Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK) DOI: 10.25126/jtiik.202071926
Vol. 7, No. 2, April 2020, hlm. 329-338 p-ISSN: 2355-7699
Akreditasi KEMENRISTEKDIKTI, No. 30/E/KPT/2018 e-ISSN: 2528-6579

ROUTING ATTACKS PADA *INTERNET OF THINGS* BERBASIS *SMART INTRUSION DETECTION SYSTEM*

Eka Lailatus Sofa[1], Subiyanto*[2]

[1,2]Jurusan Teknik Elektro, Universitas Negeri Semarang
Email: [1]ekasofa9@gmail.com, [2]subiyanto@mail.unnes.ac.i
*Penulis Korespondensi

(Naskah masuk: 09 April 2019, diterima untuk diterbitkan: 10 Februari 2020)

Abstrak

Internet of Things (IoT) telah memasuki berbagai aspek kehidupan manusia, diantaranya *smart city, smart home, smart street,* dan *smart industry* yang memanfaatkan internet untuk memantau informasi yang dibutuhkan. Meskipun sudah dienkripsi dan diautentikasi, protokol jaringan IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) yang dapat menghubungkan benda-benda yang terbatas sumber daya di IoT masih belum dapat diandalkan. Hal ini dikarenakan benda-benda tersebut masih dapat terpapar oleh *routing attacks* yang berasal dari jaringan 6LoWPAN dan internet. Makalah ini menyajikan kinerja *Smart Intrusion Detection System* berdasarkan *Compression Header Analyzer* untuk menganalisis model *routing attacks* lainnya pada jaringan IoT. IDS menggunakan *compression header* 6LoWPAN sebagai fitur untuk *machine learning algorithm* dalam mempelajari jenis *routing attacks*. Skenario simulasi dikembangkan untuk mendeteksi *routing attacks* berupa *selective forwarding attack* dan *sinkhole attack*. Pengujian dilakukan menggunakan *feature selection* dan *machine learning algorithm*. *Feature selection* digunakan untuk menentukan fitur signifikan yang dapat membedakan antara aktivitas normal dan abnormal. Sementara *machine learning algorithm* digunakan untuk mengklasifikasikan *routing attacks* pada jaringan IoT. Ada tujuh *machine learning algorithm* yang digunakan dalam klasifikasi antara lain *Random Forest, Random Tree, J48, Bayes Net, JRip, SMO,* dan *Naive Bayes*. Hasil percobaan disajikan untuk menunjukkan kinerja *Smart Intrusion Detection System* berdasarkan *Compression Header Analyzer* dalam menganalisis *routing attacks*. Hasil evaluasi menunjukkan bahwa IDS ini dapat mendeteksi antara serangan dan *non*-serangan.

Kata kunci: *Compression Header, IoT, Machine Learning Algorithms, Routing Attacks, Smart IDS*

ROUTING ATTACKS DETECTION ON 6LoWPAN IN THE INTERNET OF THINGS USING SMART INTRUSION DETECTION SYSTEM

Abstract

Internet of Things (IoT) has entered various aspects of human life including smart city, smart home, smart street, and smart industries that use the internet to get the information they need. Even though it's encrypted and authenticated, Internet protocol IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) networks that can connect limited resources to IoT are still unreliable. This is because these objects can still be exposed to attacks from 6LoWPAN and the internet. This paper presents the performance of an Smart Intrusion Detection System based on Compression Header Analyzer to analyze other routing attack models on IoT networks. IDS uses a 6LoWPAN compression header as a feature for machine learning algorithms in learning the types of routing attacks. Simulation scenario was developed to detect routing attacks in the form of selective forwarding and sinkhole. Testing is done using the feature selection and machine learning algorithm. Feature selection is used to determine significant features that can distinguish between normal and abnormal activities. While machine learning algorithm is used to classify attacks on IoT networks. There were seven machine learning algorithms used in the classification including Random Forests, Random Trees, J48, Bayes Net, JRip, SMO, and Naive Bayes. Experiment Results to show the results of the Smart Intrusion Detection System based on Compression Header Analyzer in analyzing routing attacks. The evaluation results show that this IDS can protect between attacks and non-attacks.

Keywords: *Compression Header, IoT, Machine Learning Algorithms, Routing Attacks, Smart IDS*

1. PENDAHULUAN

Internet of Things (IoT) merupakan suatu konsep teknologi yang bertujuan untuk memperluas manfaat dari konekvitas internet yang ada saat ini. *Internet of Thing* memungkinkan pertukaran data melalui ketersediaan semua objek yang berkaitan dengan pengguna (Kim and Lee, 2017). Dengan adanya IoT, meningkatkan penggunaan benda-benda fisik menjadi objek digital, mulai dari *smart city, smart home, smart street*, dan *smart industry*, yang menggunakan internet untuk memantau informasi yang dibutuhkan manusia (Sonar and Upadhyay, 2016).

Internet of Things menyediakan kemampuan bagi manusia dan komputer untuk belajar dan berkomunikasi dari miliaran hal yang meliputi sensor, aktuator, layanan dan objek yang terhubung ke internet lainnya (Ngu *et al.*, 2016). Teknologi utama dalam realisasi sistem IoT adalah *middleware*, yang biasanya digambarkan sebagai sistem perangkat lunak yang dirancang untuk menjadi perantara antara perangkat dan aplikasi IoT (Ngu *et al.*, 2016). *Middleware* memainkan peran penting karena bertanggung jawab atas sebagian besar kecerdasan dalam IoT, mengintegrasikan data dari perangkat, memungkinkan perangkat untuk berkomunikasi, dan membuat keputusan berdasarkan data yang dikumpulkan (Cruz *et al.*, 2018).

Sistem IoT sendiri mengacu pada penggunaan protokol internet standar untuk komunikasi manusia ke objek dan objek ke objek dalam jaringan tertanam (Heer *et al.*, 2011). Protokol internet pada objek di IoT menggunakan IPv6 untuk mengakomodasi ruang di internet, karena ruang alamat pada IPv4 terbatas (Pavan Pongle, 2015). Dalam buku (Shelby & Bormann, 2009) menyatakan bahwa protokol internet tradisional memerlukan panjang *frame* yang cukup dan membutuhka*n bandwidth* besar untuk aplikasi berat. Sehingga membuat *Internet Engineering Task Force* (IETF) menciptakan 6LoWPAN untuk mengatasi masalah ini dan memungkinkan IPv6 untuk dapat digunakan dengan perangkat yang terbatas di jaringan IoT.

