CESS (Journal of Computer Engineering System and Science) **p-ISSN :2502-7131**
Vol. 6 No. 1 Januari 2021 **e-ISSN :2502-714x**

PEMODELAN IDENTIFIKASI TRAFIK BITTORRENT DENGAN PENDEKATAN *CORRELATION BASED FEATURE SELECTION* (CFS) MENGGUNAKAN ALGORITME *DECISION TREE* (C4.5)

Hesmi Aria Yanti[1], Heru Sukoco[2], Shelvie Nidya Neyman[3]

[1]Program Studi Magister Ilmu Komputer, Departemen Ilmu Komputer FMIPA IPB University
Jl. Raya Dramaga Kampus IPB Dramaga Bogor, West Java, 16680, Indonesia
[1]hesmiaria11@gmail.com; [2]hsrkom@apps.ipb.ac.id; [3]shelvie.neyman@gmail.com

Abstrak—**BitTorrent merupakan protokol P2P file *sharing* perangkat lunak yang memungkinkan *client* mendistribusikan data ke *client* lainnya dan dapat mempengaruhi kinerja layanan jaringan. Pengambilan data trafik *client* BitTorrent menggunakan data sekunder yang diambil dari sumber resmi pada link https://unb.ca/cic/*datasets*/index.html pada tahun 2016. Data trafik digunakan sebagai model identifkasi trafik BitTorrent menggunakan *correlation based feature selection* (CFS) dan analisis model identifikasi trafik menggunakan Algoritme *Decision Tree* (C4.5). Seleksi fitur dilakukan guna membersihkan fitur-fitur yang tidak relevan sehingga dapat mempengaruhi hasil nilai *accuracy*. Hasil seleksi fitur didapat 7 fitur dan 1 kategori dengan 244.689 *record* dan identifikasi menentukan model *rule tree training data* dipilih empat nilai *accuracy* terbaik. Selanjutnya model *training data* dilakukan uji *testing data*, guna identifikasi trafik BitTorrent. Hasil uji *testing data* didapatlah nilai *accuracy* trafik BitTorrent terbaik 98.82% dengan jumlah data 73.406 *record* pada uji testing data 30%.**

Kata Kunci— **algoritme C4.5, BitTorrent, *correlation based feature selection,* identifikasi trafik, pemodelan.**

Abstract— **BitTorrent is a P2P file sharing software protocol that allows clients to apply data to other clients and can affect network performance. BitTorrent client traffic data collection uses secondary data taken from official sources on the link https://unb.ca/cic/datasets/index.html in 2016. Traffic data is used as a model for BitTorrent traffic identification using feature-based correlation selection (CFS) and traffic analysis model analysis using Decision Tree Algorithm (C4.5). Feature selection is done to clean irrelevant features so that they can affect the results of the accuracy value. The results of feature selection obtained 7 features and 1 category with 244,689 records and the system connecting the rule tree data training model selected the four best accuracy values. Furthermore, the model training data is carried out by testing the BitTorrent traffic trial data. The results of data testing obtained the best BitTorrent traffic accuracy value of 98.82% with 73,406 records on the 30% data test.**

Keywords— **BitTorrent, C4.5 algorithm, correlation based feature selection, traffic identification, modelling.**

I. PENDAHULUAN

Identifikasi trafik data pada layanan jaringan internet khususnya pada BitTorrent sangat dibutuhkan untuk pengelolaan dan pemantauan jaringan guna menjaga kualitas dan keamanan layanan jaringan trafik *peer-to-peer* (P2P) [1]. BitTorrent merupakan *protocol* P2P *file sharing client* mendistribusikan data ke *client* lainnya dengan *file* berskala besar maupun skala kecil [2]. Identifikasi IP menggunakan *port* dan *payload* masih banyak memiliki kekurangan, sehingga alternatif yang digunakan sebelum identifikasi yaitu seleksi fitur. Metode seleksi fitur dapat meningkatkan kinerja komputasi untuk identifikasi trafik.

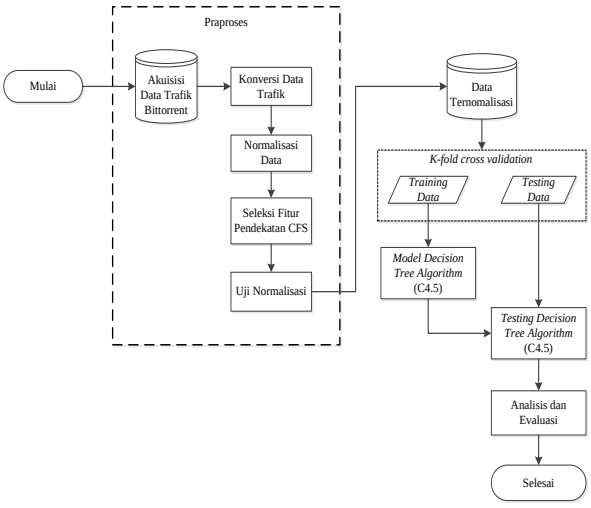
Fitur berbasis *correlation-based feature selection* (CFS) dan evaluasi konsistensi *consistency-based feature selection* (CON). Kemudian melakukan uji *maching learning bayesian network*, C4.5 *decision tree*, *naïve bayes* dan *naive bayes tree* untuk menentukan nilai *accuracy* terbaik. Hasil uji *accuracy* dengan *maching learning*, didapatlah klasifikasi terbaik yaitu *decision tree* (C4.5) *algorithm* [3]. *Machine learning decision tree* (C4.5) *algorithm*, *suport vector machine* (SVM), *naïve bayes* dan random forest (RF) untuk menentukan klasifikasi trafik yang bersifat parametric. *Berdasarkan machine learning* untuk identifikasi trafik pada *port based* dalam identifikasi nomor *port* di *header* TCP jika *payload* tidak dienkripsi, maka identifikasi menggunakan *machine learning* (ML) *decision tree* (C4.5) *algorithm* sangat baik dalam menentukan model klasifikasi trafik yang bersifat parametric [4]. Selanjutnya melakukan seleksi fitur dengan membandingkan CFS, *consistency-based subset evaluation* (CSE) *dan principal component analysis* (PCA) pada trafik Botnet menggunakan *algoritme decision tree* (C4.5), *naïve bayes* dan *bayes network*. Hasil *accuracy* terbaik dengan pendekatan seleksi fitur menggunakan CSE 98.24%, CFS yaitu 98.18% dan PCA sebanyak 97.37% dengan klasifikasi menggunakan *decision tree* (C4.5). *Decision trees* dan *regression tree* (CART) *algorithm* digunakan sebagai

teknik seleksi fitur Botnet (P2P) yang relevan. Adapun teknik menentukan fitur Botnet (P2P) yaitu dengan melakukan uji metode *decision tree, principal component analysis* dan *Relief algorithm*, dimana model *neural network* dengan menggunakan *decision tree* menghasilkan nilai *accuracy* identifikasi lebih baik yaitu nilai rata-rata 99.08% tingkat kesalahan lebih rendah positif dan dengan tingkat positif palsu 0.75% [6].

