# Deteksi Serangan DoS Pada Jaringan SDN Berbasis P4 Programmable Dataplane menggunakan Machine Learning

**Tugas Akhir**

**diajukan untuk memenuhi salah satu syarat**

**memperoleh gelar sarjana**

**dari Program Studi S1 Informatika**

**Fakultas Informatika**

**Universitas Telkom**


**1301184219**

**Sya Raihan Heggi**



**Program Studi Sarjana S1 Informatika**

**Fakultas Informatika**

**Universitas Telkom**

**Bandung**

**2022**

# LEMBAR PENGESAHAN

**Deteksi Serangan DoS Pada Jaringan SDN Berbasis P4 Programmable Dataplane menggunakan Machine Learning**

***DoS Attack Detection on SDN with P4 Programmable Dataplane using Machine Learning***

**NIM: 1301184219**

**Sya Raihan Heggi**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi sebagian syarat memperoleh gelar pada Program Studi Sarjana S1 Informatika

Fakultas Informatika

Universitas Telkom

Bandung, 02 Juli 2022

Menyetujui

| Pembimbing 1 | Pembimbing 2 |
|---|---|
| Parman Sukarno, S.T., M.Sc., Ph.D. | Satria Akbar Mugitama, S.Kom., M.Kom. |
| NIP: 17770073-1 | NIP: 20910043 |

Ketua Program Studi

Sarjana S1 Informatika,

Dr. Erwin Budi Setiawan, S Si., M.T.

NIP: 00760045

# LEMBAR PERNYATAAN

Dengan ini saya, Sya Raihan Heggi, menyatakan sesungguhnya bahwa Tugas Akhir saya dengan judul "**Deteksi Serangan DoS Pada Jaringan SDN Berbasis P4 Programmable Dataplane menggunakan Machine Learning**" beserta dengan seluruh isinya adalah merupakan hasil karya sendiri, dan saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang belaku dalam masyarakat keilmuan. Saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam buku TA atau jika ada klaim dari pihak lain terhadap keaslian karya.

Bandung, 02 Juli 2022
Yang Menyatakan,

Sya Raihan Heggi

# Deteksi Serangan DoS Pada Jaringan SDN Berbasis P4 Programmable Dataplane menggunakan Machine Learning

**Sya Raihan Heggi[1], Parman Sukarno[2], Satria Akbar Mugitama**

[1,2,3]Fakultas Informatika, Universitas Telkom, Bandung
[1]heggiraihan@students.telkomuniversity.ac.id, [2]psukarno@telkomuniversity.ac.id,
[3]satriamugitama@telkomuniversity.ac.id

**Abstrak**

**Pada Tugas Akhir ini mengusulkan penggunaan LSTM-NB, LSTM-NB adalah kombinasi dari dua buat algoritma yaitu Long Short-Term Memory (LSTM) dan Naive Bayes (NB), algoritma ini akan digunakan untuk mendeteksi serangan Denial-of-Service (D0S) pada Programming Protocol-Independent Packet Processor (P4) Language-based Software Defined Network (SDN). Implementasi SDN saat ini semakin populer. Namun, didalam arsitektur SDN ini terdapat aspek yang kritis, salah satunya adalah rentan terhadap serangan DoS yang menyebabkan jaringan kehilangan prinsip CIA Triangle. Saat ini sudah beberapa penelitian yang melakukan pencegahan dari serangan. Namun, ancaman serangan DoS masih ada. Metode yang diusulkan menghasilkan akurasi 88% pada dataset SDN-DL, 98% pada NSL-KDD, dan 96% pada CICIDS2017 dengan nilai FNR 1-2%, selain pengujian menggunakan dataset dilakukan pengujian menggunakan data berdasarkan simulasi Iperf3 dan Hping3. Metode yang diusulkan akan di bandingkan dengan metode machine-learning dan deep-learning lainnya, berdasarkan evaluasi yang ekstensif, dapat disimpulkan metode yang diusulkan menunjukkan potensi kuat untuk melakukan deteksi serangan DoS di lingkungan SDN.**

**Kata kunci : Computer Network Security, Intrusion Detection System (IDS), Machine Learning, Deep Learning, Denial of Service (DoS)**

**Abstract**

**This thesis proposes LSTM-NB, a combination of Long Short-Term Memory (LSTM) and Naive Bayes (NB) algorithms to tackle Denial of Service (DoS) attacks on Programming Protocol-independent Packet Processors (P4) language-based Software Defined Network (SDN). The implementation of SDN is becoming more popular. However, there are critical aspects of the SDN architecture, one of which is that it is vulnerable to DoS attacks that can cause the network to lose the availability principle of the CIA Triangle. There are a number of works have been proposed to overcome this vulnerability, however, the threat is still exist. The proposed technique achieves an accuracy of 88% on SDN-DL Dataset, 98% on NSL-KDD, and 96% on CICIDS2017 with FNR score between 1-2%. In addition, we compare our proposed technique with other machine-learning and deep-learning methods. Through extensive experimental evaluation, we conclude that our proposed approach exhibits a strong potential for DoS detection in the SDN environments.**

**Keywords: Computer Network Security, Intrusion Detection System (IDS), Machine Learning, Deep Learning, Denial of Service (DoS)**

## 1. Pendahuluan

### 1.1 Latar Belakang

Jaringan komputer merupakan hal yang kompleks dan sulit untuk dikelola. Hal ini terjadi karena banyaknya perangkat yang digunakan pada jaringan [5]. Selain itu, perangkat pada jaringan tradisional memiliki desain, perangkat lunak, dan perangkat keras yang terkait dengan satu vendor. Setiap vendor memiliki desain dan perangkat yang berbeda [16]. Untuk mengatasi kompleksitas tersebut, diusulkan teknik Software Defined Network (SDN) [5]. SDN memiliki karakteristik yang berbeda jika dibandingkan dengan jaringan tradisional yaitu pemisahan control plane dan data plane dari perangkat jaringan [5, 16]. SDN menerapkan konsep sentralisasi pada arsitektur

jaringannya seperti jaringan tradisional. Namun, itu membuat jaringan arsitektur SDN rentan terhadap serangan cyber [9]. Ada tiga jenis serangan yang menargetkan jaringan SDN. Ini adalah serangan penipuan, serangan intrusi, dan serangan perusakan berbahaya [13]. Salah satu serangan terhadap jaringan SDN adalah Denial of Services; kerentanan ini disebabkan oleh arsitektur SDN [11].

Denial of Services atau biasa dikenal dengan DoS adalah kejahatan dunia maya dengan metode pengiriman paket secara berlebihan dan bertujuan untuk mengeksploitasi sumber daya jaringan [8]. Pemisahan bidang kontrol dan data menyebabkan serangan DoS di SDN [11]. Penyerang memanfaatkan control plane dan data plane [14], sehingga akan mengganggu flow rule decision dan juga dapat mengakibatkan terjadinya bottleneck pada jaringan. Ini bisa berbahaya jika ada kegagalan pada komponen jaringan [4]. Ada dua jenis serangan DoS, yaitu serangan volumetrik seperti ICMP-Flood, UDP-Flood, dan TCP-SYN Flood, dan serangan *application layer* [4, 15, 17], untuk mencegah serangan DoS dapat dibuatnya *Intrusion Detection System* (IDS) [18], IDS nantinya dapat berbasiskan *anomaly-based* dan *signature-based*. Namun kedua metode ini memiliki kekurangan jika diterapkan secara tradisional [19], terkait dengan kemampuan adaptasi dan skenario yang berbeda, oleh karena itu masalah ini dapat diselesaikan dengan menerapkan Machine-Learning . Pada tugas akhir ini diusulkan LSTM-NB, kombinasi dari algoritma Long Short-Term Memory (LSTM) dan Naive Bayes (NB), untuk memecahkan masalah ini.