Beberapa penelitian telah menyelidiki keamanan pesan untuk IoT, yaitu menggunakan *lightweight* DTLS (Kothmayr, 2011), *IPsec* (Raza, S., Duquennoy, 2011), dan keamanan lapisan tautan IEEE 802.15.4 (Kamesh & Sakthi Priya, 2014). Namun, meskipun jaringan IoT sudah menggunakan enkripsi dan otentikasi untuk keamanan pesan, *routing attacks* masih dapat dipengaruhi oleh jaringan 6LoWPAN dan internet (Wallgren, Raza & Voigt, 2013). Serangan yang beroperasi di lapisan jaringan disebut dengan *routing attacks* (Amish and Vaghela, 2016). *Routing attacks* meliputi *wormhole, sinkhole, blackhole, spoofing*, *hello flood, selective forwarding* dan sebagainya (CHELLI, 2015).

Oleh karena itu, *Intrusion Detection System* (IDS) perlu diimplementasikan dalam menangani serangan yang mengganggu jaringan IoT (Raza, Wallgren & Voigt, 2013). IDS adalah aplikasi perangkat lunak yang dapat memantau aktivitas di jaringan dan dapat melaporkan ke sistem manajemen (Jabez & Muthukumar, 2015)(Pharate, 2015). Ada dua kelas penting dalam IDS, yaitu IDS berbasis *signature* dan IDS berbasis *anomaly*. Sedangkan IDS berbasis *hybrid* adalah kombinasi dari *signature* dan *anomaly* (Alrajeh, Khan & Shams, 2013).

Penelitian yang dilakukan oleh (Aydin, Zaim & Ceylan, 2009) mengusulkan IDS berbasis *hybrid* dengan menggabungkan *Packet Header Anomaly Detection* (PHAD) dan *Network Traffic Anomaly Detection* (NETAD) menggunakan SNORT. Model ini tergantung pada perubahan cepat dalam statistik jaringan dalam jangka pendek.

Raza, Wallgren & Voigt (2013) mengusulkan IDS baru untuk IoT yang disebut SVELTE. Target SVELTE adalah *routing attacks* yang meliputi *sinkhole, selective forwarding, spoofing* dan dapat diperluas untuk mendeteksi serangan lain. SVELTE menggunakan RPL yang memiliki dua komponen utama, yaitu 6LoWPAN *Mapper (6Mapper)* dan modul deteksi intrusi. SVELTE juga menggunakan IDS berbasis *hybrid*. Selain itu, Pongle & Chavan (2015) mengusulkan IDS untuk IoT yang mampu mendeteksi *wormhole attack*. Metode yang diusulkan menggunakan informasi lokasi *node* dan informasi *node neighbor* untuk mengidentifikasi *wormhole attack*. Penerapan IDS berbasis *hybrid system*, dimana modul dipusatkan pada *IPv6 border router* (6BR) dan modul yang berpusat pada *node* sensor bekerja bersama untuk mendeteksi serangan.

Namun, SVELTE (Raza, Wallgren & Voigt, 2013) dan IDS (Pongle & Chavan, 2015) kurang efektif dalam menentukan serangan kompleks, karena mereka hanya mendeteksi serangan tertentu. Kemudian Napiah *et al.*, (2018) mengusulkan *Compression Header Analyzer Intrusion Detection System* (CHA-IDS) yang dapat menganalisis *compression header* 6LoWPAN untuk mengurangi berbagai serangan individu dan kombinasi yang ditemukan pada jaringan IoT. CHA-IDS menggunakan *feature selection* untuk memilih fitur signifikan dan *machine learning algorithm* untuk mengklasifikasikan antara serangan dan *non*-serangan.

Menurut Jabez & Muthukumar (2015), *machine learning* dapat digunakan untuk mendeteksi dan mengklasifikasi serangan pada jaringan. *Machine learning* telah dikembangkan untuk menemukan algoritma terbaik dalam mesin pendeteksi pada IDS. Dalam penelitian (Meenakshi & Geetika, 2014) perbandingan berbagai teknik klasifikasi dilakukan menggunakan *software Weka*. Penelitian ini bertujuan untuk menyelidiki kinerja berbagai metode klasifikasi pada suatu data.

Penelitian yang dilakukan oleh Kolias *et al.*, (2016) menggunakan beberapa *machine learning alorithm* untuk menganalisis 156 fitur yang dikumpulkan dari jaringan 802.11. Pemilihan fitur

ini dilakukan untuk memilih hanya fitur penting yang digunakan untuk mengurangi konsumsi sumber daya perangkat. Penelitian tentang pemilihan fitur juga dilakukan oleh Kang, (2015). Dalam penelitian tersebut mengusulkan algoritma pemilihan fitur berdasarkan pada algoritma pencarian lokal untuk memilih *subset* fitur yang optimal untuk IDS dalam mendeteksi serangan DoS. Selain itu, ada penelitian (Desale, Ketan Sanjay & Ade, 2015) yang mengusulkan pendekatan evolusi untuk pemilihan fitur berdasarkan prinsip-prinsip matematika ketika memilih fitur dari *dataset* NSL-KDD.

Pada saat ini, telah dikembangkan model CHA-IDS yang dapat menganalisis model *routing attacks*. Akan tetapi, CHA-IDS belum diselidiki lebih lanjut untuk mendeteksi tipe *routing attacks* yang lainnya. Penelitian ini akan membahas mengenai kinerja *Smart Intrusion Detection System* yang mengacu pada *Compression Header Analyzer* dalam menganalisis tipe *routing attacks* lainnya yang ada dalam jaringan IoT. IDS menggunakan *compression header* 6LoWPAN yang dapat membedakan aktivitas normal dan abnormal. *Feature selection* dan *machine learning algorithm* digunakan untuk mendeteksi *routing attacks* pada jaringan IoT. Simulasi *routing attacks* menggunakan *Cooja Simulator* dan diuji menggunakan *software Wireshark*. *Software Weka* digunakan untuk klasifikasi antara serangan dan non-serangan di IoT.

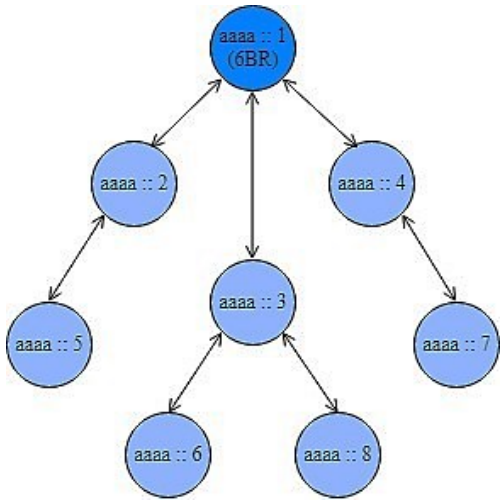
2. METODE PENELITIAN

Pada bagian ini, metode *Smart Intrusion Detection System* akan disajikan dalam empat bagian. Bagian pertama menjelaskan informasi *routing* pada *compression header* 6LoWPAN. Bagian kedua menjelaskan struktur metode *Smart IDS Compression Header* yang terdiri dari tiga *layer*. Kemudian bagian ketiga menyajikan evaluasi metode *Smart Intrusion Detection System*. Bagian keempat menyajikan evaluasi mengenai *Smart Intrusion Detection System*. Empat bagian dijelaskan secara lebih rinci sebagai berikut:

2.1 Compression Header 6LoWPAN

6LoWPAN merupakan perluasan dari IPv6 yang efisien kedalam lingkungan nirkabel, sehingga memungkinkan jaringan dan fitur IP *end-to-end* untuk dapat berbagi aplikasi *embedded* (Shelby & Bormann, 2009). Untuk dapat menghubungkan semua *node* perangkat ke internet, terdapat *gateway* yang dikenal sebagi *6LoWPAN Border Router* (6BR) yang serupa dengan *node sink* di jaringan WSN. Dalam *compression header*, 6LoWPAN mendefinisikan HC1 *encoding* sebagai skema kompresi yang dioptimalkan untuk komunikasi *link-local* IPv6. *Compression header* merupakan parameter penting untuk mengurangi biaya *overhead* di 6LoWPAN (Shah, Shrimali and Parikh, 2016).

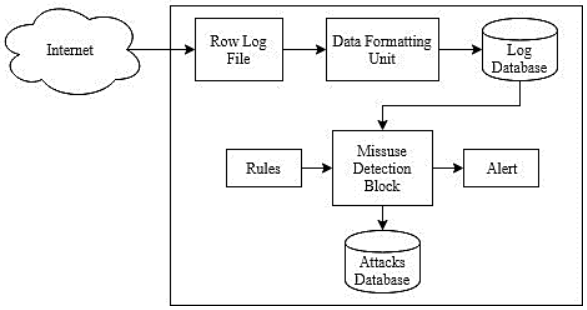
Protokol *routing* untuk IoT sendiri disebut dengan *IPv6 Routing Protocol for Low-Power and Lossy Network* (RPL) (Wallgren, Raza & Voigt, 2013)(Raza, Wallgren & Voigt, 2013). RPL membuat *Destination-Oriented Directed Acyclic Graph* (DODAG) antara *node* pada jaringan 6LoWPAN. Gambar 1 (Wallgren, Raza & Voigt, 2013) menunjukkan contoh DODAG untuk setiap *node* yang memiliki alamat IPv6 yang unik, seperti aaaa :: 1.



Gambar 1. Contoh DODAG untuk node yang memiliki alamat IPv6 unik

Informasi *routing* di *compression header* 6LoWPAN sebanyak 77 yang dapat ditemukan di *Wieshark*. Informasi *routing* pada 6LoWPAN memiliki potensi yang dapat digunakan sebagai fitur untuk membedakan antara aktivitas normal dan abnormal.

2.2 Arsitektur *Intrusion Detection System* (IDS)



Gambar 2. Diagram Blok IDS

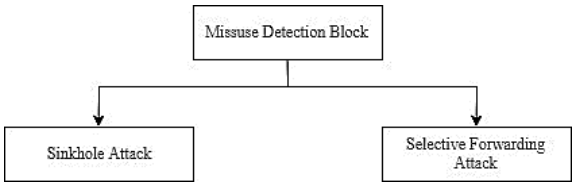
Gambar 2 (Ambhore, 2014) menunjukkan diagram blok dari *Intrusion Detection System*. Terdiri dari blok-blok berikut:

1. *Row Log File*: mengumpulkan paket *header* yang berasal dari internet.
2. *Data Formatting Unit*: data dari *log file* diklasifikasikan menurut bidang dalam paket *header*. Protokol yang digunakan diidentifikasi menggunakan bidang atau nilai tertentu.
3. *Log Database*: berisi tabel yang berbeda-beda sesuai dengan protokol yang digunakan. Tabel berisi atribut protokol.
4. *Misuse Detection Block*: mendeteksi serangan yang diketahui sesuai dengan aturan yang telah

ditentukan. Jika sesuai, IDS akan menyatakan sebagai gangguan dan memberitahu admin.

5. *Attacks Database*: berisi tabel untuk *database* serangan.
6. *Rules*: berisi aturan yang digunakan IDS dalam menganalisis aktivitas di jaringan.
7. *Alerts*: memberikan aksi dalam bentuk peringatan kepada pengguna jika ada serangan. Tindakan ini dilakukan oleh admin.

Misuse Detection Block (Singh, 2013) adalah bagian dari IDS yang berguna untuk mendeteksi serangan yang diketahui sesuai dengan aturan yang telah ditentukan. Jika sesuai, IDS akan menyatakan sebagai gangguan dan melaporkan ke admin. Dalam penelitian ini, *misuse detection* terdapat dua *routing attacks* dalam bentuk *selective forwarding* dan *sinkhole*. Gambar 3 menunjukkan diagram blok *misuse detection*.



Gambar 3. Diagram Blok *Missue Detection*

2.3 Struktur Metode *Smart Intrusion Detection System*

Dalam penelitian ini, IDS didasarkan pada *hybrid* yang menerapkan *signature* dan *anomaly*. IDS menggunakan *compression header* 6LoWPAN sebagai fitur yang digunakan oleh *machine learning algorithm* dalam mengklasifikasikan jenis *routing attacks*. Aturan yang dibuat oleh *machine learning algorithm* ditempatkan pada *6LoWPAN Border Router* (6BR). Gambar 4 (Wallgren, Raza & Voigt, 2013) menunjukkan interkoneksi objek dalam jaringan 6LoWPAN yang terhubung ke Internet menggunakan 6BR.

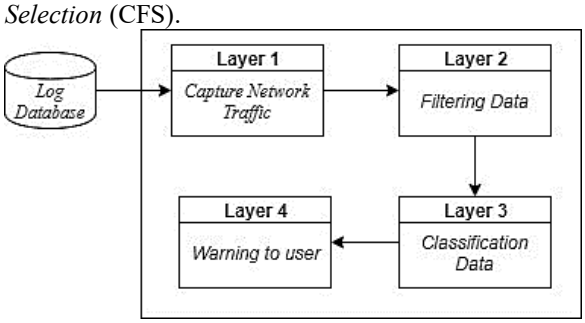
Gambar 5 (Napiah *et al.*, 2018) menunjukkan struktur metode *Smart Intrusion Detection System* berbasis *Compression Header Analyzer*. Terdiri dari tiga *layer* sebagaai berikut:

1. *Layer 1, Sensor Agent* (SA)

Layer SA ini bertujuan untuk menangkap lalu lintas jaringan. Cara yang digunakan adalah mengumpulkan paket data yang diterima dari semua *node* dalam jaringan. *Cooja Traffic Analyzer*, terutama untuk 6LoWPAN digunakan untuk menangkap pesan radio dan disimpan dalam file *packet capture (pcap)*.

