Buletin Poltanesa Vol. 23 No. 2 (Desember 2022) p-ISSN 2721-5350 e-ISSN 2721-5369
https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda

Identifikasi Serangan *Low-Rate* DDOS Berbasis *Deep Learning*

Wahyuni*
Teknik Informatika, STMIK Widya Cipta Dharma,
Samarinda, 75123
wahyuni@wicida.ac.id
**Corresponding author*

Pitrasacha Adytia
Sistem Informasi, STMIK Widya Cipta Dharma,
Samarinda, 75123
pitra@wicida.ac.id

***Abstrak*—** LowRate DDoS (LDDoS) is a variation of DDoS attack that sends fewer packets than conventional DDoS attacks. However, by sending a smaller number of packets and using a unique attack period, low-rate DDoS is very effective in reducing the quality of an internet network-based service due to full access. On the other hand, the low-rate DDoS with its nature also makes it difficult to detect because it looks more mixed with normal user access. The Deep Learning model that will be used in this research is the RNN LSTM (Long Short Term Memory) model. LSTM is a neural network architecture which is good enough to process sequential data. This model is better than the simple RNN model. The research method is adapted to the SKKNI No. 299 of 2020. However, this research will be carried out until the model development stage, namely the evaluation model. From the results of the research that has been done, it can be concluded that the RNN LSTM model can be used to classify low-rate DDOS attacks using feature selection. The accuracy of the training data on the validation data is around 98% and after visualizing the data for accuracy and loss, it can be concluded that the model is quite good, aka there is no underfitting or overfitting. While the accuracy obtained for testing data is 0.97%.

***Kata Kunci*—**Low-Rate DDOS, Deep Learning, RNN LSTM, Machine Learning, DDOS.

I. PENDAHULUAN

Distributed Denial of Service (DDOS) merupakan penyebab utama yang dapat mengganggu *Software Defined Network*. DDOS merupakan jenis serangan Denial of Service yang menggunakan banyak host penyerang baik itu menggunakan komputer yang didedikasikan untuk penyerangan atau komputer yang dipaksa menjadi zombie untuk menyerang satu buah host target dalam sebuah jaringan (Junaedi & Fratelli, 2019). Ddos adalah sebuah metode serangan dengan mengirimkan banyak paket kedalam sebuah jaringan yang menyebabkan perangkat jaringan tidak berjalan sesuai fungsinya (Riadi, Umar, & Aini, 2019). Serangan Distributed Denial-of-Service (DDoS) dianggap sebagai ancaman keamanan utama bagi server online dan penyedia cloud. Ada beberapa jenis serangan ddos yang sering terjadi seperti UDP Flooding, SYN Flooding, Ping Of Death, dan Remote Controlled Attack (Darryl & Subali, 2021). Pada bulan Oktober hingga akhir Desember 2021 para peneliti Kaspersky mengamati peningkatan besar-besaran dalam bentuk jumlah serangan ddos. Serangan ddos mencapai rekor tertinggi di Q4 2021 dibandingkan dengan Q3 2021, jumlah total serangan ddos menunjukkan peningkatan 52%. Serangan ddos Q4 dilaporkan di beberapa negara seperti Amerika Serikat (43,55%), China (9,96%), Hong Kong (8,80%), Jerman (4,85%), dan Prancis (3,75%).

Serangan DDoS adalah bentuk serangan yang dilakukan dengan mengirim paket secara terus menerus kepada mesin bahkan jaringan komputer. Serangan ini akan mengakibatkan sumber daya mesin ataupun jaringan tidak bisa diakses atau digunakan oleh pengguna. Serangan DDoS merupakan varian dari serangan DOS, yang perbedaanya terletak pada dispersi sumber serangan (Gupta & Dahiya, 2021). Serangan DDOS adalah serangan yang berbahaya dan mengancam dalam suatu jaringan, karena dapat membanjiri jaringan dan memblokir akses ke server dengan mengirim paket dalam jumlah besar dan menggunakan sumber daya jaringan untuk menolak akses lainnya (Rahmatullah, 2022). Pada serangan DDoS jalur serangan digenerate dari beberapa sumber, sedangkan serangan DOS hanya bersumber dari satu tempat. Berbagai macam metode dilakukan dalam pendeteksian serangan DDOS pada SDN (Valdinos I. A., Perez-Diaz, Choo, & Botero, 2021), yaitu dengan cara statistikal, machine learning, arsitektur SDN, blockchain, Network Function Virtualization, honeynets, network slicing, dan moving target defense. Selain itu sudah banyak pula pada penelitian sebelumnya yang mengangkat topik mendeteksi serangan DDOS pada SDN menggunakan machine learning.