**Tabel** 1. Akurasi beberapa metode

| Asal Paper | Akurasi dan Metode | Dataset Digunakan |
|---|---|---|
| Musumeci et al. [9] | SVM (97%), KNN (97%), **RandomForest(98%)** | Dataset Pribadi |
| Tang et al. [16] | **GRU-RNN (89%)**, RNN (44.39%), DNN (75.9%), NBtree (82.2%), dan SVM (69.52%) | NSL-KDD dan CICIDS2017 |
| Ahuja et al [2] | CNN (98%), LSTM(95%), **CNN-LSTM (99%)**, SVC-SOM (95%), dan SAE-MLP (99%) | SDN-DL |

Selain itu dilakukan pula evaluasi dengan membandingkan performansi dari sistem yang diusulkan dengan algoritma lainnya, diantaranya dapat dilihat pada tabel 1, untuk dataset yang digunakan untuk evaluasi akan menggunakan dataset SDN-DL[1], CICIDS2017, dan NSL-KDD[2]. selain evaluasi tersebut akan dilakukan evaluasi menggunakan data yang berasal dari simulasi menggunakan Hping3 dan Iperf3 pada simulasi P4-Mininet.

## 1.2 Topik dan Batasannya

Rumusan masalah yang menjadi dasar dari tugas akhir ini, karena saat ini IDS tradisional dan penggunaan IDS berbasiskan *Shallow Learning* masih memiliki keterbatasan, keterbatasan ini baik dari sisi akurasi dan kedinamisan model (tidak perlu *retrain* model berkali-kali). Sehingga permasalahan tersebut akan diselesaikan dengan menggunakan *Deep Learning*, yaitu dengan metode LSTM yang dikombinasikan dengan Naive Bayes (BA). Untuk pengerjaan tugas akhir ini akan menggunakan NSL-KDD, CICIDS 2017 [16] dan SDN-DL [10] ketiga dataset ini digunakan untuk melakukan perhitungan performansi, selanjutnya nanti akan diterapkan pada simulasi menggunakan Mininet dengan Dataplane P4.

## 1.3 Tujuan

Tujuan dari penelitian ini adalah untuk meningkatkan keamanan jaringan dari serangan DoS, hal ini dicapai dengan menggunakan Algoritma LSTM-NB, selain itu dibuat juga simulasi jaringan menggunakan P4-Mininet. Keterkaitan antara tujuan, pengujian, dan kesimpulan dapat dilihat pada Tabel 2

**Tabel** 2. Keterkaitan antara tujuan, pengujian dan kesimpulan

| No | Tujuan | Pengujian | Kesimpulan |
|---|---|---|---|
| 1 | Meningkatkan Keamanan Jaringan SDN dari Serangan DoS menggunakan Algoritma LSTM-NB | Algoritma LSTM-NB dapat melakukan deteksi serangan DoS, evaluasi yang dilakukan akan menggunakan dataset yang sudah disediakan yaitu NSL-KDD, CICIDS2017, dan SDN-DL | Algoritma LSTM-NB memiliki akurasi 98% pada Dataset NSL-KDD dengan nilai FNR 1.5%, 96% pada Dataset CICIDS2017 dengan nilai FNR 2.3%, dan pada dataset SDN-DL memiliki akurasi 89% dengan nilai FNR 13% |
| 2 | Melakukan Simulasi Menggunakan P4-Mininet | Algoritma LSTM-NB dapat melakukan deteksi pada jaringan SDN yang dibangun dengan menggunakan P4-Mininet | Sistem dapat melakukan deteksi ketika dilakukan simulasi dengan data yang dibuat menggunakan Iperf3 dan Hping3, untuk metode yang digunakan dapat menyiapkan data terlebih dahulu atau dapat membuat sistem \textit{realtime} untuk melakukan klasifikasi paket yang masuk |

[1] https://data.mendeley.com/datasets/jxpfjc64kr/1
[2] http://www.di.uniba.it/ andresini/datasets.html

### 1.4 Organisasi Tulisan

Pada jurnal tugas akhir ini akan memiliki organisasi tulisan sebagai berikut. Bab 1 akan membahas mengenai pendahuluan yang akan berisi latar belakang, topik dan batasan, tujuan dan organisasi tulisan dari jurnal tugas akhir. Bab 2 akan membahas mengenai dasar atau kajian teori yang mendukung pengerjaan dan penulisan tugas akhir. Bab 3 akan menjelaskan mengenai sistem yang akan dibangun, rencana evaluasi, dan metode evaluasi yang akan digunakan. Bab 4 akan berisi hasil evaluasi yang sudah dilakukan dan menjelaskan apa yang menjadi tujuan dari penelitian ini. Di akhir jurnal ini akan terdapat Bab 5 dimana didalamnya akan berisi kesimpulan dari kegiatan penelitian yang dilakukan.

## 2. Studi Terkait

### 2.1 Denial of Service Attack

Denial of Service adalah salah satu serangan *cybersecurity* yang menyerang jaringan komputer dan membuat jaringan komputer tidak dapat diakses [20]. Serangan ini menargetkan jalur-jalur komunikasi seperti infrastruktur atau komponen jaringan. Metode yang digunakan biasanya dengan membanjiri jaringan dengan paket dan membuat jaringan kekurangan sumber daya (overwhelm), hal ini membuat jaringan offline untuk beberapa waktu. Saat ini, terdapat beberapa jenis serangan DoS yang lazim terjadi, yaitu UDP Flood, ICMP Flood, TCP Flood, HTTP Flood, dan HTTPS Flood [1]. Serangan-serangan tersebut dapat dikelompokkan berdasarkan protokol seperti serangan UDP, TCP, HTTP, dan ICMP. Serangan dapat dengan mudah diklasifikasikan berdasarkan metode yang digunakan. Serangan DoS memiliki variasi yang luas dan metode yang digunakan. Namun, biasanya memiliki kesamaan yaitu dengan metode membanjiri jaringan menggunakan paket [8]. Menurut data Kaspersky Lab, sejak awal pandemi COVID-19, tingkat serangan DoS meningkat hingga 20% sejak pertumbuhan aktivitas online meningkat pesat. Serangan tersebut terutama terdiri dari tiga jenis serangan SYN, UDP, dan serangan TCP, dimana serangan SYN adalah 78,20%, diikuti oleh UDP 15,17%, dan diikuti oleh serangan TCP sebanyak 5,67%. Untuk meminimalkan dampak serangan DoS, dapat dilakukan pencegahan dengan membuat *early detection system*.

### 2.2 Software Defined Network

Software-Defined Network (SDN) adalah inovasi dalam jaringan yang mengubah cara mendesain dan mengelola jaringan, dengan menerapkan SDN memanajemen, mengontrol, dan menciptakan inovasi menjadi lebih mudah dan memungkinkan untuk diimplementasikan [5]. SDN memberikan paradigma baru dalam jaringan yang menerapkan sistem terpusat; hal ini dapat dicapai dengan memisahkan dua bagian dari jaringan. Control plane bertugas untuk mengontrol jaringan, dan data plane bertugas untuk meneruskan paket [5, 12, 3]. Selain itu, jaringan tradisional bersifat *closed*, dalam artian memiliki perbedaan antara vendor, sehingga administrator jaringan sulit untuk mengelola dan mengkonfigurasi [5, 16]. SDN dibangun di atas *layer* Aplikasi, Kontrol, dan Infrastruktur. Arsitektur ini memungkinkan SDN untuk mengimplementasikan arsitektur terpusat dan kemudian memiliki keuntungan untuk modifikasi jaringan. Selanjutnya, infrastruktur dan semua lapisan ini terhubung menggunakan *southbond* dan *northbond* untuk berkomunikasi satu dengan yang lainnya. Saat ini, protokol SDN yang terkenal adalah Openflow, yang menyediakan sistem SDN yang sederhana dan kuat. Namun, OpenFlow memiliki kelemahan yaitu kurangnya programabilitas. Untuk mengatasi masalah ini, maka digunakan solusi yaitu menggunakan P4 Programmable Dataplane.