2. *Layer 2, Aggregator Agent* (AGA)

Pada layer ini, aktivitas normal dan abnormal yang ditemukan dalam *file pcap* dari *layer* SA di *filter*. Tujuan dari pemfilteran *file* adalah untuk memilih data yang hanya terkait dengan protokol 6LoWPAN. Kemudian informasi *routing* disimpan lagi dalam *format Comma Separated Value (CSV)*. *Search algorithm* dan *feature selection* digunakan untuk memproses *file CSV*. Dalam melakukan pencarian fitur yang signifikan menggunakan algoritma *Best First Search* (BFS) dan *Greedy Stepwise* (GS). Sedangkan untuk mengevaluasi fitur signifikan menggunakan *Correlation Feature Selection* (CFS).

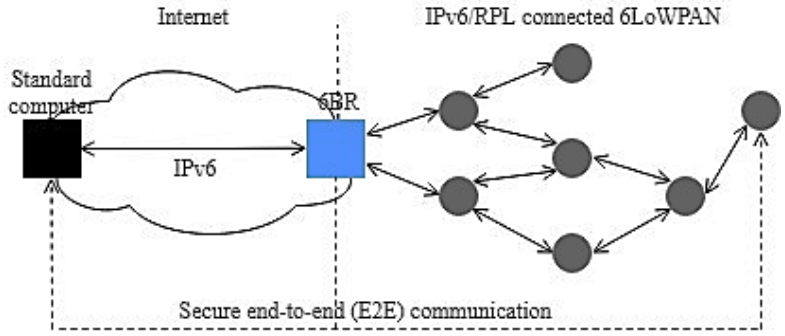


Gambar 5. Kerangka IDS *Compression Header*

CFS mengevaluasi kelayakan subset fitur menggunakan metode *heuristik*. Setiap atribut *subset* dipilih dengan mempertimbangkan tingkat redundansi antara fitur yang ada dan kemampuan prediksi masing-masing fitur. Fungsi untuk mengevaluasi fitur terbaik terdapat pada (Kang, 2015)(Hall, 1999) ditunjukkan dalam persamaan (1):

$$Merit_s = \frac{\overline{kr}_{cf}}{\sqrt{k+k(k-1)\overline{rff}}} \quad (1)$$

Dimana $Merit_s$ adalah *heuristik "merit"* dari *subset* fitur yang berisi fitur k, rfc adalah korelasi kelas fitur rata-rata, dan rff adalah fitur rata-rata antar-korelasi (Kang, 2015)(Hall, 1999).



Gambar 4. Interkoneksi objek yang terhubung dengan IPv6/RPL di 6LoWPAN

Salah satu strategi pencarian sederhana, disebut dengan *greedy hill climbing*. Perubahan lokal dibuat hanya dengan menambahkan atau menghapus satu fitur dari *subset*. *Search algorithm* dapat memilih yang terbaik dengan mempertimbangkan perubahan lokal dalam suatu himpunan bagian, atau hanya memilih perubahan pertama yang meningkatkan manfaat dari himpunan bagian fitur saat ini (Hall, 1999). Algoritma 1 menunjukkan algoritma *greedy hill search*.

BFS adalah strategi pencarian AI. Seperti *hill climbing*, langkah *best first* adalah membuat perubahan lokal ke *subset* fitur pada saat itu melalui ruang pencarian. Namun, jika jalan yang dilalui mulai terlihat kurang menjanjikan, BFS dapat kembali ke bagian sebelumnya yang lebih baik dan melanjutkan pencarian dari sana, tidak seperti *hill climbing* (Hall, 1999). Algoritma 2 menunjukkan algoritma *best first search*.

Algoritma 2: ***Best First Search Algorithm***

1. *Begin with the OPEN list containing the start state, the CLOSED list empty, and BEST ← start state.*
2. *Let s = arg max e(μ) (get the state from OPEN with the highest evaluation).*
3. *Remove s from OPEN and add to CLOSED.*
4. *If e(s) ≥ e(BEST), then BEST ← s.*
5. *For each child t of s that is not in the OPEN or CLOSED list, evaluate and add to OPEN.*
6. *If BEST changed in the last set of expansions, go to 2.*
7. *Return BEST*

Dalam penelitian ini, algoritma BFS dan GS digunakan untuk melakukan pencarian fitur yang signifikan, sementara CFS digunakan untuk mengevaluasi fitur signifikan yang dapat membedakan antara aktivitas normal dan abnormal dalam jaringan.

3. *Layer 3, Analyzer Agent* (ANA)

Di *layer* ini, ada dua *routing attacks* pada IoT yang diidentifikasi, yaitu *selective forwarding* dan *sinkhole*. Paket data diplot terhadap waktu yang telah ditentukan. Kemudian setiap kelas dibandingkan dengan menggunakan *machine learning algorithm* yang terdiri dari *Random Forest, Random Tree, J48, Bayes Net, JRip, SMO,* dan *Naïve Bayes* yang terdapat dalam *software Weka*.

4. *Layer 4, Actuator Agent* (ACA)

Layer ACA akan mengambil tindakan dengan memberikan peringatan kepada pengguna. Nilai *threshold* akan dibandingkan dari respons ACA terhadap proses ANA. Jika hasil dalam proses ANA melebihi nilai *threshold*, maka akan memicu alarm. Nilai *threshold* diatur 10% untuk mengurangi kesalahan alarm.

2.4 Evaluasi Kerangka *Smart Intrusion Detection System*

Pada bagian ini, tujuh *machine learning algorithm* digunakan (*Random Forest, Random Tree, J48, Bayes Net, JRip, SMO, Naive Bayes)* dan dievaluasi untuk menemukan algoritma terbaik dalam mengklasifikasikan *routing attacks* menggunakan *software Weka*.

Confusion Matrix adalah tabel untuk memvisualisasikan kinerja suatu algoritma. Ada empat faktor pengukuran dalam *confusion matrix*, yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Sahu & Mehtre, 2015).

1. *True Positive* (TP): menunjukkan jumlah

Algoritma 1: ***Greedy Hill Search Algorithm***

1. *Let s ← start state.*
2. *Expand s by making each possible local change.*
3. *Evaluate each child t of s.*
4. *Let s' ← child t with the highest evaluation e(t).*
5. *If e(s') ≥ e(s) then s ← s', go to 2.*
6. *Return s.*

2. prediksi yang benar dalam sebuah contoh milik kelas yang sama.
3. *True Negative* (TN): menunjukkan jumlah prediksi salah dalam contoh milik kelas lain.
4. *False Positive* (FP): menunjukkan jumlah prediksi yang benar termasuk dalam kelas yang sama.
5. *False Negative* (FN): menunjukkan jumlah prediksi yang salah dari contoh milik kelas lain.