Penelitian di atas belum terdapat analisis mengenai pemodelan identifikasi trafik BitTorrent menggunakan pendekatan CFS dengan algoritme *decision tree* (C4.5). Penelitian ini bertujuan untuk menentukan model trafik BitTorrent dengan menggunakan pendekatan CFS dalam melakukan seleksi fitur dan menggunakan Algoritme *Decision Tree* (C4.5) untuk identifikasi trafik BitTorrent.

II. METODOLOGI

Dataset yang digunakan merupakan *dataset* sekunder BitTorrent, Facebook dan Youtube. Diunduh menggunakan *local area network* (LAN) diruang diskusi S2 Ilmu Komputer Institut Pertanian Bogor pada bulan Maret tahun 2019. Alur tahapan metode penelitian ini dapat dilihat pada Gbr. 1.



Gbr. 1 Tahapan penelitian

A. *Pra proses*

Tahapan pra proses yang dilakukan yaitu akuisisi data trafik BitTorrent, normalisasi data, seleksi fitur, dan uji normalisasi. Data trafik BitTorrent yang digunakan merupakan *dataset* sekunder Unb ISCX dengan format *pcap*, *dataset* di *input* dalam format *query*, pesan berformat *query* merupakan *request* data yang dibutuhkan. Data trafik yang sesuai akan masuk kesistem, kemudian data diolah dan data mentah dimanipulasi menjadi informasi. Kemudian data di konversi untuk selanjutnya di normalisasi data, seleksi fitur dan uji normalisasi. Penelitian ini menggunakan pendekatan CFS untuk seleksi fitur. Selanjutnya dilakukan uji normalisasi data untuk menentukan fitur yang telah diseleksi berdistribusi normal atau diambil dari data trafik yang normal.

1) Akuisisi Paket Data Trafik BitTorrent: Akuisisi paket data trafik menggunakan Wireshark dalam format *pcap* yang diambil berupa data trafik sekunder yang tersedia dari sumber resmi yaitu *Dataset* Unb ISCX pada link https://www.unb.ca/cic/datasets/url-2016.html di University of New Brunswick tahun 2016. Pemodelan yang akan digunakan pada identifikasi trafik yaitu menggunakan *dataset* trafik *anomaly*.

2) Konversi Data Trafik: Data trafik dengan format *pcap* dikonversi keformat .CSV file [8]. Konversi data trafik bertujuan untuk mempermudah peneliti dalam melakukan proses olah data di pemrograman R studio dan identifikasi trafik dalam menentukan pemodelan trafik BitTorrent.

3) Normalisasi Data: Normalisasi data merupakan proses penskalaan *dataset*, sehingga didapatkan nilai rentang tertentu untuk melakukan distribusi normal. Distribusi normal merupakan parameter dari simpangan baku untuk menganalisis data normal, dimana *mean* = 0 dan simpangan baku = 1, format distribusi normal seperti genta (*bell-sphaped*) yang simetris. Distribusi normal yaitu melakukan normalisasi data terhadap data asli untuk menghasilkan nilai normal, antara *dataset* sebelum dan sesudah diproses menggunakan metode *min-max* [9]. Adapun proses normalisasi menggunakan metode *min-max* (1) [10].

$$x' = \frac{x_i - min(x)}{max(x) - min(x)} \qquad (1)$$

x' merupakan hasil nilai normalisasi, x_i adalah fitur intence, sedangkan x ialah set mewakili tipe fitur, min (x) adalah nilai minimum pada fitur *x* dan max(x) adalah nilai maksimum pada fitur *x*.

4) Seleksi Fitur Pendekatan CFS: Proses seleksi fitur dilakukan untuk mengurangi jumlah fitur tidak relevan yang memiliki banyak *noise, missing value, inkonsentesi* dan *error* dan mencari ciri-ciri raw data berkorelasi, mempengaruhi identifikasi trafik BitTorrent dalam menentukan pemodelan berupa nilai *accuracy* menggunakan metode CFS. Pengambilan data dilakukan sistem skenario rata-rata fitur dengan melakukan pengujian hipotesis *univariate* ($x1, x2,...,x_n$) dimana sampel random dari fitur berdistribusi normal (μ,σ^2) dengan nilai fitur σ^2 yang sudah diketahui [11]. Sistem skenario pengambilan sampel menggunakan teknik *simpel random sampling dan* diambil nilai *mean* dapat dilihat pada (2) [12].

$$\bar{\mathbf{X}} = \frac{\sum f_i x_i}{\sum f_i} \qquad (2)$$

$\sum$ lambang penjumlahan semua data pengamatan, sedangkan f_i frekuensi data ke-i, n banyaknya sampel

data, $\bar{x}$ merupakan nilai rata-rata sampel. Proses seleksi fitur bertujuan untuk mencari ciri-ciri yang diinginkan dari raw data, selanjutnya data akan diolah berdasarkan korelasinya. Proses ini guna memperkecil jumlah raw data dan memberikan informasi data yang dibutuhkan.

Metode CFS ini adalah bagian dari metode *heuristic* dengan cara melihat fungsi-fungsi dari setiap fitur yang digunakan untuk prediksi kelas antar fitur dengan korelasi antar fitur. Nilai suatu *subset* fitur s yang terdiri dari fitur k (3) dan kriteria CFS (4) [13].

$$r_{sk} = \frac{kr_{cf}}{\sqrt{k+k(k-1)r_{ff}}} \quad (3)$$

$$CFS = max_{sk} \left[\frac{r_{cf1} + r_{cf2} + \cdots + r_{cfk}}{\sqrt{k+k(k-1)r_{ff}}} \right] \quad (4)$$

Dimana r_{sk} adalah hubungan antara fitur, k adalah jumlah fitur sedangkan r_{cf} merupakan rata-rata hubungan fitur dan kelas, r_{ff} rata-rata korelasi antara bagian fitur dan max_{sk} adalah maksimal fitur.