### 2.3 P4 Language

P4 Language adalah bahasa pemrograman tingkat tinggi untuk router dan switch, yang dirancang untuk memungkinkan pemrograman pada komponen *data-plane* seperti perangkat keras atau perangkat lunak dari sebuah switch, *network interface cards*, router, dll [9]. P4 merupakan sebuah open-source berbeda dengan OpenFlow, P4 menggunakan desain top-down. Perbedaan utama antara P4 dan OpenFlow adalah *programmibility* dimana P4 memungkinkan untuk membuat program sesuai dengan kebutuhan dari pengguna. Dalam tugas akhir ini, simulasi menggunakan *behavioral-model* kemudian menggunakan simulator Mininet untuk mensimulasikan serangan jaringan. Berdasarkan Musumeci et al. program P4 terdiri dari komponen berikut:

- *Parsers* memiliki fungsi untuk mengidentifikasi protokol dan bidang yang diizinkan dalam program. Biasanya, mereka berisi nama header yang digunakan dan ukurannya dalam bit.

- *Control Plane (Ingress/Egress)* memiliki fungsi untuk menggambarkan urutan aturan pemrosesan yang diterapkan pada paket.
- *Table* memiliki banyak aturan pemrosesan yang membentuk ”match-action”. Ketika program P4 memproses paket, saluran masuk dijalankan untuk mencari aturan yang cocok yang sesuai dengan paket yang masuk.

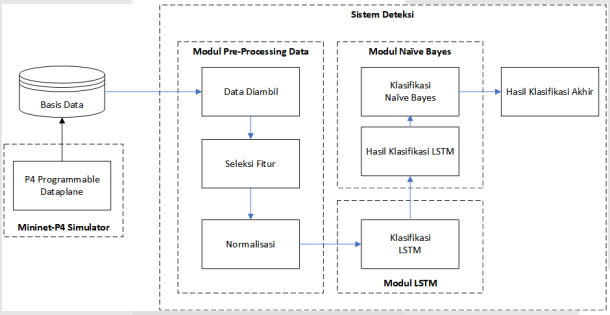
### 2.4 Dataset

Untuk melakukan evaluasi perlu menggunakan dataset, dataset yang digunakan untuk evaluasi dan populer diantaranya adalah NSL-KDD dan CICIDS 2017 [16]. Dataset NSL-KDD adalah salah satu dataset paling populer yang digunakan pada *NIDS Performance*. Dataset ini diperkenalkan oleh Tavallace et al., tetapi Dataset ini sudah lampau dan tidak memiliki keragaman lalu lintas dan fitur. CICIDS2017 digunakan untuk mengatasi masalah tersebut karena datasetnya relatif baru. Kedua dataset ini tidak dibuat untuk arsitektur SDN. Namun, dataset dapat digunakan karena kurangnya dataset publik untuk serangan DoS pada arsitektur SDN. Beberapa peneliti menghasilkan dataset SDN secara manual, tetapi kebanyakan bersifat tertutup dan oleh karena itu peneliti masih menggunakan dataset konvensional untuk mengevaluasi model yang dibuat [16]. Dalam tugas akhir ini, selain NSL-KDD dan CICIDS 2017, juga digunakan dataset terkait arsitektur SDN yang disediakan oleh Ahuja et al., untuk mengembangkan *anomaly-detection system* berbasiskan Deep-Learning pada jaringan SDN [2]. Dataset ini digunakan untuk membuktikan bahwa sistem yang diusulkan bekerja pada arsitektur SDN dan secara signifikan meningkatkan kinerja. Terakhir, dataset simulasi dibuat menggunakan simulasi SDN menggunakan Iperf3 dan Hping3 untuk menghasilkan lalu lintas jaringan dan kemudian ditangkap oleh aplikasi *network sniffing*.

## 3. Sistem LSTM-NB dan Kerangka Pengujian

### 3.1 Desain Sistem

Pada tugas akhir ini sistem yang dibangun dapat dilihat seperti pada Gambar 1, pada gambar akan terdapat **kotak dengan garis putus-putus** menandakan berada didalam satu kesatuan, **garis dengan panah** menandakan alur perpindahan data, **kotak** merupakan tahapan yang dilakukan, dan terakhir **simbol basis data** yang merupakan basis data tempat paket yang lewat disimpan.




**Gambar** 1. Desain Sistem Yang Dibangun

Dari Gambar 1 didapatkan informasi bahwa data akan didapatkan dari jaringan simulasi dan dikirimkan ke basis data, kemudian sistem deteksi yang didalamnya akan terdiri dari Modul Pre-Processing Data, Modul LSTM, dan Modul Naive Bayes yang berkerja untuk menghasilkan prediksi, sehingga bisa disimpulkan akan terdapat tiga komponen dalam sistem deteksi, yaitu: Data Processing, LSTM-Modul, dan Naive Bayes Modul.

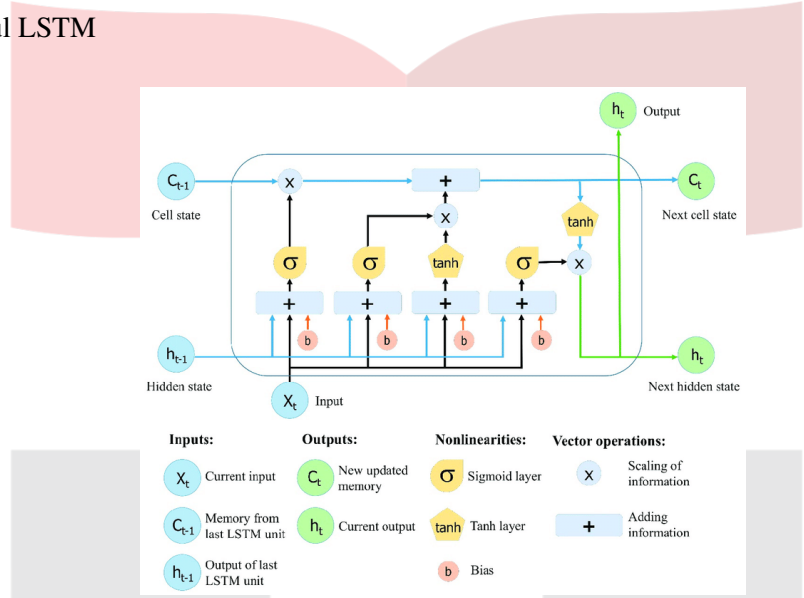
#### 3.1.1 Data Preprocessing

Pada NSL-KDD dan CICIDS 2017, kami melakukan *encoding* dan *scalling*, kemudian dilakukan transformasi menggunakan *one-hot-encoding* untuk fitur yang memerlukan transformasi dari objek ke numerik. pada dataset ini tidak dilakukan proses seleksi fitur saat dimasukan kedalam modul LSTM, tetapi fitur akan dipilih ketika menggunakan modul Naive Bayes dimana yang dipilih adalah hasil klasifikasi LSTM dan Protokolnya. Untuk dataset SDN-DL, dilakukan seleksi fitur menggunakan Heatmap, Chi-Square, Extra Tree Classifier, dan Irisan dari Ketiga

Metode. Selain itu, dilakukan penanganan *missing valu*e dan *duplicate value*. Terakhir, dilakukan normalisasi menggunakan metode min-max. Kemudian tidak lupa untuk melakukan LabelEncoding untuk beberapa fitur seperti protokol. Kemudian, pada dataset terakhir yang kami gunakan adalah dataset simulasi yang kami buat sebelum kami melakukan normalisasi data dan pemilihan fitur karena data ini digunakan untuk melatih model kami ke data simulasi.

Sehingga dari metode preprocessing yang sudah dilakukan dapat dibuat modul preprocessing ini akan melakukan *feature-selection*, *feature transformation*, dan *normalization*, sebelum nantinya datanya akan di lewatkan ke modul LSTM.