Saat ini, serangan DDOS tidak hanya dari high-level traffic, tetapi juga berupa lowlevel traffic. Serangan DDOS dengan low-level traffic disebut serangan low-rate DDOS. LowRate DDoS (LDDoS) merupakan salah satu variasi serangan DDoS yang mengirimkan jumlah paket lebih sedikit dibandingkan dengan serangan DDoS konvensional (Sudar & Deepalakshmi, 2022). Namun dengan mengirimkan jumlah paket yang lebih sedikit serta menggunakan periode serangan yang unik membuat low-rate DDoS sangat efektif untuk menurunkan kualitas suatu layanan berbasis jaringan internet akibat penuhnya

https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda

akses. Di sisi lain low-rate DDoS dengan sifatnya tersebut juga membuatnya menjadi sulit dideteksi karena terlihat lebih membaur dengan akses pengguna normal.

Berbagai macam metode dilakukan dalam pendeteksian serangan DDOS pada SDN (Valdinos et al, 2021), yaitu dengan cara statistikal, machine learning, arsitektur SDN, blockchain, Network Function Virtualization, honeynets, network slicing, dan moving target defense. Selain itu sudah banyak pula pada penelitian sebelumnya yang mengangkat topik mendeteksi serangan DDOS pada SDN menggunakan machine learning. Pada penelitian (Aljuhani, 2021)mengatakan bahwa ada beberapa hal menantang yang dapat di angkat dalam topik penelitian ini yaitu mendeteksi serangan LowRate DDOS. Pada penelitian (Perez-Diaz et al, 2020) juga mengatakan bahwa penelitian mengenai topik ini bisa dikembangkan dengan menggunakan metode terbaru dari machine learning dan menggunakan deep learning dengan tujuan untuk meningkatkan performansi dalam pendeteksian serangan. Penelitian (Sultana et al, 2018) mengatakan bahwa penggunaan deep learning sangat penting karena efisien dalam mengevaluasi keamanan jaringan.

Penelitian terdahulu yang telah dilakukan mengenai identifikasi low-rate DDOS, antara lain: Machine Learning Approaches for Combating Distributed Denial of Service Attacks in Modern Networking Environments (Aljuhani, 2021); A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDOS Attacks Using Machine Learning (Perez-Diaz & all, 2020); dan Flow-Base Detection and Mitigation of Low-Rate DDOS Attack in SDN Environment Using Machine Learning Techniques (Li & Li, 2018).

Pada penelitian (Aljuhani, 2021), dijelaskan mengenai pendekatan *Machine Learning* untuk melawan serangan DDOS. Dilakukan analisis mengenai pendekatan single dan *hybrid machine learning* dalam mendeteksi serangan DDOS. Penelitian tersebut juga mendiskusikan mengenai perbedaan sistem penangkal DDOS berbasis machine learning yang menggunakan *virtual environment*, termasuk *cloud computing*, *software defined network* dan *network function virtualization*.

Pada penelitian (Perez-Diaz & all, 2020), dibuatlah modul arsitektur yang flexibel di dalam SDN untuk mengidentifikasi dan memitigasi *Low-Rate* DDOS. Dilakukan uji coba *Intrusion Detection System* (IDS) di dalam arsitektur tersebut dengan menggunakan enam buah algoritma *machine learning*. Antara lain adalah, J48, *Random Tree, REP Tree, Random Forest, Multi Layer Perceptron (MLP), Support Vector Machines (SVM)*. Dan akurasi berhasil didapatkan sebesar 95% dengan menggunakan algoritma MLP.

Pada penelitian (Li & Li, 2018), dibuatlah suatu *flow-based* untuk mendeteksi dan memitigasi *low-rate* DDOS menggunakan teknik *machine learning*. Algoritma *machine learning* yang digunakan adalah *Support Vector Machine (*SVM), C4.5 *Decision Tree* dan *Naive Bayes.* Hasil dari penelitian tersebut adalah SVM memberikan hasil akurasi yang paling baik di antara C4.5 *Decision Tree* dan *Naive Bayes.*

Oleh karena itu, berdasarkan penelitian-penelitian tersebut, maka di ambilah judul penelitian "Identifikasi Serangan Low-Rate DDOS Berbasis Deep Learning." Adapun model Deep Learning yang akan digunakan dalam penelitian ini adalah model RNN LSTM (*Long Short Term Memory*). Dengan penelitian tersebut, diharapkan memberikan pengetahuan, insight yang baru pada peneliti yang berfokus pada bidang ini.