Rumus untuk klasifikasi *machine learning algorithm* yang ditemukan dalam *tool Weka* terdapat pada penelitian (Napiah *et al.*, 2018) dalam bentuk *True Positive* (TP), *False Positive* (FP), *Precision, Recall, Mean Absolute Error* (MAE), *Root Means Squared Error* (RMSE), *Relative Absolute Error* (RAE) dihitung dan dinyatakan dalam rumus persamaan (2 sampai 8) sebagai berikut:

$$TP_{rate} = \frac{TP}{TP+FN} \quad (2)$$

$$FP_{rate} = \frac{FP}{TN+FP} \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|yi - \hat{y}i| \quad (6)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2} \quad (7)$$

$$RAE_i = \frac{\sum_{j=1}^{n}|P_{ij} - T_j|}{\sum_{j=1}^{n}|T_j - \bar{T}|} \text{ dimana } \bar{T} = \frac{1}{n}\sum_{j=1}^{n} T_j \quad (8)$$

Selain itu, *machine learning algorithm* menggunakan akurasi untuk mengukur tingkat deteksi *routing attacks* di jaringan IoT. Akurasi dihitung dengan menggunakan *confusion matrix* seperti dalam penelitian (Sahu & Mehtre, 2015) yang ditunjukkan dalam persamaan (9) yaitu:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

3. HASIL DAN PEMBAHASAN

a. Simulasi *Routing Attacks*

Jaringan *Contiki* dengan *Cooja Simulator* digunakan untuk meluncurkan dua *routing attacks* pada IoT, yaitu *selective forwarding* dan *sinkhole*. Dalam membuat *node client* menggunakan *Tmote sky* dan untuk 6BR atau *node sink* menggunakan *Cooja* dalam simulasi. Serangan disimulasikan secara individual dan ada dua bagian:

1. Simulasi dijalankan selama 30 menit. Pada tahap ini, komunikasi pesan radio untuk setiap *node* dicatat. Ini dimaksudkan sebagai pengumpulan data untuk mengetahui fitur-fitur yang dapat digunakan dalam membuat *database*.
2. Dalam setiap serangan, *dataset* dikumpulkan pada menit ke 10, 20 dan 30 selama simulasi.

Pada awal simulasi, jaringan 6LoWPAN dibuat tanpa *node* serangan atau dalam keadaan normal. Kemudian untuk kondisi abnormal, dibuat dengan *node* serangan. Serangan diluncurkan dalam bentuk *selective forwarding* dan *sinkhole* secara bersamaan. Jaringan dibiarkan acak untuk membuat pola serangan baru untuk anomali jaringan.

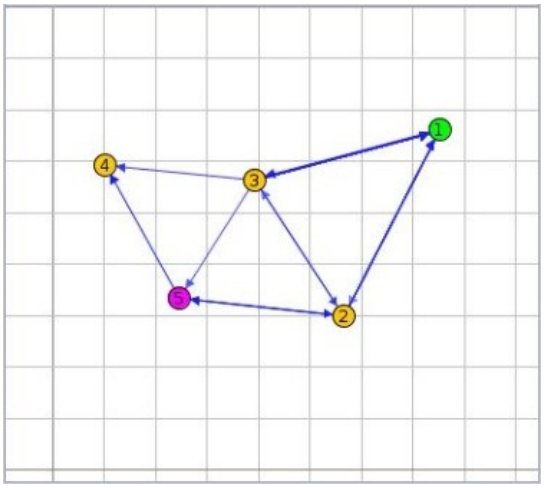
Terdapat dua *routing attacks* pada simulasi yaitu *selective forwarding attack* dan *sinkhole attack*.

1. *Selective Forwarding Attack*

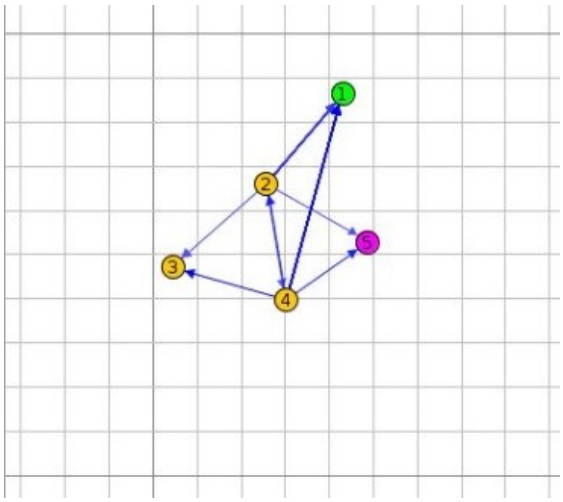
Dalam s*elective forwarding attack*, memungkinkan untuk meluncurkan serangan DoS. Dimana *node attack* secara selektif meneruskan paket. Serangan ini ditargetkan untuk mengganggu jalur perutean, tetapi juga digunakan untuk menyaring protokol apa pun. Penyerang dapat meneruskan semua pesan kontrol RPL, dan membuang sisa lalu lintasnya. Gambar 6 menunjukkan *selective forwarding attack* yang terdiri dari *node sink, node client* dan *node attack*. *Node sink* ditunjukkan dengan angka 1, yang berwarna hijau. *Node client* ditunjukkan dengan angka 2, 3, 4 berwarna kuning. Kemudian *node attack* berwarna ungu dengan ditunjukkan angka 5. Lalu lintas jaringan ditunjukkan oleh panah diantara *node-node* tersebut. Sebagai contoh, *node sink* 1 menerima pesan dari *node client* 2 atau sebaliknya. Namun, jika terdapat *node client* di dekat *node attack*, pesan akan disaring oleh penyerang dan akan diteruskan ke *node* lain yang digunakan untuk mengganggu lalu lintas di sekitar *node sink* dan *node client*.

2. *Sinkhole Attack*

Dalam *sinkhole attack*, penyerang mengiklankan jalur perutean yang menyebabkan banyak *node* merutekan lalu lintas melaluinya. Dalam RPL, penyerang dapat meluncurkan *sinkhole* dengan mengiklankan peringkat yang lebih baik, sehingga *node* dalam DODAG memilihnya sebagai *node* induk. Gambar 7 menunjukkan simulasi *sinkhole* dengan *node* 1 sebagai *node sink* (6BR) berwarna hijau, *node client* (2,3,4) berwarna kuning, dan 5 sebagai *node attack* berwarna ungu. Ketika lalu lintas antara *node sink* dan *node client* sedang berjalan, *node attack* mengiklankan peringkat yang lebih baik daripada *node* lainnya, yang membuat *node* lain memilih *node attack* sebagai *node* induk. Sehingga membuat lalu lintas di jaringan tidak terkendali atau berantakan. Lalu lintas jaringan ditunjukkan oleh panah antara *node sink*, *node client* dan *node attack*. Selain itu, panah menunjukkan komunikasi pesan yang dikirim dan diterima antara *nod*e yang ada.