1) Uji Normalisasi: Uji normalisasi diperlukan untuk menilai *dataset* trafik yang telah dinormalisasi berdistribusi normal atau tidak normal. Uji normalisasi dapat dilihat dengan pemeriksaan menggunakan uji *shapiro-wilk* atau uji w [14]. Uji normalitas statistik yang digunakan yaitu dengan asumsi jika *P*-value < $\propto$ maka tolak H_0 dan sebaliknya jika *P*-value > $\propto$ gagal tolak H_0, dimana H_0 adalah data distribusi normal dan H_1 adalah data distribusi tidak normal. Uji normalisasi data terhadap fitur-fitur yang berkorelasi dilihat menggunakan plot. Evaluasi normalisasi sampel yang lebih besar terdistribusi normal dapat dilihat dari hasil histogram atau plot Q-Q [15].

B. Dataset Ternormalisasi

Dataset ternormalisasi merupakan hasil dari proses normalisasi data trafik pada tahapan praproses, dimana *dataset* ternormalisasi akan digunakan sebagai implementasi uji *training data* dan *testing data* menggunakan metode *k-fold cross validation.*

1) K-Fold Cross Validation: k-fold cross validation merupakan implementasi uji *dataset* ternormalisasi. *Cross validation* merupakan validasi model untuk menilai keakuratan hasil analisis. Adapun *dataset* ternormalisasi dilakukan tahap uji *testing data* dan *training data* untuk proses klasifikasi. *Dataset* ternormalisasi dibagi menjadi K bagian yaitu K satu digunakan sebagai *training data* dan K-1 digunakan sebagai *testing data* [16]. Pembagian data dilakukan menggunakan *k-fold cross validation* dengan nilai k sama dengan 10 (5).

$$E = \frac{1}{K} \sum_{i=1}^{k} E_i \quad (5)$$

E merupakan data di normalisasi dan K jumlah pembagian/bagian. Tahap selanjutnya melakukan pembagian data kemudian diklasifikasikan menggunakan metode *k-fold cross validation* dengan nilai K=10.

2) Pemodelan Identifikasi Trafik: Melakukan identifikasi trafik BitTorrent menggunakan sistem operasi *Windows*, jaringan di partisi menggunakan *switch layer*-3 dan data *client* BitTorrent diambil dari bank data sebagai model untuk identifikasi trafik. Identifikasi dalam akuisisi *data*set trafik yang relevan menggunakan pendekatan CFS, CFS mengidentifikasi dan menyaring fitur yang tidak relevan, serta mengidentifikasi fitur yang relevan [17]. Paket data yang digunakan untuk identifikasi menentukan pemodelan trafik BitTorrent menggunakan Algoritme *Decision Tree* (C4.5).

3) Algoritme Decision Tree (C4.5): Merupakan pengembangan dari *Iterative Dichotomies Algorithm* (ID3). *Decision tree* berguna untuk mengeksplorasi data dengan menemukan hubungan yang tersembunyi antara fitur *input* dengan fitur target. Data *input* pada C4.5 *algorithm* berupa tabel dan menghasilkan *output* berupa pohon keputusan [18]. Membangun model pohon keputusan untuk pemilihan fitur menjadi *node* menggunakan tiga pendekatan yaitu : informasi total nilai *entropy* pada *dataset*, menghitung nilai informasi *entropy* dalam setiap fitur *dataset*, dan menghitung nilai *gain* pada setiap fitur dalam *dataset* [19]. Adapun total nilai *entropy dataset* diperoleh dengan cara membandingkan atau membagi jumlah keseluruhan fitur dalam *dataset* dengan nilai kelas atau kategori berdasarkan kriteria kemudian dikalikan dengan nilai log_2.. total nilai *entropy dataset* didefinisikan pada (6).

$$Entropy\ (S) = -\sum_{i=1}^{N} P_j\ log_2 P_j \quad (6)$$

S adalah Jumlah total fitur *dataset*, P_j merupakan probabilitas munculnya fitur dan N banyaknya fitur yang digunakan. Menghitung informasi nilai *entropy* pada setiap fitur *dataset* untuk mendapatkan informasi nilai *entropy* yaitu dengan menjumlahkan nilai perbandingan jumlah kelas atau kategori berdasarkan kriteria dengan jumlah data yang dimiliki sebuah fitur berdasarkan kategori, kemudian dikalikan dengan nilai log_2. Perhitungan nilai *entropy* pada setiap fitur dalam *dataset* disebut dengan *split info* (7).

$$Split\ info\ (S,A) = -\sum_{i=1}^{m} \frac{|S_i|}{S}\ x\ log_2 \frac{|S_i|}{S} \quad (7)$$

(*S*,*A*) Nilai *entropy* dari fitur berdasarkan *subset* fitur,S_i Jumlah *subset* dari fitur *S*, m Jumlah *subset* dari fitur *S* dan $|S_i|$ Nilai mutlak atau *absolut* jumlah *subset*. Tahap ketiga yaitu menghitung nilai *gain* pada setiap fitur dalam *dataset* (8).

$$Gain\ (S,A) = Entropy - \sum_{i=1}^{n} \frac{|S_i|}{|S|}\ x\ Entropy\ (S_i) \quad (8)$$

S Data sampel yang digunakan untuk *training*, A Atribut, n banyaknya jumlah fitur, $|S_i|$ Jumlah sampel untuk nilai v, |S| Jumlah seluruh sampel data, (Si) *Entropy* untuk sampel-sampel yang memiliki nilai i.Tahapan ketiga merupakan tahapan penyempurnaan terhadap definisi nilai *gain* menjadi nilai *gain ratio* (9).

$$Gain\ Ratio\ (S,A) = \frac{Gain\ (S,A)}{SplitInfo\ (S,A)} \qquad (9)$$

Definisi nilai *gain* dan nilai *gain ratio* memiliki perbedaan, dimana ID3 *algorithm* nilai sebuah fitur didefinisikan dengan menjumlahkan total nilai *entropy*, dikurangi dengan jumlah nilai fitur yang terpilih, kemudian kategori dibagi nilai kelas fitur terpilih dari *dataset*, dikalikan dengan nilai *entropy* kelas atau kategori fitur. Sedangkan C4.5 *algorithm* yaitu total jumlah *gain* fitur *dataset* yang terpilih dibagi dengan total *split informasi* [20].