### 3.1.2 Modul LSTM




**Gambar** 2. Arsitektur LSTM [6]

Pada modul ini akan diletakan LSTM-classifier. LSTM merupakan penerapan dari Recurrent Neural Network yang memiliki kemampuan untuk mempelajari dependensi lama [11], di mana struktur sel LSTM itu sendiri digambarkan seperti yang ditunjukkan pada Gambar 2.Pada Gambar 2, setiap waktu $t$ sel LSTM ini dikendalikan oleh berbagai gerbang logika, yang bertujuan untuk mempertahankan atau mengatur ulang nilai dalam sel. Gambar 2 menunjukkan tiga jenis gerbang, gerbang terletak berurutan dari sisi kiri, gerbang terdiri dari *Forget Gate* ($f_t$), *Input Gate* ($i_t$), dan *Output Gate* ($o_t$) yang semuanya memiliki fungsi aktivasi sigmoid. Oleh karena itu, ada satu gerbang yang menggunakan fungsi tanh yang disebut *candidate value gate*.

$$c_t = f_t \odot c_{t-1} + i_t \odot c_t' \tag{1}$$

$$h_t = o_t \odot \tanh(c_{t-1}) \tag{2}$$

Proses pertama pada LSTM adalah *forget-gate* untuk memutuskan berapa banyak nilai yang dihapus dari keadaan sel, dan gerbang input tahap selanjutnya menentukan berapa banyak informasi baru yang disimpan, untuk langkah selanjutnya ($c_t$) dijelaskan berdasarkan persamaan dan untuk LSTM hasil ($h_t$) didefinisikan oleh persamaan 1 dan 2. Pada tugas akhir ini modul LSTM ini dibangun dengan desain layer LSTM (100 *perceptron*), kemudian dilanjutkan ke layer dropout (0.5), selanjutnya LSTM (32 *perceptron*), dan diakhiri dengan *fully-connected-layer* yang terdiri dari dua layer, yaitu layer ReLU (10 *perceptron*) dan layer output yang menggunakan Softmax (2 *perceptron*).

### 3.1.3 Modul Naive Bayes

Pada Modul ini akan berisi Naive Bayes Classifier. Naive Bayes adalah metode untuk mengklasifikasi berdasarkan teori Naive Bayes [7]. Metode ini merupakan klasifikasi dengan menggunakan pendekatan probabilistik

sederhana. Kita perlu menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari data.

$$P(H|E) = \frac{P(H|E)P(H)}{P(E))} \tag{3}$$

$$P(H|E) = P(E1|H) \times P(E2|H) \times \cdots \times P(En|H) \times P(H) \tag{4}$$

Nilai dapat diberikan ke atribut dan pola yang dihasilkan dari perhitungan probabilitas naive bayes, dari hasil perhitungan ini dapat digunakan untuk membuat klasifikasi. Naive Bayes sendiri diklasifikasikan sebagai *supervised learning* dalam *machine-learning* [7]. Naive Bayes sendiri dapat dijelaskan sebagai persamaan 3 dan persamaan 4. Pada Naive Bayes, perhitungan peluang suatu kejadian H yang merupakan kondisi pada data E dilakukan dengan terlebih dahulu menghitung peluang dari data E dengan kondisi H **(P(H|E))** , setelah itu, hasil **(P(H|E))** dikalikan dengan kondisi probabilitas H **(P(H))**, kemudian dibagi dengan data probabilitas E (P(E)). Oleh karena itu, pendeteksian serangan DoS dapat dilakukan dengan menggunakan perhitungan probabilitas serangan yang terjadi pada data kemudian dibagi dengan probabilitas data E [7].

## 3.2 Metode Evaluasi

Pada tugas akhir ini menggunakan parameter evaluasi seperti Accuracy, Precision, dan False Negative Rate, untuk menghitung kinerja dari sistem deteksi. Untuk menghasilkan itu, kita perlu membuat *Confusion Matrix*. Berdasarkan penelitian Qin et al. *Confusion Matrix* berisi item-item yang didefinisikan di bawah ini.

- **True Positive (TP)** adalah parameter dari Paket DoS yang diklasifikasikan sebagai Kondisi DoS.
- **True Negative (TN)** adalah parameter dari Paket Normal yang diklasifikasikan sebagai Kondisi Normal.
- **False Positive (FP)** adalah parameter Paket Normal yang diklasifikasikan sebagai Kondisi DoS.
- **False Negative (FN)** adalah parameter Paket DoS yang tergolong Kondisi Normal.

Berdasarkan parameter yang dijelaskan sebelumnya, akan dihitung parameter kinerja seperti Accuracy, Precision, dan False Negative Rate. Persamaan yang berasal dari penelitian yang dilakukan Li et al.dan Aljarwaneh et al. yang nantinya akan digunakan untuk mengkalkulasi performansi dari sistem yang dibangun.

- ***Accuracy*** didapatkan dengan membandingkan klasifikasi yang benar dengan jumlah total dalam dataset.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \tag{5}$$

- ***Precision*** merupakan bagian dari data yang tergolong positif dan bernilai benar positif.

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

- ***False Negative Rate*** adalah perbandingan yang menunjukkan jumlah paket yang salah yang diklasifikasikan sebagai benar.

$$FNR = \frac{FN}{TP+FP} \tag{7}$$

## 3.3 Metode Pengujian

Pada tugas akhir ini akan dilakukan dua jenis pengujian, pengujian pertama akan menggunakan dataset yang sudah *proper* dan dibuat dengan baik sehingga hasilnya bisa dipertanggungjawabkan, kemudian pengujian dataset ini akan dibandingkan dengan hasil pengujian yang sudah ada sebelumnya salah satunya dengan pengujian yang dilakukan oleh Ahuja et al. yang melakukan pengujian dengan menggunakan dataset SDN-DL, kemudian dibandingkan juga dengan penelitian yang dilakukan oleh Tang et al. yang menggunakan CICIDS2017 dan NSL-KDD, kemudian dilakukan pengujian dengan data yang didapatkan dari simulasi yang dilakukan menggunakan Iperf3 dan Hping3, kemudian hasil yang akan diamati adalah Akurasi, Presisi, dan FNR, yang menunjukan performansi dari sistem yang dibuat.

### 3.4 Kebutuhan Sistem Komputer

Dalam mengerjakan tugas akhir ini akan diimplementasikan pada laptop dengan menggunakan mesin virtual, kemudian perangkat tersebut memiliki spesifikasi yang dijelaskan pada Tabel 3.

**Tabel** 3. Spesifikasi Perangkat

| **Spesifikasi Laptop** | |
|---|---|
| **CPU** | Intel® Core™ i7-8565U (1.8 GHz base frequency, up to 4.6 GHz with Intel® Turbo Boost Technology, 8 MB cache, 4 cores) |
| **RAM** | 16 GB |
| **OS** | Windows 10 |
| **Spesifikasi Virtual Machine** | |
| **CPU** | 2 Cores Virtual Processor |
| **RAM** | 4 GB |
| **OS** | Ubuntu 18.04 |

Pada Tabel 3 terdapat spesifikasi *Virtual Machine*, *Virtual Machine* ini digunakan untuk melakukan simulasi jaringan SDN pada Mininet, dimana untuk melakukan hal ini dapat dilakukan dengan menggunakan sistem operasi Ubuntu 18.04.

## 4. Evaluasi

### 4.1 Hasil Pengujian

Pada bagian ini akan dipaparkan hasil dan analisis dari pengujian yang dilakukan, pengujian yang dilakukan terdiri dari pengujian dataset SDN-DL, pengujian dataset NSL-KDD dan CICIDS2017 ,dan pengujian menggunakan hasil simulasi.

#### 4.1.1 Hasil Dataset SDN-DL

Seperti yang sudah dijelaskan, pada pengujian SDN-DL ini memiliki tujuan untuk mencari fitur apa saja yang dapat digunakan untuk membuat sistem deteksi pada lingkungan SDN, hal ini didasari dari penelitian yang dilakukan Ahuja et al. pada penelitiannya tidak dilakukan pemilihan fitur oleh karena itu dilakukan pada pengujian tugas akhir ini, hasil yang didapatkan dituliskan seperti pada Tabel 4.