Gambar 6. *Selective Forwarding Attack*

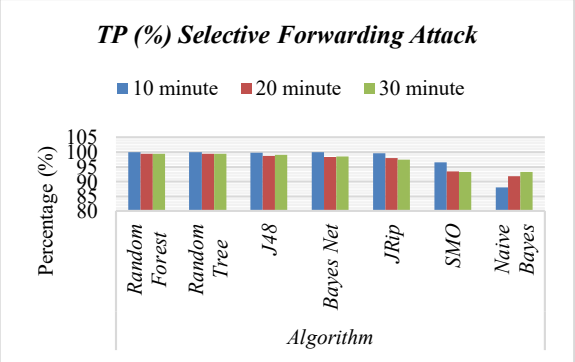


Gambar 7. *Sinkhole Attack*

b. Klasifikasi *Routing Attacks*

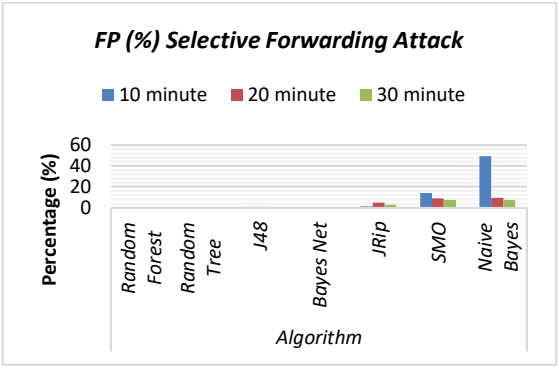
Berikut ini adalah analisis dari tujuh *machine learning algorithm* (*Random Forest, Random Tree,* J48*, Bayes Net, JRip,* SMO*,* dan *Naive Bayes*) yang digunakan untuk mengklasifikasikan *routing attacks*. Hasil kinerja dari tujuh *machine learning algorithm* untuk serangan individu ditunjukkan pada Gambar 8 dan Gambar 9 untuk *selective forwarding attack*,

sedangkan Gambar 10 dan Gambar 11 untuk *sinkhole attack*.



Gambar 8. Nilai TP *Selective Forwarding Attack*

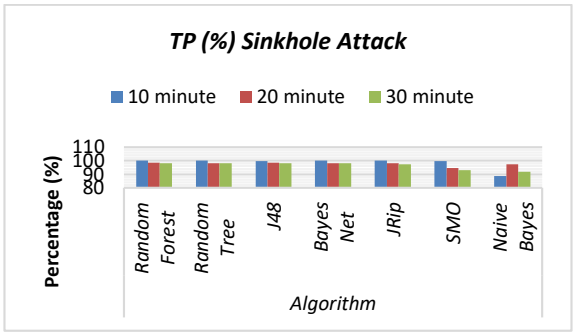
Gambar 8 menunjukkan nilai TP dari *selective forwarding attack*, dimana terdapat *machine learning algorithm* yang digunakan dalam klasifikasi serangan pada setiap menit simulasi. *Random Forest* memiliki nilai TP 100% pada menit 10, sedangkan untuk menit 20 dan 30 nilai TP turun di kisaran 99,5%. *Random Tree* memiliki nilai TP yang sama dengan *Random Forest* selama simulasi. Kemudian ada J48 yang memiliki nilai TP hampir 100%, dan memiliki selisih nilai yang kecil dengan *Random Forest* dan *Random Tree*. Berikutnya ada *Bayes Net, Jrip,* dan SMO. Pada posisi terakhir terdapat *Naive Bayes*, dengan nilai TP setiap menit simulasi meningkat. Namun, nilai TP masih jauh dari 100% dengan kisaran nilai di 88% di menit 10, 91,9% di menit 20, dan 93,3% di menit 30.



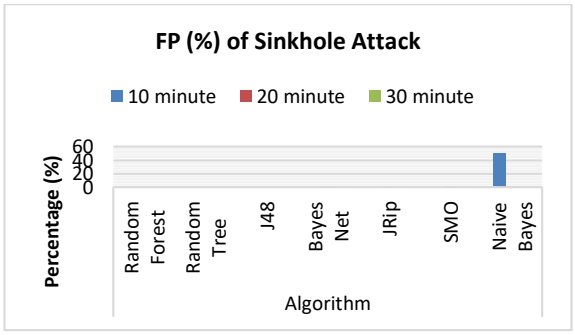
Gambar 9. Nilai FP *Selective Forwarding Attack*

Gambar 9 menunjukkan nilai FP tertinggi diperoleh oleh *Naive Bayes* pada menit ke-10 yaitu 49,5%. Kemudian nilai FP *Naive Bayes* turun menjadi 9,3% di menit ke-20, dan 7,5% di menit ke-30. SMO berada di posisi kedua, dengan nilai FP selama simulasi juga menurun, tetapi tidak terlalu signifikan. *JRip* memiliki nilai FP di bawah kisaran 5%, diikuti oleh *Bayes Net* dan J48. Sedangkan untuk *Random Tree* dan *Random Forest* tidak memiliki nilai FP, karena nilai FP di kedua algoritma adalah 0%.

Gambar 10 menunjukkan nilai TP tertinggi diperoleh oleh *Random Forest* dengan nilai 99,9% pada menit ke-10, 98,4% pada menit ke-20, dan 98,3% pada menit ke-30. Kemudian diikuti oleh *Random Tree* dengan nilai TP hampir mendekati *Random Forest*, dengan perbedaan nilai hanya 0,1% pada menit ke-20 dan ke-30. Sedangkan pada menit ke 10, nilai TP pada *Random Tree* sama dengan *Random Forest*. J48 di posisi berikutnya, yang memiliki selisih nilai 0,1% dengan *Random Tree*. *Bayes Net* dan *JRip* memiliki nilai TP yang sama pada menit ke-10 dan 20 simulasi, yaitu 99,9% dan 98,1%. Pada menit ke-30, *Bayest Net* dan *JRip* memiliki selisih 1%. Posisi berikutnya adalah SMO, diikuti oleh *Naive Bayes* di posisi terakhir.



Gambar 10. Nilai TP *Sinkhole Attack*



Gambar 11. Nilai FP *Sinkhole Attack*

Gambar 11 menunjukkan nilai FP *Random Forest, Random Tree,* dan J48 yaitu 0% atau tidak memiliki nilai FP selama simulasi. *Bayes Net, JRip*, dan SMO juga memiliki nilai FP 0% di menit ke-10. Namun, nilai FP pada menit berikutnya mengalami sedikit peningkatan untuk ketiga algoritma tersebut. Untuk *Bayes Net* dan SMO, simulasi pada menit ke-20 dan ke-30 memiliki nilai FP yang sama yaitu 0,1% dan 0,3%. *Naive Bayes* memiliki nilai FP tertinggi pada menit ke-10 sebesar 49,6%. Kemudian nilai FP turun sangat signifikan menjadi 0,1% di menit ke-20, dan 0,2% di menit ke-30.