C. Analisis dan Evaluasi

Evaluasi kinerja berdasarkan perbandingan klasifikasi *parametric* nilai keakuratan terbaik yaitu *machine learning Decision Tree* (C4.5) *Algorithm*, maka untuk pengukuran menentukan klasifikasi data trafik menggunakan metode *matrix. Confusion matrix* dapat diartikan sebagai suatu *classifier* analisa dalam mengenali *tuple* dari kelas yang berbeda. Nilai dari *True Positive* (TP) dan *True Negative* (TN) memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, *False Positive* (FP) dan *False Negative* (FN) memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data Tabel I [21].

TABEL I
CONFUSION MATRIX MEAMPILKAN TOTAL *POSITIVE* DAN *NEGATIVE TUPLE*

		Predicted class		
		Yes	**No**	Total
Actual class	**Yes**	TP	FN	P
	No	FP	TN	N
	Total	P'	N'	P+N

Confuison matrix merupakan metode klasifikasi yang mengandung informasi guna membandingkan hasil klasifikasi dilakukan oleh sistem dengan hasil yang relevan. Kinerja *confuison matrix* pada pengukuran kinerja suatu sistem klasifikasi data dapat dibagi menjadi 4 (empat) jenis yaitu klasifikasi *binary*, *multi-class*, *multi-label*, dan *hierarchical*. Pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah : *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP,FP dan FN adalah jumlah positif yang benar, dan negative palsu sebagai representasi hasil proses klasifikasi [4].

Accuracy ialah penjumlahan dari rasio yang memiliki nilai positif benar dengan jumlah semua positif benar dan positif palsu untuk semua kelas. *Matrix* terakhir digunakan untuk mengevaluasi kualitas hasil identifikasi setiap kelas aplikasi dan nilai *accuracy* menggambarkan tingkat keakuratan sistem dalam mengklasifikasikan data secara benar (10).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}\ x\ 100\% \qquad (12)$$

Penggunaan metode *confuison matrix* pada penelitian ini bertujuan untuk mendapatkan pemodelan identifikasi trafik BitTorrent dengan nilai keakuratan data yang baik.

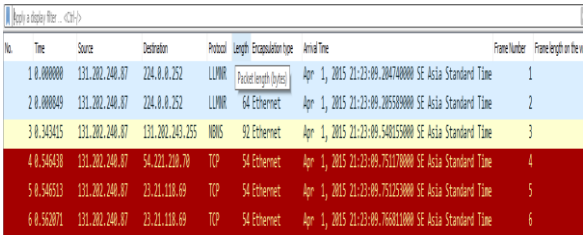
III. HASIL DAN PEMBAHASAN

A. Pra proses

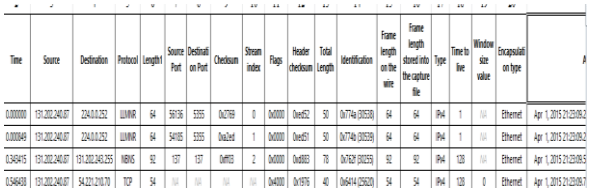
Pra proses data melakukan analisis akuisisi paket data trafik berupa raw data dalam format *pcap*. *Dataset* trafik dalam format *pcap* dikonversi ke format .CSV dan diberi label berdasarkan kategorinya. Data yang telah dikonversi dilakukan normalisasi data. kemudian proses seleksi fitur mengunakan pendekatan CFS. Fitur yang telah diseleksi kemudian dilakukan uji normalisasi. Hasil uji normalisasi data dapat digunakan pada *training* dan *testing*.

1) Akuisisi Paket Data Trafik BitTorrent: Akuisisi data trafik BitTorrent dilakukan pada data sekunder yang diperoleh dari raw *dataset* Unb ISCX. *Dataset* Unb ISCX di publikasikan oleh University of new Brunswick di Canada tahun 2016. Selain data trafik bittorent, akuisisi data trafik dilakukan pada beberapa data trafik lainnya diantaranya Facebook dan Youtube. *Dataset* trafik yang digunakan mempunyai kategori atau *class* target yaitu BitTorrent, Facebook dan Youtube. Jumlah paket data trafik yang didapat sebanyak 67 fitur dengan jumlah *dataset* pada BitTorrent 108.541 *record*, Facebook 1.061 *record* dan Youtube sebanyak 135.087 *record*, total *dataset* trafik yaitu 244.689 *record*. Jenis *dataset* trafik pada penelitian ini yaitu dalam format *pcap.* Kemudian untuk mempermudah proses olah data pada pemrograman R studio, *dataset* dalam format *pcap* dikonversi.

2) Konversi Data Trafik: Konversi data trafik digunakan untuk mengubah data format *pcap* data ke format .CSV. Adapun *Sampling dataset* trafik dengan format *packet capture (pcap)* dapat dilihat Gbr. 2, sedangkan *sampling dataset* trafik dikonversi ke format *comma separted values* (.csv) pada Gbr. 3.



Gbr.2. *Dataset* trafik dengan format *packet capture (pcap)*



Time	Source	Destination	Protocol	Length	Source Port	Destination Port	Checksum	Stream index	Flags	Header checksum	Total Length	Identification	Frame length on the wire	Frame length stored into the capture file	Type	Time to live	Window size value	Encapsulation type	A
0.000000	131.202.240.87	224.0.0.252	LLMNR	64	56136	5355	0x27d9	0	0x0000	0xe652	50	0x774b (30539)	64	64	IPv4	1	NA	Ethernet	Apr 1, 2015 21:23:09.2
0.000849	131.202.240.87	224.0.0.252	LLMNR	64	54185	5355	0xa2ed	1	0x0000	0xe651	50	0x774b (30539)	64	64	IPv4	1	NA	Ethernet	Apr 1, 2015 21:23:09.2
1.949415	131.202.240.87	131.202.243.255	NBNS	92	137	137	0xeff03	2	0x0000	0xd883	78	0x762f (30255)	92	92	IPv4	128	NA	Ethernet	Apr 1, 2015 21:23:09.5
0.546408	131.202.240.87	54.221.210.70	TCP	54	NA	NA	NA	NA	0x4000	0x1976	40	0x6414 (25620)	54	54	IPv4	128	0	Ethernet	Apr 1, 2015 21:23:09.7

Gbr. 3. *Dataset* trafik dikonversi ke format *comma separted values* (.csv)

Pengambilan ciri *dataset* yang telah dikonversi ke format .CSV guna mempermudah analisa data trafik dan diidentifikasi sistem. *Dataset* trafik dalam format .CSV diberi label berdasarkan kategorinya. Adapun format pada *dataset* yaitu berupa nominal dan bilangan deskripsi, format *dataset* dalam bentuk bilangan deskripsi dikonversi ke bilangan nominal. Data trafik yang dikonversi yaitu sebanyak 67 fitur dan 244.689 *record*. *Pseudocode* konversi data trafik ditampilkan dibawah ini.

```
library(forcats)
Dataset_trafik<- map_df(Dataset_trafik, as.numeric)
Dataset_trafik$Category<-
as.factor(Dataset_trafik$Category)
```

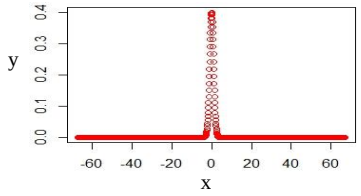
forcats merupakan *library* konversi *dataset* string atau karakter kedalam bentuk deskripsi. *Dataset* trafik kolom *category* berupa deskripsi dikonversi dalam bentuk nominal. Hasil konversi pelabelan pada fitur kategori trafik BitTorrent, Facebook, dan Youtube dapat dilihat pada Tabel II.