**Tabel** 4. Feature Selection Result

| **Algorithm** | **Accuracy (%)** | | **Precision (%)** | | **FNR (%)** | |
|---|---|---|---|---|---|---|
| | **Train** | **Test** | **Train** | **Test** | **Train** | **Test** |
| **ANN (Full Feature )** | **92.96** | **92.85** | **93** | **92** | **14.5** | **14.6** |
| ANN (Chi Square ) | 69.2 | 69.29 | 78 | 78 | 43.9 | 43.9 |
| ANN (Extra Tress) | 93.81 | 93.87 | 94 | 94 | 15.6 | 15.2 |
| ANN (Heatmap Slice ) | 84.99 | 85.10 | 84 | 84 | 10.9 | 10.8 |
| **LSTM (Full Feature )** | **94.24** | **94.09** | **94** | **94** | **14.2** | **14.2** |
| LSTM (Chi Square ) | 80.78 | 80.39 | 81 | 81 | 16.6 | 16.6 |
| LSTM (Extra Tress) | 90.79 | 90.78 | 90 | 90 | 13.4 | 13.4 |
| LSTM (Heatmap Slice ) | 88.57 | 87.86 | 88 | 87 | 12.6 | 12.5 |
| NB (Full Feature ) | 60.79 | 61.08 | 30 | 31 | - | - |
| NB (Chi Square ) | 60.79 | 61.08 | 30 | 31 | - | - |
| **NB (Extra Tress)** | **61.16** | **60.51** | **31** | **30** | - | - |
| NB (Heatmap Slice ) | 61.01 | 60.78 | 31 | 30 | - | - |

Dari Tabel 4, bisa disimpulkan bahwa pada Deep Learning ada baiknya untuk menggunakan semua fitur hal ini terlihat dari hasil pengujian LSTM dan ANN, kemudian untuk algoritma Shallow Learnign ada baiknya untuk dilakukan pemilihan fitur, dikarenakan algoritma yang diusulkan merupakan kombinasi Deep-Shallow Learning maka pemilihan fitur tetap menjadi pilihan. Kemudian pengujian berikutnya adalah melakukan perbaikan terhadap

performansi dari fitur yang ada, pada tahapan ini dipilih fitur yang memiliki performansi paling rendah yaitu **Heatmap Slice**, kemudian dilakukan pengujian menggunakan LSTM-NB, LSTM, Naive Bayes, ANN. dan CNN-LSTM kemudian hasil didapatkan dituliskan pada Tabel 5.

**Tabel** 5. Perbandingan Performansi Metode

| Algoritma | Akurasi | | | Presisi | | | FNR | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Training** | **Validasi** | **Testing** | **Training** | **Validasi** | **Testing** | **Training** | **Validasi** | **Testing** |
| LSTM + NB (Combined) | 88.77 | 88.18 | 88.02 | 88.18 | 82.41 | 10.78 | **88.02** | **84.16** | **14.53** |
| LSTM | 88.59 | 87.89 | 88.57 | 87.89 | 85.03 | 16.67 | 88.57 | 83.14 | 10.58 |
| Naıve Bayes | 62.01 | 61.46 | 61.85 | 61.46 | 51.20 | 62.26 | 61.85 | 51.30 | 64.47 |
| ANN | 87.698 | 87.67 | 87.69 | 87.67 | 83.51 | 14.09 | 87.69 | 83.36 | 14.09 |
| **CNN-LSTM [2]** | **99.48** | | | **99.55** | | | **3** | | |

Pada Tabel 5 metode CNN-LSTM hanya memiliki satu nilai, dikarenakan hasil ini merupakan hasil dari pengujian yang dilakukan oleh Ahuja et al. sementara pada pengujian metode yang dilakukan pada tugas akhir ini turut disertakan hasil *Training*, Validasi, dan *Testing*. hal ini ditujukan selain menunjukan hasil performansi dari sistem diusulkan tetapi menunjukan juga bahwa apakah metode yang digunakan terjadi Overfitting atau tidak, kemudian mengapa memilih algoritma yang ada pada Tabel 5 sebagai pengujian, LSTM dan Naive Bayes dipilih karena merupakan dasar dari metode yang diusulkan LSTM-NB, kemudian ANN untuk mewakili penggunaan Jaringan Syaraf Tiruan, dan CNN-LSTM merupakan inovasi yang diusulkan oleh peneliti sebelumnya.

### 4.1.2 Hasil Dataset NSL-KDD dan CICIDS2017

Pada pengujian kedua ini digunakan untuk melakukan komparasi performansi dari metode yang diusulkan dengan penelitian yang sudah ada terhadap dataset publik. kemudian alasan lainnya kenapa pengujian ini dilakukan adalah untuk membandingkan LSTM dengan basis RNN lainnya, dikarenakan seperti yang ditulis bahwa LSTM mampu memperbaiki kekurangan yang dimiliki oleh RNN, kemudian pada penggujian ini akan menggunakan dua buah dataset publik yaitu NSL-KDD dan CICIDS2017, NSL-KDD dipilih karena sudah umum digunakan dan cukup banyak fitur yang dapat ditelusuri namun memiliki kekurangan sudah lampau dan variasi serangan kurang. Oleh karena itu digunakan dataset lainnya yaitu CICIDS2017 yang masih tergolong baru dan dapat ditemukan dengan mudah, selanjutnya akan dilakukan perbandingan dengan algoritma yang sudah dilakukan pengujian oleh Tang et al. GRU-RNN merupakan algoritma berbasis RNN yang diusulkan oleh Tang et al. kemudian beberapa model lainnya yang memiliki basis yang berbeda diantaranya adalah DNN, VanillaRNN, SVM, dan NBTree, kemudian hasil yang didapatkan dituliskan pada Tabel 6, pada tabel tersebut ada beberapa kolom yang diisi dengan 0 dan - hal ini menandakan peneliti sebelumnya tidak menyertakan nilai dari parameter tersebut.

**Tabel** 6. Perbandingan Performansi Pada Dataset NSL-KDD and CICIDS2017

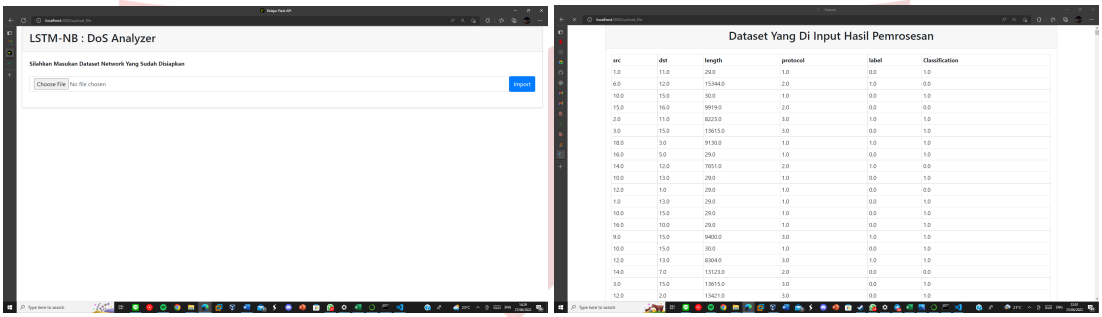
| Algorithm | NSL-KDD | | CICIDS2017 | |
|---|---|---|---|---|
| | **Accuracy** | **Precision** | **Accuracy** | **Precision** |
| LSTM + NB (Combined) | 98.85 | 99.03 | 96.4 | 97.8 |
| GRU-RNN [16] | 89 | 91 | 99 | 99 |
| DNN [16] | 75.9 | 0 | 75.75 | 0 |
| SVM [16] | 65.67 | 0 | 69.52 | 0 |
| NBTree [16] | **-** | **-** | 82.02 | 0 |
| VanillaRNN [16] | 44.39 | 0 | - | - |

