matrix KNN

Analisis Decision Tree

Hasil evaluasi klasifikasi Decision Tree menggunakan beberapa metrik evaluasi yang biasa digunakan. Metrik seperti akurasi, recall, precision, dan F1-Score memberikan informasi berbeda tentang bagaimana model bekerja. Hasil evaluasi menunjukkan bahwa model Decision Tree mampu mencapai akurasi sebesar 92.41%, yang mencakup kemampuan model untuk mengklasifikasikan secara benar. Nilai pengembalian sebesar 92.72% menunjukkan kemampuan model untuk menemukan sebagian besar paket malis secara efektif, dan nilai akurasi sebesar 93.46% menunjukkan seberapa presisi model dalam mengklasifikasikan paket sebagai malis. Skor F1-nya, yang menggabungkan pengembalian dan akurasi, menunjukkan bahwa model Decision Tree mampu Analisis ini meningkatkan pemahaman kami tentang kehandalan model Decision Tree untuk melakukan klasifikasi pada dataset paket malis.



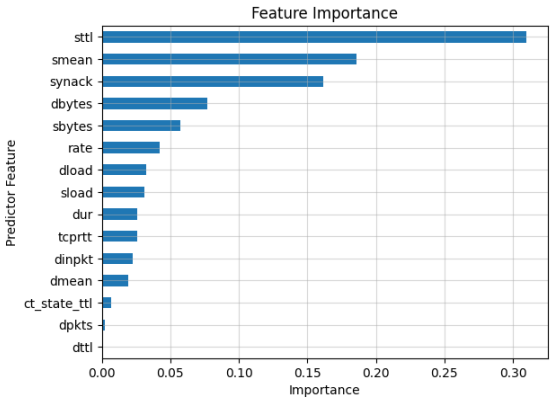
Gambar 7. Confusion Matrix Decision Tree

Visualisasi Feature Importance pada Decision Tree

Visualisasi ini memberikan gambaran jelas tentang karakteristik terpenting dalam pengklasifikasian paket malis oleh Decision Tree. Hal ini penting untuk pemahaman lebih lanjut tentang komponen utama dalam deteksi paket malis dengan algoritma Decision Tree.

```
importance = pd.Series(DT.feature_importances_, index=predictors)
importance = importance.groupby(level=0).mean()
importance.nlargest(len(predictors)).plot(kind='barh').invert_yaxis()
plt.grid(alpha=0.5)
plt.title('Feature Importance')
plt.xlabel('Importance')
plt.ylabel('Predictor Feature')
plt.savefig('/content/drive/MyDrive/SKRIP-MALIS/images/importance.png')
plt.show()
plt.close()
```

Gambar 8. Visualisasi Feature Importance

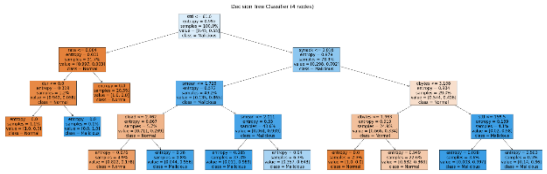


Gambar 9. Feature Importance Decision Tree

Visualisasi Decision Tree Classifier

Visualisasi ini memberikan gambaran parameter entropy yang digunakan untuk membangun decision tree classifier dengan

kedalaman maksimum sebanyak 4 node. Visualisasi ini memungkinkan untuk memahami bagaimana algoritma Decision Tree membuat keputusan berdasarkan fitur-fitur yang relevan. Dalam classifier, setiap node menunjukkan pembagian atau keputusan berdasarkan nilai fitur tertentu. Di setiap node, proporsi kelas "Normal" dan "Malicious" ditampilkan dengan persentase, memberikan gambaran visual tentang distribusi kelas pada setiap tingkat keputusan. Visualisasi ini membantu peneliti dan pembaca memahami dan menganalisis cara model Decision Tree membuat keputusan untuk dataset yang digunakan.



Gambar 10. Decision Tree Classifier

Perbandingan Peforma

Hasil evaluasi menunjukkan bahwa, meskipun perbedaan akurasi yang kecil, algoritma K-Nearest Neighbors (K-NN) dan Decision Tree memiliki tingkat akurasi yang tinggi. Model K-NN mencapai akurasi sebesar 91.54% dan Decision Tree mencapai akurasi sebesar 92.41%.

Recall menggambarkan kemampuan model untuk mendeteksi sebagian besar paket malis secara efektif, sementara Decision Tree sedikit lebih baik dalam hal ini. Nilai recall K-NN adalah 90.17%, sedangkan nilai Decision Tree 92.72%.

Dalam hal precision, nilai K-NN sebesar 94.21% dan nilai Decision Tree sebesar 93.46% menunjukkan seberapa presisi model dalam mengklasifikasikan paket sebagai malis. K-NN menonjol dalam hal ini, meskipun perbedaan kecil.

Analisis ini menunjukkan bahwa model K-NN memiliki nilai 92.15% F1-Score, yang merupakan gabungan dari akurasi dan recall, dan Decision Tree memiliki nilai 93.09% F1-Score. Ini memberikan gambaran lengkap tentang bagaimana kedua algoritma bekerja sama dalam deteksi paket malis.

4. SIMPULAN

Simpulan dari penelitian ini adalah bahwa algoritma K-Nearest Neighbor (K-NN) dan Decision Tree memiliki performa yang cukup baik dalam mendeteksi paket malis pada dataset UNCW-NB15. Decision Tree mencapai akurasi 92.41%, sedangkan K-NN mencapai akurasi 91.54%. Meskipun Decision Tree sedikit lebih akurat daripada K-NN, K-NN memiliki keunggulan dalam recall dan F1-Score. Oleh karena itu, pemilihan antara kedua algoritma ini sebaiknya bergantung pada prioritas deteksi yang diinginkan: apakah mengutamakan keakuratan umum atau mendeteksi paket malis lebih banyak.