TABEL II
LABEL FITUR KATEGORI

No	Kategori	Kode *Numeric*
1	BitTorrent	1
2	Facebook	2
3	Youtube	3

Hasil konversi pada fitur katagori yaitu dari bilangan deskripsi ke bilangan nominal, diaman BitTorrent yaitu 1 (satu), sedangkan Facebook 2 (dua) dan Youtube 3 (tiga). Data trafik yang telah dikonversi lakukan normalisasi data.

3) Normalisasi Data: Data yang telah dikonversi kemudian dilakukan normalisasi, dimana jumlah *record packet* yang tumpang tindih mempengaruhi proses dan olah data. Plot hasil penskalaan *dataset* terdistribusi normal dapat dilihat pada Gbr. 4.



Gbr. 4. Nilai normal *dataset* (0, 1)

Hasil nilai normal penskalaan *dataset* terdistribusi normal, dimana puncak nilai x = 0. x merupakan jumlah *record packet* yang terdapat pada fitur *dataset*. *Dataset* terdistribusi normal jika ukuran minimum dan rentang vektor memiliki puncak x = 0 dan 1. Normalisasi data untuk penskalaan *dataset* dapat meningkatkan performa lebih cepat karena komputasinya lebih di sederhanakan.

4) Seleksi Fitur Pendekatan CFS: Seleksi fitur dapat mempengaruhi model dan nilai *accuracy*, sehingga penelitian ini menggunakan sistem skenario menentukan persentase *sampling* yang akan digunakan. Adapun jumlah total *dataset* trafik pada analisis ini yaitu 244.689 *record packet* dan 67 fitur. Tabel III menunjukan persentase *sampling dataset* trafik.

TABEL III
SAMPLING DATASET TRAFIK.

No	***Sampling* %**	Fitur
1	0	20
2	10	43
3	20	43
4	30	43
5	40	43
6	50	51
7	60	52
8	70	53
9	80	53
10	90	53

Pengambilan *dataset* trafik dengan menggunakan skenario *random sampling* dimana data didistribusikan secara acak. Kemudian fitur yang tidak relevan dapat menyebabkan sampel bias dan mempengaruhi nilai *accuracy*, sehingga fitur dianggap tidak relevan dihapus [22]. *Pseudocode* skenario seleksi fitur dilakukan dengan menggunakan *library* pemrograman R studio.

```
Dataset_trafik<-Dataset_trafik[,-
which(colMeans(is.na(Dataset_trafik))> 0.4)]
remove_features <- function(df, features) {
  rem_vec <- unlist(strsplit(features, ', '))
  res <- df[,!(names(df) %in% rem_vec)]
  return(res)
}
```

Pada *Dataset*_trafik dilihat rata-rata *not available* (NA) lebih dari 0.4 (40%), kemudian fitur yang rata-rata NA kurang dari 40% dihilangkan atau dihapus. Sedangkan pada *remove_features* menghilangkan *feature* dari *data frame* (parameter df), dimana *feature* yang akan di *remove* didefinisikan terlebih dahulu berdasarkan *feature* yang dipilih.

Hasil dari seleksi fitur trafik menggunakan skenario *sampling* 40%, didapatlah 43 fitur. Berdasarkan hasil dari skenario *sampling* diatas masih terdapat fitur yang tidak relevan, sehinggga dilakukan penghapusan fitur secara manual. Penghapusan fitur secara manual dipilih berdasarkan hasil tampilan pada *console* R studio yang mengalami *error* atau tidak relevan. Hasil dari penghapusan fitur yang tidak relevan dengan mengunakan *pseudocode* diatas, maka didapatlah 7 fitur dengan 244,689 *record*. Adapun hasil dari seleksi fitur dapat dilihat pada Tabel IV.

TABEL IV
HASIL SELEKSI FITUR

Fitur	Deskripsi
Time	Throughput
Source	Alamat pengirim
Destination	Alamat tujuan
Protocol	Nomer dari jenis protokol yang digunakan
Header checksum	Digunakan untuk pengecekan apabila data rusak
Identification	Identifikasikan paket IP
Protocols in frame	Protokol pada paket

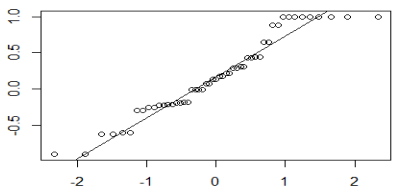
Selanjutnya fitur yang dihasilkaan digunakan untuk perhitungan korelasi antar fitur secara linier menggunakan pendekatan CFS. Hasil korelasi fitur menggunakan pendekatan CFS dapat dilihat pada Tabel V.

TABEL V
SELEKSI FITUR MENGGUNAKAN PENDEKATAN *CORRELATION BASED FEATURE SELECTION* (CFS)

	Time	Source	Destination	Protocol	Header Check Sum	Identification	Protocols in frame
Time	1						
Source	0.14	1					
Destination	-0.01	-0.90	1				
Protocol	-0.22	0.17	-0.23	1			
Header Checksum	0.05	-0.06	0.07	-0.02	1		
Identification	-0.13	0.17	-0.19	0.01	-0.43	1	
Protocols in frame	-0.19	0.21	-0.29	0.64	0.01	-0.01	1

Tabel V menunjukkan nilai *threshold* pada fitur, dimana setiap fitur memiliki korelasi negative dan positif dengan kategori atau kelas. Pemilihan fitur merupakan proses yang efektif untuk mengurangi fitur-fitur yang tidak relevan dan dapat meningkatkan nilai *accuracy* identifikasi trafik BitTorrent dengan menggunakan uji rata-rata info gain dan gain ratio. Dimana fitur yang paling relevan yaitu *time*, *source*, *destination*, *protocol*, *header checksum*, *identification* dan *protocols in frame*. Parameter untuk CFS ialah nilai *threshold*, nilai *threshold* merupakan nilai batas korelasi (*minimum* dari nilai *symmetrical uncertainty*).