### 4.1.3 Hasil Dataset Simulasi

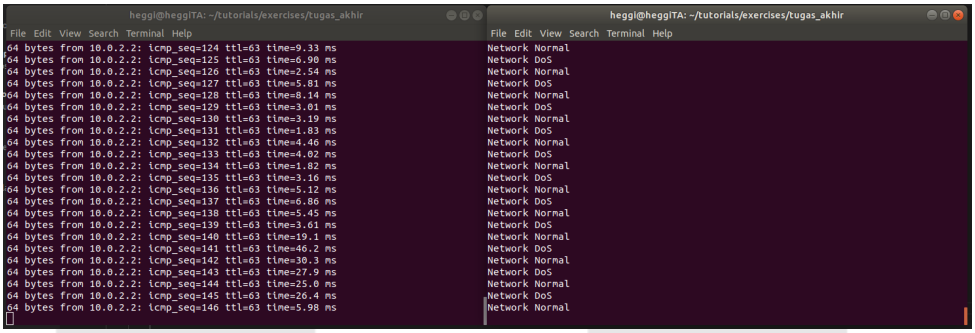
Pada pengujian ini dilakukan pengujian terhadap data yang didapatkan dari simulasi yang dilakukan pada P4-Mininet, hal ini bertujuan untuk membuktikan bahwa sistem dapat melakukan deteksi pada lingkungan simulasi P4-Mininet, hal pertama yang dilakukan adalah melakukan *sniffing* terhadap jalur jaringan, dan kemudian dilakukan sesi simulasi jaringan menggunakan Iperf3 dan Hping3, kemudian berdasarkan sesi yang dilakukan diberikan label terhadap data yang diberikan, kemudian dilakukan pengacakan data dari semua sesi, kemudian disatukan,

dan dipilih 4 fitur yaitu *'src'*. *'dst'*. *'length'*, dan *'protocol'*. Kemudian dari pengujian didapatkan hasil akurasi sebesar 100% dan FNR 0%. Namun, pengujian diperbaiki lagi karena menggunakan *tools* yang sederhana, dan simulasi jaringan yang bisa dibilang masih belum baik karena fokus dari pengujian ini hanya membuktikan bahwa sistem dapat di implementasikan.

Kemudian implementasi dari sistem sendiri dibuat dua buah variasi yaitu dengan menggunakan *network sniffer* dengan bantuan python scapy, dan juga sebuah website yang dibangun menggunakan *micro-framework* flask yang berfungsi untuk menerima data untuk di analisis menggunakan LSTM-NB dan mengeluarkan hasil klasifikasi terhadap data tersebut, untuk sistem yang dibuat dapat dilihat pada Gambar 3




(a) (b)




(c)

**Gambar** 3. (a) Contoh Input pada Website LSTM Analyzer (b) Output yang dihasilkan (c) Sistem Realtime menggunakan python scapy

## 4.2 Analisis Hasil Pengujian

Pada sub-bagian ini akan dipaparkan, analisis dari hasil pengujian yang sudah dilakukan, yang didalamnya akan memuat bagaimana pengujian dilakukan, bagaimana model itu dibangun, apa usaha yang dilakukan untuk melakukan optimalisasi performansi dari sistem, dll.

### 4.2.1 Analisis Hasil Evaluasi Dataset SDN-DL

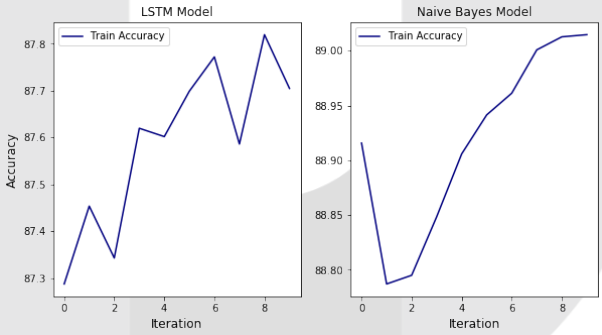
Pengujian pertama ini bertujuan untuk melakukan analisis secara komparatif terhadap sistem yang dibuat dengan penelitian yang sudah ada sebelumnya, menggunakan dataset yang sama namun memiliki metode yang berbeda, selain melakukan komparasi pada pengujian ini memiliki tujuan untuk menentukan fitur dan bagaimana fitur itu dapat digunakan, untuk mencapai hal ini maka dilakukan tahapan-tahapan. Tahap pertama adalah data preprocessing, pada tahapan ini dilakukan eksplorasi data, pada tahapan ini ditemukan bahwa dataset yang dimiliki terdiri dari data berjumlah 104345 data dengan 23 atribut, kemudian memiliki tipe int64, float64, dan object. Karena ada data yang memiliki maka perlu dilakukan tahapan *Encode*, pada tugas akhir ini digunakan metode LabelEncoding yang pada dasarnya melakukan penggantian nilai pada data bertipe object menjadi numerik, sebagai contoh adalah data protocol dimana akan terdiri dari 'TCP', 'UDP'. dan 'ICMP' jika dilakukan LabelEncoding maka data dapat dirubah menjadi TCP menjadi 1, UDP menjadi 2, dan ICMP menjadi 3, mengapa tahapan ini dilakukan hal ini

dilakukan untuk mempermudah ketika data digunakan untuk model pengujian lainnya karena ada beberapa metode yang hanya mampu untuk menerima masukan berupa data numerik, hal ini dapat terjadi karena pada dasarnya didalam sebuah metode yang dibuat terdapat perhitungan matematis dibaliknya,

Kemudian tahapan selanjutnya pembersihan terhadap dataset yang dimiliki, pada data ini ternyata memiliki *missing value* sebanyak 506 data pada atribut/fitur label, karena ini fitur yang seharusnya dilakukan oleh seorang ahli maka solusi yang dapat dilakukan hanya melakukan drop data, selain itu tidak ditemukan data duplikat pada dataset ini. Kemudian dilakukan *feature transformation* pada tahapan ini dilakukan dengan menggunakan metode Normalisasi dan Standarisasi. Normalisasi adalah sebuah metode untuk merubah skala menjadi seragam dengan menggunakan nilai maksimum dan minimum dari data, sementara untuk Standarisasi akan merubah skala dari data menggunakan nilai Mean dan Standar Deviasinya, tetapi dari pengujian yang dilakukan metode Standarisasi tidak dapat digunakan pada beberapa metode dengan dataset yang dimiliki, penyebab masalah ini adalah ada beberapa data yang menjadi memiliki rentang nilai dari minus, pada akhirnya Normalisasi yang digunakan, selanjutnya dikarenakan kita menggunakan tahapan normalisasi maka outlier akan tetap menjadi permasalahan yang harus diselesaikan.

Oleh karena itu dilakukan pengecekan terhadap outlier, pada dataset ini didapati outlier pada atribut pktrate, pktperflow, dst, dan pktcount, setelah dilakukan decision maka outlier ini tidak dihapus, alasan yang mendasari mengapa outlier ini tidak dihapus karena pada atribut-atribut yang terindikasi ada outlier memang ada kemungkinan data yang tidak dalam outlier, karena memang unik atau data tersebut merupakan bagian yang memiliki anomali, dan jika diamati memang atribut-atribut tersebut memang seharusnya memiliki data yang luas dan beragam. Selanjutnya setelah dataset dianggap bersih maka dilakukan pembagian data untuk dataset ini dilakukan pembagian 70% data untuk latih dan validasi dan 30% untuk data tes.

Tahapan berikutnya adalah melakukan *feature selection*, pada tahapan ini dilakukan dengan tiga metode yaitu Menggunakan ExtraTreesClassifier yang akan menggunakan algoritma ExtraTrees untuk melakukan perhitungan terhadap nilai fitur/atribut, kemudian menggunakan perhitungan Chi-Square, dan terakhir menggunakan Heatmap dan pertimbangan dari dua metode sebelumnya, karena jika fitur tersebut muncul di beberapa fitur seharusnya fitur itu seharusnya kritikal dan harus digunakan, maka akan dilakukan pengujian fitur yang didapatkan. Pada metode pertama digunakan perhitungan menggunakan Chi-Square dan didapati fitur *'dst','src','Protocol','pktcount','pktcount'*, selanjutnya menggunakan ExtraTreesClassifier didapatkan fitur textit'pktrate','pktperflow','Protocol', 'src','dst', dan dari metode terakhir *'dst','src','Protocol','bytecount'*, setelah fitur didapatkan maka akan dilakukan evaluasi menggunakan metode LSTM, VanillaNB, and ANN dan hasilnya terdapat pada Tabel 4.