Dari nilai TP dan FP untuk *selective forwarding* dan *sinkhole attack*, dapat disimpulkan bahwa *Random Forest* memiliki nilai TP tertinggi, dan memiliki nilai FP terendah selama simulasi. Sedangkan *Naive Bayes* yang berbanding terbalik dengan *Random Forest*. *Naive Bayes* memiliki nilai TP terendah, dan memiliki nilai FP tertinggi dibandingkan dengan algoritma lainnya.

Selain nilai TP dan nilai FP, penelitian ini juga memperhitungkan tingkat *accuracy, precision* dan

recall yang ditunjukkan pada Tabel 1 untuk *selective forwarding attack* dan Tabel 2 untuk *sinkhole attack*. Hasil untuk kedua *routing attacks* diperoleh dari simulasi selama 30 menit.

Tabel 1. Klasifikasi *Selective Forwarding Attack*

Algoritma	*TP rate*	*FP rate*	*Accuracy %*	*Precision*	*Recall*
Random Forest	0,995	0,000	99,5452	0,996	0,995
Random Tree	0,995	0,000	99,4543	0,995	0,995
J48	0,991	0,003	99,0905	0,992	0,991
Bayes Net	0,986	0,001	98,6357	0,986	0,986
Jrip	0,975	0,031	97,4989	0,976	0,975
SMO	0,933	0,077	93,3151	0,879	0,933
Naïve Bayes	0,933	0,075	93,3151	0,900	0,933

Tabel 2. Klasifikasi *Sinkhole Attack*

Algoritma	*TP rate*	*FP rate*	*Accuracy %*	*Precision*	*Recall*
Random Forest	0,983	0,000	98,2797	0,983	0,983
Random Tree	0,982	0,000	98,2423	0,982	0,982
J48	0,982	0,000	98,2423	0,982	0,982
Bayes Net	0,982	0,001	98,1675	0,979	0,982
Jrip	0,972	0,013	97,1578	0,970	0,972
SMO	0,931	0,003	93,1189	0,904	0,931
Naïve Bayes	0,920	0,002	92,0344	0,908	0,920

Tabel 3. Evaluasi Kombinasi Dataset

Algoritma	*Accuracy %*	*TP rate*	*FP rate*	*MAE*	*RMSE*	*RAE %*
Random Forest	99,4721	0,995	0,000	0,0008	0,0207	1,668
Random Tree	99,4281	0,994	0,000	0,0006	0,0243	1,2724
J48	99,4281	0,994	0,000	0,0011	0,0239	2,2935
Bayes Net	98,7681	0,988	0,001	0,002	0,0296	4,2368
Jrip	98,4162	0,984	0,007	0,002	0,0396	4,1629
SMO	96,3044	0,963	0,036	0,0942	0,213	195,6737
Naive Bayes	95,2046	0,952	0,001	0,0057	0,055	11,7472

Pada Tabel 1, *Random Forest* memiliki nilai terbaik untuk *precision, recall* dan *accuracy* yang hampir mencapai 100%. Kemudian diikuti oleh *Random Tree*, yang memiliki selisih nilai kecil untuk setiap data klasifikasi. J48 juga memiliki selisih nilai kecil dengan Random Tree, yang berada di tempat ketiga. *Bayes Net* dan *JRip* berada di posisi berikutnya, diikuti oleh SMO dan *Naive Bayes*. Pada saat pengujian, nilai SMO dan *Naive Bayes* sama yaitu 93,3151%.

Pada Tabel 2, *Random Forest* menunjukkan hasil terbaik dengan kisaran nilai 98,3% pada *precision, recall* dan *accuracy*. Selain itu, ada *Random Tree* dan J48, yang memiliki nilai yang sama pada tiga kategori klasifikasi, yaitu sebesar 98,2%. Selanjutnya ada *Bayes Net* dengan nilai *accuracy* 98,1675%, *Jrip* dengan nilai 97,1578%, SMO dengan nilai 93,1189%, dan di posisi terakhir ada *Naive Bayes* dengan nilai 92,0344%.

Dari dua tabel klasifikasi tersebut, dapat dilihat bahwa *Random Forest* memiliki nilai tertinggi pada *accuracy, precision* dan *recall*. Sementara *Random Tree* dan J48 berada di bawah *Random Forest* dengan perbedaan nilai yang sangat kecil. SMO dan *Naive Bayes* berada di posisi bawah, dengan nilai *accuracy, precision* dan *recall* yang sama untuk *selective forwarding attack* dan terdapat perbedaan nilai kecil untuk *sinkhole attack* pada ketiga kategori klasifikasi.

Evaluasi gabungan dari dua serangan individu juga dihitung dan ditunjukkan dalam Tabel 3.

Tabel 3 menunjukkan bahwa *Random Forest* memiliki nilai TP tertinggi sebesar 0,995 dengan tingkat akurasi sebesar 99,4721% dibandingkan dengan algoritma lainnya. *Random Tree* dan J48 berada di posisi yang sama, dengan nilai TP sebesar 0,994 dan tingkat akurasi sebesar 99,4281%. Kemudian terdapat *Bayes Net* dengan nilai TP yaitu 0,988, *JRip* mencapai nilai TP 0,984, dan nilai TP untuk SMO yaitu 0,963. Posisi terakhir ditempati oleh *Naïve Bayes* dengan nilai TP mencapai 0,952 dengan tingkat akurasi 95,2046%.

Kerangka kerja penelitian ini hampir sama dengan penelitain CHA-IDS yang diusulkan oleh Napiah *et al*. Pada CHA-IDS, sistem dievaluasi menggunakan tiga *routing attacks* berupa *hello flood, sinkhole*, dan *wormhole*. Sedangkan di dalam penelitian ini dievaluasi menggunakan varian baru model *routing attacks* lainnya, yaitu *selective forwarding* dan *sinkhole*.

Penelitian ini menggunakan tujuh *machine learning algorithm* berupa *Random Forest, Random Tree,* J48*, Bayes Net, JRip,* SMO*,* dan *Naive Bayes* dalam menentukkan tingkat akurasi setiap *routing attacks* yang ada. Sedangkan pada CHA-IDS menggunakan enam *machine leaning algorithm* berupa J48, *Logistic,* MLP, *Naive Bayes, Random Forest,* dan SMO.

Berdasarkan *machine learning algorithm* yang digunakan, hasil kinerja antara penelitian ini dengan CHA-IDS berbeda. Di dalam CHA-IDS, hasil kinerja terbaik dalam mengklasifikasikan antara serangan dan *non*-serangan ditunjukkan oleh J48 dengan tingkat akurasi sebesar 99,4444%, dengan nilai TP sebesar 0,994 dan nilai MAE sebesar 0,0041. Kemudian diikuti oleh *Random Forest*, *Naïve Bayes,* MLP, *Logistic*, dan yang terakhir SVM.