5) Uji Normalisasi: untuk mendektesi data redundan, data rendundan dapat mempengaruhi hasil *accuracy* identifikasi trafik. Uji normalisasi dilakukan untuk menghasilkan struktur label yang normal atau tidak normal [14]. *Output* dari uji normalisasi menggunakan uji *shapiro-wilk* dan hasil seleksi fitur menggunakan pendekatan *correlation matrix* [1:9] yaitu W= 0.91997 dan *P*-value = 8.576e-05, dimana *P*-value = 8.576e-05 ialah 0.0009 < 0.05, sehingga perlu dilakukan tranformasi data. Hasil transformasi data yaitu 0.00028 < 0.05, dimana data tidak terdistribusi normal. Metode *shapiro-wilk* hanya dapat mengolah sebanyak 5000 data. Sehingga uji normalisasi *dataset* yang berkorelasi positif dan negatif terdistribusi normal dengan data lebih besar dapat dilihat menggunakan plot normal Q-Q. *Output* uji normalisasi menggunakan plot normal Q-Q dapat dilihat pada Gbr. 5



Gbr. 5. Hasil plot normalisasi

Sumbu horizontal menggambarkan nilai Pi=(i-0.5)/n. n adalah banyaknya *dataset* [23]. Hasil plot Q-Q pada uji normalisasi, nilai *P*-value ditempatkan pada sumbu vertikal (Y) dan nilai korelasi ditempatkan pada sumbu horizontal (X). Lingkaran tersebar mendekati garis lurus (kurva), sehingga data yang berkorelasi dapat dinyatakan terdistribusi normal.

B. Dataset Ternormalisasi

Dataset yang ternormalisasi selanjutnya digunakan sebagai *training data* dan *testing data* untuk identifikasi trafik BitTorrent dengan mengunakan *k-fold cross validation.* Fitur yang digunakan *yaitu time, source, destination, protocol, header checksum, identification, protocols in frame,* dan kategori dengan total paket data 244,689 *record.*

1) K-Fold Cross Validation: Penggunaan *k-fold cross validation* pada uji *training data* dan *testing data* menggunakan skenario pembagian *training data* untuk 10 *folds. Pseudocode* uji *dataset* ditampilkan dibawah ini.

```
library(caret)
index=createDataPartition(y=Dataset_trafik$Category,
p=0.7,ist=FALSE)
train.set=Dataset_trafik[index,]
test.set=Dataset_trafik[-index,]
iris.tree = train(Category ~ .,
            data=train.set,
            method="rpart", preProcess = "scale",
trControl=trainControl(method="repeatedcv",number
= 10, repeats=3))
```

Berdasarkan *pseudocode* diatas *package caret* digunakan dalam menentukan klasifikasi dan regresi *training*, *create data partition* digunakan untuk mendapatkan sampel pada pembagian *training data* dan *testing data* secara *random*, berdasarkan *dataset* fitur *category* dengan menggunakan metode *rpart* dan dilakukan 10 *folds* dan 3 *Epochs.* Hasil uji analisis

dengan cara *sampling* yaitu pembagiaan *dataset training* dan *testing* Tabel VI.

TABEL VI
SAMPLING DATA TRANING DAN *DATA TESTING*

Sampling %		***Training Record***	***Testing Record***
Training Data	***Testing Data***		
10	90	24.471	220.218
20	80	48.940	195.749
30	70	73.409	171.280
40	60	97.877	146.812
50	50	122.346	122.343
60	40	146.815	97.874
70	30	171.283	73.406
80	20	195.752	48.937
90	10	220.221	24.468

Dari pembagian data *training* dan *testing*, maka akan digunakan sebagai uji model untuk identifikasi trafik BitTorrent menggunakan *decision tree* (C4.5) *algorithm*.

2) Pemodelan Identifikasi Trafik: Pemodelan identifikasi trafik BitTorrent menggunakan algoritme *decision tree* (C4.5) diimplementasikan dengan menggunakan metode *rpart* pada pemrograman R studio dan hsil dari uji data diimplementasikan menjadi *rule tree*. *Pseudocode* pemodelan identifikasi trafik dapat dilihat dibawah ini.

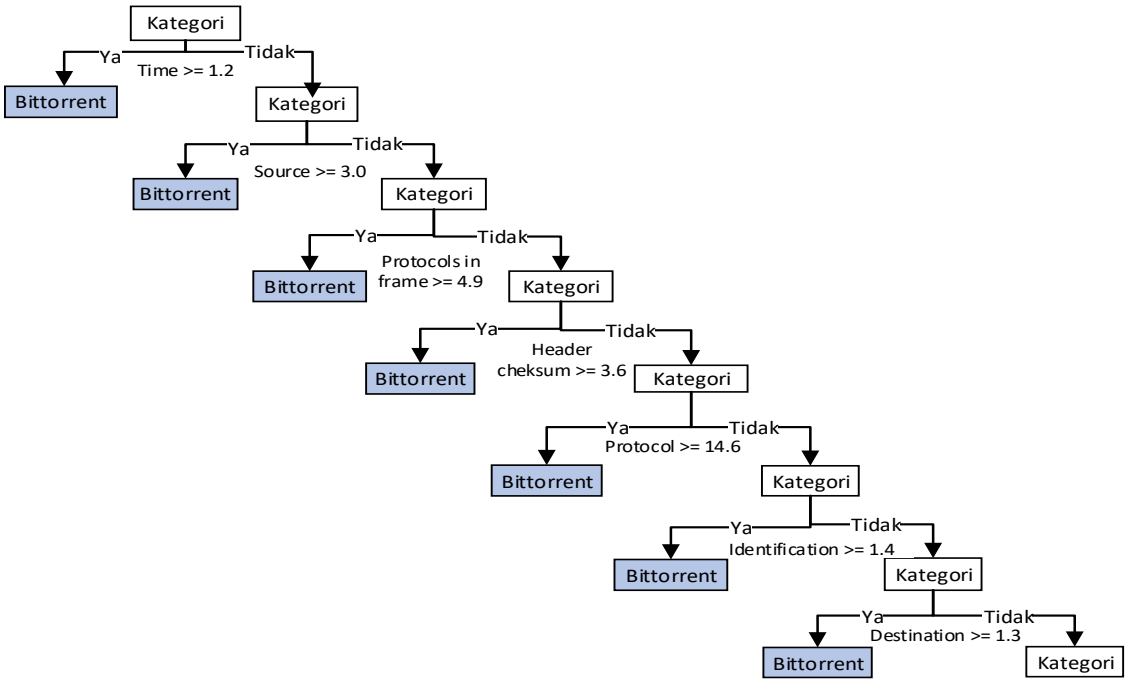
```
iris.tree = train(Category ~ .,
                  data=train.set,
                  method="rpart",preProcess="scale",
trControl=trainControl(method="repeatedcv",number=10,
repeats=3))
text(iris.tree$finalModel,use.n = TRUE, cex=.4)
plot(iris.tree$finalModel,uniform = TRUE,
    main="Classification Tree")
```