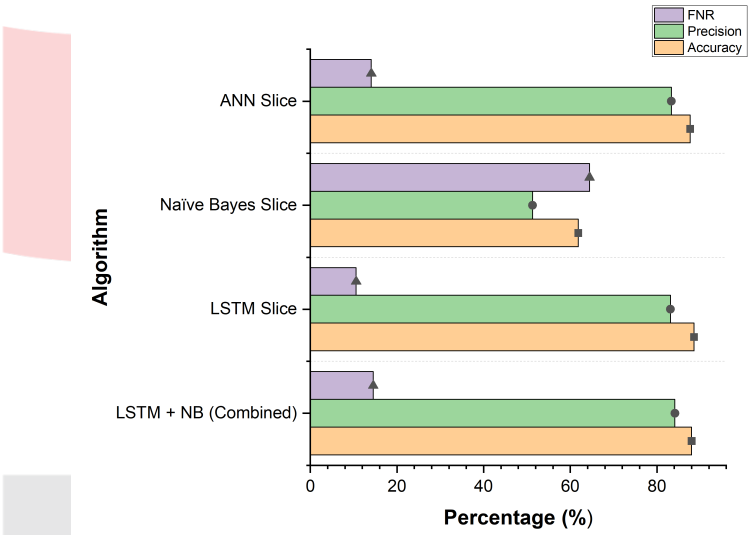



**Gambar** 4. Performansi LSTM dan Naive Bayes

Dari Tabel 4, bisa disimpulkan bahwa memang benar untuk penggunaan Deep Learning ada baiknya untuk menyertakan semua fitur/atribut, karena hasilnya menjadi lebih optimal hal ini ditunjukan dari hasil pengujian menggunakan LSTM dan ANN, namun untuk algoritma Shallow-Learning masih diperlukan seperti yang dilihat dari hasil yang didapatkan pada pengujian menggunakan Naive Bayes, selanjutnya jika kita melihat metode pemilihan fiturnya sendiri dapat disimpulkan bahwa hasil yang optimum dapat menggunakan ExtraTreesClassifier karena memiliki nilai performansi yang terbaik, dan yang terendah adalah menggunakan Heatmap Slice, sehingga Heatmap Slice ini yang akan digunakan untuk mengembangan klasifier pada tugas akhir ini, hal ini ditujukan untuk membuktikan bahwa sistem yang diajukan memang benar meningkatkan dari performansi sebelumnya.

Sebelum melakukan percobaan maka akan dilakuakan pembangunan model, untuk membangun model sendiri pada tahapan LSTM, digunakan 10 epoch untuk melatih dengan adam optimizer, serta metrik yang di awasi adalah MSE dan Akurasi, kemudian dari klasifikasi yang dilakukan modul LSTM ini nantinya akan mengeluarkan output

*LSTM_Result_1* dan *LSTM_Result_2*, setelah itu dilakukan pembangunan model Naive Bayes untuk membangun model ini digunakan Cross-Validation yaitu metode untuk melakukan validasi secara silang untuk memastikan nilai parameter optimal, pada metode ini digunakan 10 Fold, dan nilai yang dicari untuk Naive Bayes sendiri dengan model Gaussian adalah var_smoothing dan didapatkan nilai optimal adalah 0.732596542821523 dan menghasilkan akurasi sebesar 89%, untuk nilai parameter selama pengujian di dokumentasikan pada Gambar 4, selanjutnya model yang sudah dibangun ini akan dibandingkan dengan metode seperti LSTM, ANN, Naive-Bayes, dan CNN-LSTM yang hasilnya dapat dilihat pada Tabel 6.




**Gambar** 5. Komparasi Hasil Pengujian

Dari Tabel 6 didapati memang metode yang diajukan masih kurang baik dalam segi performansi akurasi, hal ini dapat terjadi karena memang dari penelitian yang dilakukan oleh Ahuja et al. memang menggunakan kombinasi dari dua buah algoritma Deep Learning, dimana sudah pasti hasilnya akan lebih baik dari Shallow Learning. Namun, seperti yang diketahui bahwa menggunakan algoritma Deep Learning maka memerlukan komputasi lebih, dan tidak diketahui mengenai fitur apa saja yang terlibat, mengapa hal ini penting karena seperti yang kita ketahui lingkungan SDN ini bisa dikatakan memiliki keterbatasan dalam melakukan ekstraksi sebuah fitur dari jaringan, oleh karena itu perlu diketahui fitur apa saja yang dapat digunakan, untuk pengembangan lanjutan. Walaupun hasilnya kurang memuaskan jika dibandingkan dengan penelitian sebelumnya tetapi jika dibandingkan dengan metode lainnya, metode yang digunakan sudah memiliki hasil yang baik seperti yang dituliskan pada 4, Untuk mempermudah melihat bahwa sistem yang dibuat memang mengoptimasi dan memiliki performansi yang baik dapat dilihat pada Gambar 5, dari gambar tersebut bisa dilihat bahwa sistem yang diusulkan memiliki akurasi dan presisi yang mirip seperti Deep Learning, walaupun FNRnya masih lebih besar dibandingkan dengan metode Deep Learning lainnya, namun lebih baik dibandingkan menggunakan metode Naive Bayes atau Shallow-Learning, kemudian hasil yang didapatkan metode yang diusulkan mampu meningkatkan kualitas dari metode lainnya yang karena tanpa metode yang digunakan Fitur Heatmap Slice akurasi sebesar 87.86% - 88.57% untuk LSTM dan 60.78% - 61.85% untuk Naive Bayes, dan FNR mencapai 10.58% - 12.5% untuk LSTM dan 64.47% untuk Naive Bayes , namun dengan LSTM-NB akurasi meningkat menjadi 88.02% - 88.77%, dengan nilai FNR 14.53%, angka ini didasarkan dari Tabel 6 dan Tabel 4.

### 4.2.2 Analisis Hasil Evaluasi Dataset NSL-KDD dan CICIDS2017

Pada Pengujian kedua ini memiliki tujaun untuk melakukan komparasi performansi sistem yang diusulkan terhadap penelitian yang dilakukan oleh Tang et al. selain itu pada dasarnya evaluasi yang dilakukan adalah membandingkan dengan metode RNN lainnya, sehingga dapat disimpulkan nantinya dari hasil analisis bahwa sistem yang diusulkan merupakan salah satu potensial. Sama seperti pengujian pada Dataset SDN-DL sebelum memulai yang dilakukan adalah melakukan pembersihan terhadap data, secara sederhana yang dilakukan adalah membersihkan data, kemudian melakukan One-Hot-Encoding metode ini sedikit berbeda dengan yang ada pada

dataset SDN-DL, dimana nantinya nilai fitur akan berubah menjadi 0 atau 1, dan Normalisasi. Selain itu dilakukan pembagian data untuk latih dan testing sebesar 70%: 30%, selanjutnya berdasarkan Tabel 5, pada tabel yang dibuat tidak dituliskan nilai FNR karena dari penelitian sebelumnya tidak ada pengujian FNR, namun pada pengujian metode yang diusulkan tetap dihitung nilai FRNnya, dapat disimpulkan bahwa metode yang diusulkan memiliki performansi yang baik, dikarenakan bisa dilihat pada Dataset NSL-KDD, metode yang diusulkan menjadi metode yang terbaik dengan akurasi 98%, presisi 99%, dan FNR 1%, sementara algoritma yang diusulkan oleh Tang et al, berada pada posisi kedua, dan menggunakan RNN saja memiliki hasil terburuk dengan akurasi mencapai 44%, sementara pada pengujian mengunakan dataset CICIDS2017 metode yang digunakan menjadi metode terbaik kedua setelah algoritma yang diusulkan Tang et al, dengan hasil performansi akurasi 96%, presisi 97%. dan FNR 2%.

Dari pengujian menggunakan Dataset SDN-DL dan CICIDS2017, dapat diketahui bahwa sistem yang diusulkan memiliki keunggulan. Pertama metode yang diusulkan dapat meningkatkan hasil dari metode vanilla, terlihat peningkatan terhadap penggunaan fitur hasil pemilihan Heatmap Slice, kemudian metode ini memiliki nilai performansi yang baik namun belum dapat mengalahkan penggunaan Deep Learning saja namun hasilnya lebih baik dari penggunaan Shallow Learning, kemudian jika dibandingkan dengan metode RNN, metode yang diusulkan memiliki performansi yang dimiliki lebih baik dibuktikan dari pengujian dengan dataset publik, kemudian juga lebih baik dari metode gabungan RNN, selanjutnya metode yang diajukan memiliki nilai FNR yang rendah dari pengujian yang ada, oleh karena itu metode yang diusulkan merupakan salah satu metode yang dapat menyelesaikan permasalahan ini, dan terakhir dengan menggunakan metode ini sudah pasti tidak memerlukan komputasi seperti menggunakan gabungan dua buah Deep Learning, namun untuk membuktikan komputasi ini perlu penelitian lebih lanjut.