Pada penelitian ini, hasil kinerja terbaik dalam mengklasifikasikan *routing attacks* diperoleh oleh *Random Forest* dengan tingkat akurasi sebesar 99,4721%. Selain itu, nilai TP yang diperoleh oleh *Random Forest* sebesar 0,995 dan nilai MAE sebesar 0,0008. Pada posisi berikutnya ditempati oleh *Random Tree*, J48, *Bayes Net, Jrip*, SMO, dan pada posisi terakhir terdapat *Naïve Bayes*.

4. KESIMPULAN

Pekerjaan ini merupakan kelanjutan dari CHA-IDS (Napiah *et al.*, 2018). Dalam hal ini, simulasi diuji dengan varian baru *routing attacks*, yaitu *selective forwarding* dan *sinkhole*. BFS dan GS dengan CFS yang merupakan *search algorithm* digunakan untuk memilih fitur terbaik yang dapat dipelajari lebih lanjut oleh *machine learning algorithm*. *Random Forest, Random Tree,* J48*, Bayes Net, JRip,* SMO*,* dan *Naïve Bayes* merupakan *machine learning algorithm* yang digunakan untuk mengklasifikasikan antara serangan dan *non*-serangan. Dari tujuh *machine learning* yang digunakan, kinerja terbaik dalam mendeteksi *routing attacks* ditunjukkan oleh *Random Forest* dengan tingkat akurasi mencapai 99,4721%. Selain itu, *Random Forest* juga mencapai nilai TP tertinggi sebesar 0,995 dan memiliki nilai MAE 0,0008.

Dalam penelitian lebih lanjut, diharapkan bahwa *Smart Intrusion Detection System* dapat ditingkatkan untuk mendeteksi jenis *routing attacks* lainnya. Selain itu, tingkat akurasi deteksi *routing attacks* dapat dihitung dengan menggunakan model *Artificial Intelligence* (AI) lainnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada UEESRG, Pusat Kajian *Smart Clean Energy*, Jurusan Teknik Elektro, PUI *Energy Hybrid* BFSC LPPM, Universitas Negeri Semarang, yang telah mendukung dan memfasilitasi penulis dalam penelitian ini.

DAFTAR PUSTAKA

ALRAJEH, N. A., KHAN, S. dan SHAMS, B. , 2013. Intrusion detection systems in wireless sensor networks: A review. *International Journal of Distributed Sensor Networks*.

AMBHORE, P., 2014. International Journal of Advanced Research in Intrusion Detection System for Intranet Security.

AMISH, P. dan VAGHELA, V. B., 2016. Detection and Prevention of Wormhole Attack in Wireless Sensor Network using AOMDV Protocol. *Procedia Computer Science*, 79, pp. 700–707, Elsevier Masson SAS.

AYDIN, M. A., ZAIM, A. H. dan CEYLAN, K. G., 2009. A hybrid intrusion detection system design for computer network security. *Computers and Electrical Engineering*, 35(3), pp. 517–526. Elsevier Ltd.

CHELLI, K., 2015. Security Issues in Wireless Sensor Networks: Attacks and Countermeasures.

CRUZ, M. A. A. *et al,*. 2018. A Reference Model for Internet of Things Middleware. 5(2), pp. 871–883.

DESALE, KETAN SANJAY dan ADE, R., 2015. Genetic Algorithm based Feature Selection Approach for Effective Intrusion Detection System. *2015 International Conference on Computer Communication and Informatics*.

HALL, M., 1999. Correlation-based Feature Selection for Machine Learning. 21i195-i20(April), pp. 1–5. *Methodology*.

HEER, T. *et al,*. 2011. Security Challenges in the IP-based Internet of Things. pp. 527–542.

JABEZ, J. dan MUTHUKUMAR, B., 2015. Intrusion detection system (ids): Anomaly detection using outlier detection approach. *Procedia Computer Science*, 48(C), pp. 338–346. Elsevier Masson SAS.

KAMESH dan SAKTHI PRIYA, N., 2014. Security enhancement of authenticated RFID generation. *International Journal of Applied Engineering Research*, 9(22), pp. 5968–5974.

KANG, S., 2015. A Feature Selection Algorithm to Find Optimal Feature Subsets for Detecting DoS Attacks. pp. 1–3.

KIM, S. dan LEE, I., 2017. IoT device security based on proxy re-encryption. *Journal of Ambient Intelligence and Humanized Computing*, 0(0), p. 0. Springer Berlin Heidelberg.

KOLIAS, C. *et al,*. 2016. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys and Tutorials*, 18(1), pp. 184–208.

KOTHMAYR, T., 2011. A Security Architecture for Wireless Sensor Networks based on DTLS.

MEENAKSHI, M. dan GEETIKA, G., 2014. Survey on Classification Methods using WEKA. *India International Journal of Computer Applications*, 86(18), pp. 16–19.

NAPIAH, M. N. *et al,*. 2018. Compression Header Analyzer Intrusion Detection System (CHA

-IDS) for 6LoWPAN Communication Protocol. *IEEE Access*, 3536(c).

NGU, A. H. H. *et al.,* 2016. IoT Middleware : A Survey on Issues and Enabling Technologies. X(X), pp. 1–20.

PAVAN PONGLE, G. C., 2015. A Survey : Attacks on RPL and 6LoWPAN in IoT. 0(c), pp. 0–5.

PHARATE, A., 2015. Classification of Intrusion Detection System. 118(7), pp. 23–26.

PONGLE, P. dan CHAVAN, G., 2015. Real Time Intrusion and Wormhole Attack Detection in Internet of Things. *International Journal of Computer Applications*, 121(9), pp. 975–8887.

RAZA, S., DUQUENNOY, S., 2011. Securing communication in 6LoWPAN with compressed IPsec. In Distributed Computing in Sensor Systems and. *IEEE Workshops (DCOSS)*, pp. 1–8.

RAZA, S., WALLGREN, L. dan VOIGT, T., 2013. SVELTE : Real-time intrusion detection in the Internet of Things. *AD HOC NETWORKS*. Elsevier B.V.

SAHU, S. dan MEHTRE, B. M., 2015. Network Intrusion Detection System Using J48 Decision Tree. *Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on*, pp. 2023–2026.

SHAH, H., SHRIMALI, R. dan PARIKH, V., 2016. Header Compression and Neighbor Discovery in 6LoWPAN based IoT - A survey. *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking*, pp. 306–311. *WiSPNET 2016.*

SHELBY, Z. dan BORMANN, C., 2009. *6LoWPAN : The Wireless Embedded Internet*.

SINGH, V. P., 2013. Signal Strength based Hello Flood Attack Detection and Prevention in Wireless Sensor Networks. 62(15), pp. 1–6.

SONAR, K. dan UPADHYAY, H., 2016. An Approach to Secure Internet of Things Against DDoS.

WALLGREN, L., RAZA, S. dan VOIGT, T., 2013. Routing Attacks and Countermeasures in the RPL-Based Internet of Things.