Iris.tree untuk membuat model identifikasi trafik menggunakan *train.set* atau *training data* berdasarkan *category*. Metode *rpart* 10 *folds* dan 3 *epochs* digunakan untuk mendapatkan model *rule tree*. Dimana persamaan (7) yaitu *informasi gain*, semakin tinggi nilai *informasi gain* maka semakin baik fitur tersebut untuk di jadikan *root*. Menentukan model untuk identifikasi trafik BitTorrent, dipilih berdasarkan nilai terbaik dari *accuracy training data* dan *tesing data* Tabel VII.

TABEL VII
ACCURACY TRAINING DATA DAN *TESTING DATA*

Training Data	10	20	30	40	50	60	70	80	90
Testing Data	90	80	70	60	50	40	30	20	10
Accuracy	98.40	98.52	98.61	98.14	98,60	98.55	98.61	98.65	98.57

Pemilihan model dipilih dari 4 nilai *accuracy* terbaik yang terdapat yang terdapat pada tabel VII. *Training data* terbaik pertama yaitu terdapat pada training data 80% dan *testing data* 20% yaitu 98.65%. Nilai *accuracy* terbaik kedua terdapat pada *training data* 70% dan *testing data* 30 nilai *accuracy* 98.61%, selanjutnya terbaik ketiga terdapat pada *training data* 30% dan *testing data* 70 nilai *accuracy* 98.61% dan nilai *accuracy* terbaik keempat yaitu pada *training data* 50% dan *testing data* 50% dengan *accuracy* 98.60%. Adapun Hasil *output* dari nilai *accuracy* sebagai model *traning data* yang dipilih yaitu berupa tampilan *rule decission tree* (C4.5) *algorithm* dapat dilihat pada Gbr.6.



Gbr. 6. *Rule decission tree* (C4.5) *algorithm*

Rule tree diatas merupakan hasil model *training data* menggunakan *decission tree* (C4.5) *Algorithm* menggunakan package *rpart. Tree* merupakan hasil identifikasi trafik, dimana *root* tertinggi terdapat pada fitur *time* yang merupakan throughput *client* mengakses BitTorrent dengan nilai 1.2 *Bits Per Second* (BPS). Apabila nilai *time* lebih dari 1.2 BPS maka dilakukan identifikasi kembali berdasarkan kategori. Selanjutnya hasil dari identifikasi trafik BitTorrent didapatlah fitur *source* sebagai *node* dengan nilai 3.0 BPS, jika lebih dari 3.0 BPS identifikasi trafik bittorent akan dilakukan pemilihan berdasarkan kategori. Selanjutnya didapatlah *protocols in frame* dengan nilai *node* 4.9 BPS, jika nilai *node* lebih dari 4.9 BPS maka dilakukan identfikasi kategori trafik bittorent berdasarkan *node header checksum* dengan nilai 3.6 BPS. Berdasarkan kategori selanjutnya berdasarkan *node protocol* dengan nilai 14.6 BPS, nilai *node* jika lebih dari 3.6 BPS maka dilakukan pemilihan fitur kategori dan didapatlah *identification* dengan nilai *node* 1.4 BPS, kemudian berdasarkan fitur *destination* nilai *node* 1.3 BPS. Hasil model *trianing data* kemudian digunakan sebagai analisis dan dievaluasi untuk *testing data* dalam menganalisis nilai *accuracy* model terbaik menggunakan metode *confusion matrix*.

C. Analisis dan Evaluasi

Analisis dan evaluasi terhadap uji *testing data* menggunakan *confusion matrix* untuk total *positive* dan *negative tuple*[21]. Berdasarkan hasil uji *testing data,* maka nilai *accuracy* terbaik terdapat *multiclass confusion matrix* yaitu 3 *class* (1, 2, 3). *Class* 1 dan bukan *class* 1, antara *class* 2 dan bukan *class* 2, antara *class* 3 dan bukan *class* 3. *class* 1 merupakan trafik BitTorrent, *class* 2 Facebook dan *class* 3 Youtube. Adapun hasil total *positive* dan *negative tuple* pada *testing data* dapat dilihat pada tabel VIII.

TABEL VIII
TOTAL *POSITIVE* DAN *NEGATIVE TUPLE* PADA *TESTING DATA.*

Testing Data	TP	TN	FP	FN
70	74731	94466	1247	836
50	53392	67503	878	570
30	32043	40500	519	344
20	21347	26996	361	233

Tabel nilai total *positive* dan *negative tuple* pada *testing data*, dimana TP merupakan *class* positif dan TN tingkat kebenaran *class* BitTorrent dalam identifikasi kesalahan *class.* Sedangkan FP tingkat kesalahan pada class positif BitTorrent dan tingkat kesalahan *class* dalam identifikasi BitTorrent. *Actual class* merupakan kelas yang sebenarnya pada data *testing*, sedangkan *predicted class* merupakan kelas hasil prediksi dari model yang dihasilkan oleh klasifikasi. Hasil dari total *positive* dan *negative tuple*, selanjutnya dilakukan evaluasi menggunakan algoritme *decision tree* (C4.5) pada *matrix* menggunakan persamaan (9) untuk mencari nilai *precision, recall* (10) dan *accuracy* persamaan (11) pada tabel IX uji *accuracy testing data*

TABEL IX
HASIL EVALUASI MODEL UJI *TESTING DATA*

Testing Data	*Precision %*	*Recall %*	*Accuracy %*
70	98.34	98.89	98.78
50	98.38	98.94	98.81
30	98.41	98.94	98.82
20	98.34	98.92	98.78

Dari keempat hasil evaluasi uji *testing data* pada tabel 9, mendapatkan model terbaik untuk identifikasi trafik BitTorrent pada persamaan (11) yaitu terdapat pada nilai *accuracy* 98.82%,dimana data yang digunakan sebanyak 73.406 *record* dengan *testing data* 30%. Adapun nilai *Precision* persamaan (9) terbaik terdapat pada *testing data* 30% dengan nilai 98.41% dan nilai *recall* menggunakan persamaan (10) yaitu 98.94%. Identifikasi trafik BitTorrent menggunakan CFS pada seleksi fitur dan algoritme *decission tree* (C4.5) dengan *training data* 70% dan *testing data* 30% dapat gunakan untuk dimplementasikan kesistem layanan jaringan internet.