### 4.2.3 Analisis Hasil Evaluasi Dataset Simulasi

Tujuan dari evaluasi 3 adalah untuk mengidentifikasi fitur mana yang dapat digunakan pada Jaringan SDN berbasis P4 karena P4 memiliki beberapa fitur berbeda yang dapat diekstraksi dengan OpenFlow SDN. Kami mengekstrak 'src'. 'dst'. 'length', dan 'protocol', Kemudian digunakan Akurasi dan FNR untuk mengevaluasi kinerja model kami dengan Simulasi P4-Mininet. Simulasi ini dibuat Hping3 dan Iperf. Kami melakukan simulasi jaringan selama 30 menit, dan setelah itu, data disimpan berdasarkan sesi yang kami lakukan. Pada sesi pertama, di simulasikan jaringan normal dengan Iperf3 dan parameter yang dimodifikasi adalah *Packet Windows* dan *Packet Length* dengan nilai antara 8 -2,4 Kbits, Selanjutnya pada sesi berikutnya adalah sesi serangan di simulasikan serangan UDP, SYN, TCP Flood, setelah itu data di proses dan diberikan label berdasarkan sesi. Kemudian dilakukan pengacakan terhadap dataset yang dibuat. Performa yang kami dapatkan adalah akurasi 100% dengan FNR 0%, yang bisa terjadi karena variasi serangan DoS sudah usang, selain itu untuk performansi ini sendiri diyakini masih memiliki banyak kekurangan, karena pada tugas akhir ini tidak difokuskan membangun dataset sehingga dataset yang dihasilkan dari simulasi kurang maksimum, sehingga perlu diperbaiki kedepannya. Namun, sistem ini terbukti dapat memberikan klasifikasi serangan DoS.

Selanjutnya untuk evaluasi ini sendiri dibuatkan sistem seperti pada Gambar 3, pada implementasi ini akan menggunakan dua buah metode, metode pertama adalah melakukan ekstraksi data terlebih dahulu dan kemudian dimasukan kedalam website untuk datanya dianalisis, kemudian pada metode kedua data didapatkan secara realtime dari simulasi, metode ini masih memiliki kekurangan dikarenakan fitur yang dapat diekstraksi tidaklah banyak, namun dapat di ekstraksi seperti pada hasil pengujian terhadap dataset simulasi.

## 5. Kesimpulan

Dari hasil, dapat disimpulkan bahwa metode yang diusulkan memberikan kinerja yang dapat diterima dan kompleksitas yang lebih sedikit dibandingkan dengan teknik Deep Learning. Metode yang diusulkan mencapai 88% akurasi, 85% presisi, dan 10% FNR. Menggunakan NSL-KDD dan CICIDS, skema yang kami usulkan mencapai akurasi 96%. Data yang diperoleh dari simulasi menggunakan P4-Mininet mencapai akurasi 100%. Ini membuktikan bahwa LSTM-NB yang kami usulkan dapat diimplementasikan dengan kinerja yang dapat diterima pada lingkungan SDN.

Saran untuk pengujian kedepannya, mungkin dapat dilakukan perbaikan terhadap arsitektur LSTM yang dibangun, selanjutnya sistem realtime yang dibangun mungkin dapat di kombinasikan dengan Controller SDN untuk dapat melakukan ekstraksi data lebih baik lagi, untuk saat ini Controller yang dapat digunakan untuk jaringan P4 adalah Onos, kemudian melakukan uji coba data traffic sebenarnya menggunakan metode yang lebih baru dan

lebih baik lagi karena pada penelitian ini hanya menggunakan Iperf3 dan Hping3, dan terakhir untuk variasi Naive Bayesnya mungkin dapat dicoba menggunakan distribusi lainnya dan dibandingkan apakah hasilnya akan menjadi lebih baik.

## Daftar Pustaka


[1] M. Aamir and M. Arif. Study and performance evaluation on recent ddos trends of attack & defense. *International Journal of Information Technology and Computer Science*, 5(8):54–65, 2013.

[2] N. Ahuja, G. Singal, and D. Mukhopadhyay. Dlsdn: Deep learning for ddos attack detection in software defined networking. In *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 683–688. IEEE, 2021.

[3] S. Azodolmolky. *Software defined networking with OpenFlow*, volume 153. Packt Publishing, 2013.

[4] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut. Ddosnet: A deep-learning model for detecting network attacks. In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 391–396. IEEE, 2020.

[5] N. Feamster, J. Rexford, and E. Zegura. The road to sdn: an intellectual history of programmable networks. *ACM SIGCOMM Computer Communication Review*, 44(2):87–98, 2014.

[6] X.-H. Le, H. V. Ho, G. Lee, and S. Jung. Application of long short-term memory (lstm) neural network for flood forecasting. *Water*, 11(7):1387, 2019.

[7] A. Mehmood, M. Mukherjee, S. H. Ahmed, H. Song, and K. M. Malik. Nbc-maids: Naïve bayesian classification technique in multi-agent system-enriched ids for securing iot against ddos attacks. *The Journal of Supercomputing*, 74(10):5156–5170, 2018.

[8] N. Moustafa, J. Hu, and J. Slay. A holistic review of network anomaly detection systems: A comprehensive survey. *Journal of Network and Computer Applications*, 128:33–55, 2019.

[9] F. Musumeci, V. Ionata, F. Paolucci, F. Cugini, and M. Tornatore. Machine-learning-assisted ddos attack detection with p4 language. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[10] D. M. Nisha Ahuja, Gaurav Singal. SDN-DDOS Dataset. `https://data.mendeley.com/datasets/jxpfjc64kr/1/`, 2020. [DOI : 10.17632/jxpfjc64kr.1].

[11] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença. Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment. *IEEE Access*, 8:83765–83781, 2020.

[12] H. Polat, O. Polat, and A. Cetin. Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, 12(3):1035, 2020.

[13] Y. Qin, J. Wei, and W. Yang. Deep learning based anomaly detection scheme in software-defined networking. In *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. IEEE, 2019.

[14] A. Sangodoyin, T. Sigwele, P. Pillai, Y. F. Hu, I. Awan, and J. Disso. Dos attack impact assessment on software defined networks. In *International Conference on Wireless and Satellite Systems*, pages 11–22. Springer, 2017.

[15] S. R. Talpur and T. Kechadi. A survey on ddos attacks: Router-based threats and defense mechanism in real-world data centers. In *2016 Future Technologies Conference (FTC)*, pages 978–984. IEEE, 2016.

[16] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho. Intrusion detection in sdn-based networks: Deep recurrent neural network approach. In *Deep Learning Applications for Cyber Security*, pages 175–195. Springer, 2019.

[17] O. Yevsieieva and S. M. Helalat. Analysis of the impact of the slow http dos and ddos attacks on the cloud environment. In *2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, pages 519–523. IEEE, 2017.

[18] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga. A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, 84:25–37, 2017.

[19] M. Zhang, J. Guo, B. Xu, and J. Gong. Detecting network intrusion using probabilistic neural network. In *2015 11th International Conference on Natural Computation (ICNC)*, pages 1151–1158. IEEE, 2015.

[20] V. Zlomislić, K. Fertalj, and V. Sruk. Denial of service attacks, defences and research challenges. *Cluster Computing*, 20(1):661–671, 2017.

## Lampiran

Lampiran dapat berupa detil data dan contoh lebih lengkapnya, data-data pendukung, detail hasil pengujian, analisis hasil pengujian, detail hasil survey, surat pernyataan dari tempat studi kasus, screenshot tampilan sistem, hasil kuesioner dan lain-lain.