DAFTAR PUSTAKA

[1] D. Setiawan, A. Nugraha, and A. Luthfiarta, “Komparasi Teknik Feature Selection Dalam Klasifikasi Serangan IoT Menggunakan Algoritma Decision Tree,” *Jurnal Media Informatika Budidarma*, vol. 8, pp. 83–93, 2024, doi: 10.30865/mib.v8i1.6987.

[2] Maulana I and Alamsyah A, “Optimalisasi Deteksi Serangan DDoS Menggunakan Algoritma Random Forest, SVM, KNN dan MLP pada Jaringan Komputer,” *Indonesian Journal of Mathematics and Natural Sciences*, vol. 46, no. 2. 2023, doi: 10.15294/ijmns.v46i2.48231.

[3] N. M. Balamurugan, R. Kannadasan, M. H. Alsharif, and P. Uthansakul, “A Novel Forward-Propagation Workflow Assessment Method for Malicious Packet Detection,” *Sensors*, vol. 22, no. 11, Jun. 2022, doi: 10.3390/s22114167.

[4] K. B. Dasari and N. Devarakonda, "Detection of different DDoS attacks using machine learning classification Algorithms," *Ingenierie des Systemes d'Information*, vol. 26, no. 5, pp. 461–468, Oct. 2021, doi: 10.18280/isi.260505.

[5] M. F. Kamarudin Shah, M. Md-Arshad, A. Abdul Samad, and F. A. Ghaleb, "Comparing FTP and SSH Password Brute Force Attack Detection using k-Nearest Neighbour (k-NN) and Decision Tree in Cloud Computing," *International Journal of Innovative Computing*, vol. 13, no. 1, pp. 29–35, May 2023, doi: 10.11113/ijic.v13n1.386.

[6] R. Firdaus, A. Id Hadiana, and F. Kasyidi, "Model Deteksi Botnet Menggunakan Algoritma Decision Tree Dengan Untuk Mengidentifikasi Serangan Click Fraud," *Journal of Informatics and Communications Technology*, vol. 4, no. 2, pp. 10–020, 2022, doi: 10.52661.

[7] H. At Thooriqoh, M. H. Naufal Azzmi, Y. Ari Tofan, and A. M. Shiddiqi, "Malicious Traffic Detection In Dns Infrastructure Using Decision Tree Algorithm," *JUTI: Jurnal Ilmiah Teknologi Informasi*, vol. 20, no. 1, pp. 45–53, Jan. 2022, doi: 10.12962/j24068535.v19i3.a1054.

[8] A. Pathak and S. Pathak, "Study on Decision Tree and KNN Algorithm for Intrusion Detection System." [Online]. Available: www.ijert.org

[9] Y. Ariyanto, V. A. H. Firdaus, and H. Pramana, "Klasifikasi Jenis serangan DOS dan Probing pada IDS menggunakan metode K-Nearest Neighbor," *SEMINAR INFORMATIKA APLIKATIF POLINEMA (SIAP)*, 2020, [Online]. Available: http://kdd.ics.uci.edu

[10] M. N. Faiz, O. Somantri, and A. W. Muhammad, "Machine Learning-Based Feature Engineering to Detect DDoS Attacks," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, vol. 11. no. 3, 2022, doi: 10.22146/jnteti.v11i3.3423.

[11] M. F. Alamsyah, T. P. Satriawan, F. N. Ramadanis, R. A. Mulyawan, C. Edmond, and R. Firmansyah, "Analisa Komparasi Algoritma Naïve Bayes, Decision Tree Dan KKN Untuk Klasifikasi Kebakaran Hutan Pada Wilayah Aljazair," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 1, no. 2, pp. 72–86, 2023, doi: 10.59581/jusiik-widyakarya.v1i2.425.

[12] A. Campazas-Vega, I. S. Crespo-Martínez, Á. M. Guerrero-Higueras, C. Álvarez-Aparicio, V. Matellán, and C. Fernández-Llamas, "Analyzing the influence of the sampling rate in the detection of malicious traffic on flow data," *Computer Networks*, vol. 235, Nov. 2023, doi: 10.1016/j.comnet.2023.109951.

[13] F. S. Pattiiha and H. Hendry, "Perbandingan Metode K-NN, Naïve Bayes, Decision Tree untuk Analisis Sentimen Tweet Twitter Terkait Opini Terhadap PT PAL Indonesia," *JURIKOM (Jurnal Riset Komputer)*, vol. 9, no. 2, pp. 506–514, Apr. 2022, doi: 10.30865/jurikom.v9i2.4016.

[14] O. Setyawan, A. Firizkiansah, and A. Nuryanto, "Klasifikasi Tingkat Keparahan Serangan Jaringan Komputer Dengan Metode Machine Learning," *Journal of Information System, Informatics and Computing*, vol. 5, no. 1, pp. 128–133, Jun. 2021, doi: 10.52362/jisicom.v5i1.443.

[15] R. Nuraeni, A. Sudiarjo, and R. Rizal, "Perbandingan Algoritma Naïve Bayes Classifier Dan Algoritma Decision Tree Untuk

Analisa Sistem Klasifikasi Judul Skripsi," *Innovation In Research Of Informatics*, vol. 3, no. 1, pp. 26–31, 2021.