IV.KESIMPULAN

Penggunaan *Correlation-based Feature Selection* (CFS) menentukan fitur model trafik BitTorrent sangat mempengaruhi proses olah data dan nilai *accuracy* identifikasi trafik BitTorrent. Hasil korelasi antar fitur memiliki nilai positif dan nilai negatif yang saling berkorelasi. Fitur yang berkorelasi digunakan sebagai *training data* dan *testing data* untuk identifikasi trafik BitTorrent menggunakan algoritme *decision tree* (C4.5). Hasil nilai *accuracy* identifikasi trafik terbaik yaitu 98.82% dengan jumlah dataset sebanyak 73.406 *record* pada uji *testing data* 30%. Sehingga penggunaan CFS dan algoritme *decision tree* (C4.5) dapat digunakan dalam mengidentifikasi trafik BitTorrent berdasarkan fitur-fitur yang dipilih, diantaranya fitur *time, source, destination, protocol, header checksum, identification* dan *protocols in frame*.

Untuk penelitian selanjutnya, akuisisi *dataset* trafik dilakukan secara *real-time* dengan menggunakan metode CFS dan algoritme *decision tree* (C4.5). Sehingga dapat diimplementasikan ke sistem layanan jaringan internet dengan menggunakan fitur-fitur yang telah dipilih.

UCAPAN TERIMA KASIH

Terima kasih kepada IPB University yang telah mengizinkan melakukan penelitian ini dan University of new Brunswick di Canada di Kanada untuk kumpulan *dataset*.

REFERENSI

[1] B. Hullar, S. Laki, and A. Gyorgy, "Early Identification of Peer-to-Peer," *Traffic Machine Learning Research*, vol.1, no. 1), pp. 2-7. 2011.

[2] I. Chandra., *Teknik Berbagi Objek Lewat Jaringan P2P*, Jakarta: PT Elex Media Komputindo, 2010.

[3] N. Williams, S. Zander, and G. Armitage, "A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification," *ACM SIGCOMM. Computer. Communication. Review*, vol. 36, no. 5, pp. 7-17, 2006.

[4] C. Gu, S. Zhang, and Y. Sun, "Real Time Encrypted Traffic Identification Using Machine Learning," *Journal of software*, vol. 6, no. 6, pp. 1009-1016, 2011.

[5] P. Narang, J. M. Reddy, and C. Hota, "Feature Selection For Detection Of Peer-To-Peer Botnet Traffic," *Computer*. Vol. 1, no. 1, pp. 1-9, 2013.

[6] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P Botnet detection scheme based on decision tree and adaptive multilayer neural networks," doi.10.1007/s00521-016-2564-5, 2016.

[7] (2016) UNB: University of New Brunswick. URL dataset. [Online]. (ISCX-URL-2016).

[8] N. Zumel, dan J. Mount., *Vtreat: a data.frame Processor for Predictive Modeling*, Microsoft: Preparing Data for Analysis Using R, 2016, vol. 1, no. 1, pp. 1-16.

[9] D.A. Nasution, H.H. Khotimah, dan N. Chamidah N, "Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN," CESS (*Journal of Computer Engineering System and Science*), vol. 4, no. 1, pp. 78-82, 2019.

[10] Guntoro. (2015) Keamanan Jaringan Openflow Menggunakan Intrusion Detection System (IDS) Berbasis Backpropagation Neural Network. [Tesis]. Bogor: Institut Pertanian Bogor.

[11] S. Permana, B. Yuniarto, S. Mariyah, S. Ibnu, dan R. Nooraeni., *Data mining dengan R konsep serta implementasi*, Bogor: In Media, 2018.

[12] N. Hartati., *Statistika untuk Analisis Data Penelitian*, Bandung: Pustaka Setia, 2017.

[13] Tasmi, S.A. Valianta, dan D. Stiawan, "Klasifikasi Trafik Terenkripsi Menggunakan Metode Deep Packet Inspection (DPI)," *Annual Research Seminar*, vol. 2, no. 1, pp. 424-429, 2016.

[14] T.J. Lorenzen, and V.L. Anderson, "Design of Experiments," *A No-Name Approach*, New York: Marcel Dekker, 1993.

[15] S.M. Karadimitriou. (2016) Statistical Hypothesis Testing and Normality Checking in R Solutions, Csv and Script Files. [Online]. Available: https://www.sheffield.ac.uk/polopoly_fs/1.579191!/file/stcp karadimitriou-normalR.pdf

[16] Y.M. Mahardika, A. Sudarsono, dan A.R. Barakbah, "An Implementation Of Botnet Dataset To Preddict Accuracy Based on Network Flow Model," *International electronic symposiumon Knowladge creation and Intelegence Computing* (IES-KCIC), vol. 1, no. 1, pp. 33-39, 2017.

[17] M. Hall., *Correlation-Based Feature Selection For Machine Learning*, Hamilton: The University Of Waikato, 1999.

[18] K. Izza, dan L.B. Handoko, "Implementasi dan Analisa Hasil Data Mining Untuk Klasifikasi Serangan pada Intrusion Detection System (IDS) Dengan Algoritma C4.5," *Techno.Com*, vol. 14, no. 3, pp. 181-188, 2015.

[19] J. Suntoro., *Data Mining: Algoritma dan Implementasi dengan Pemrograman PHP*, Jakarta: PT Elex Media Komputindo, 2019.

[20] H. Qian, and Z. Qiu, "Feature selection using C4.5 Algorithm for electricity price prediction," *International Conference on Machine Learning and Cybernetics*, IEEE, pp. 175-180, 2014.

[21] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, 3nd ed., San Fancisco: Morgan Kaufman, 2011.

[22] S. Datta, and H. Kargupta. (2014) Uniform Data Sampling from a Peer-to-Peer Network, *Proceedings- International Conference on Distributed Computing Systems*. [Online]. Available:.https://www.researchgate.net/publication/221459465.

[23] Aunuddin., *Statistika: Rancangan dan Analisis Data*, Bogor: IPB Press, 2005.