II. STUDI PUSTAKA

A. Low Rate DDOS

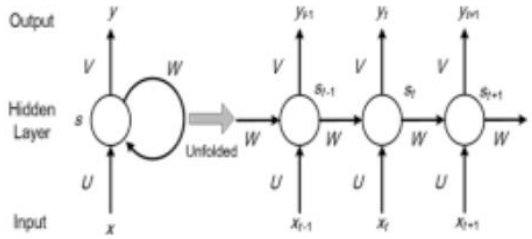
Low-Rate DDoS (LDDoS) merupakan salah satu variasi serangan DDoS yang mengirimkan jumlah paket lebih sedikit dibandingkan dengan serangan DDoS konvensional (Li & Li, 2018). Namun dengan mengirimkan jumlah paket yang lebih sedikit serta menggunakan periode serangan yang unik membuat LDDoS sangat efektif untuk menurunkan kualitas suatu layanan berbasisn jaringan internet akibat penuhnya akses. Low-rate DDOS telah menjadi salah satu ancaman terbesar bagi Internet, platform komputasi awan, dan pusat data besar (Zhijun et al, 2020). Sebagai spesies evolusi serangan DDoS, serangan LDDoS pada dasarnya berbeda dari serangan DDoS. Serangan DDoS adalah perilaku berbahaya memblokir lalu lintas jaringan yang sah dengan menghancurkan target dan infrastruktur di sekitarnya dengan lalu lintas jaringan yang sangat besar. Sementara, serangan LDoS adalah perilaku yang sengaja menurunkan kualitas tautan TCP dengan membatasi aliran TCP ke sebagian kecil dari laju idealnya dengan urutan pulsa kecil berkala. Di sisi lain LDDoS dengan sifatnya tersebut juga membuatnya menjadi sulit dideteksi karena terlihat lebih membaur dengan akses pengguna normal. Pembauran dengan akses pengguna normal ini terjadi karena intensitas suatu aliran akses (flow) serangan LDDoS mirip dengan intensitas dari akses pengguna biasa. Serangan *Low-rate* DDOS disebut juga dengan *shrew attack* (Li & Li, 2018). Serangan ini berbanding terbalik dengan serangan DDOS yang disebut *flood attack*. Jumlah serangan low-rate berkisar 10 – 20 % dari trafik normal dan sangat sulit ditemukan. Saat ini, serangan low-rate DDOS berfokus pada *high speed* dan layanan terpusat seperti *cloud*, dan *big data*.

B. Recurrent Neural Network (RNN)

RNN adalah jenis arsitektur jaringan saraf tiruan yang pemrosesannya berulang kali dipanggil untuk memproses input data sekuensial. RNN termasuk dalam kategori Deep Learning karena data diproses melalui banyak lapisan (Firmansyah & al, 2020). RNN adalah varian ANN yang yang memiliki fitur koneksi antara lapisan tersembunyi yang disebarkan melalui waktu untuk mempelajari urutan dan untuk melakukan pemrosesan data sekuensial (Nugraha & al, 2020). RNN merupakan jenis khusus dari deep learning yang diadaptasi untuk menyelesaikan data deret waktu atau data yang melibatkan urutan (Sarno et al, 2022). RNN disebut berulang karena melakukan tugas yang sama untuk setiap elemen ururtan, dengan keluaran yangb ergantung pada

https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda

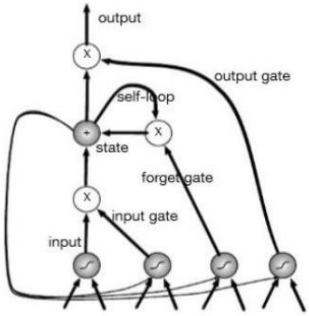
perhitungan sebelumnya (Nugroho, 2022). Pemodelan RNN dapat menyelesaikan berbagai tugas kategorisasi kalimat dan dapat melakukan klasifikasi, karena kemampuan dalam memprosesnya dipanggil berulang ulang dengan hasil dapat menangani input dan output variable yang panjangnya bervariasi. Pada intinya RNN adalah jaringan syaraf tiruan yang menggunakan rekurensi dengan memanfaatkan data masa lalu. Karena itu, beberapa studi terbaru mengenai RNN cukup kuat untuk permasalahan klasifikasi. Dapat dilihat pada gambar 1 merupakan gambaran dari metode RNN



Gambar 1. RNN

C. RNN Long Short Term Memory (LSTM)

LSTM adalah model yang semakin popular yang kekuatannya menangani kesenjangan ukuran yang tidak diketahui antara sinyal dalam kebis ingan data (Puspita et al, 2022). LSTM secara fundamental tidak memiliki perbedaan arsitektur dari RNN (Firmansyah & al, 2020), LSTM memiliki fungsi untuk mengkomputasi hidden state yang dapat merekam long-term dependencies (ketergantungan jangka pajang) selain itu LSTM adalah arsitektur jaringan saraf yang cukup baik untuk memproses data sekuensial. Model ini lebih baik dibandingkan dengan model RNN sederhana (Shandi & al, 2020). LSTM merupakan salah satu varian dari model recurrent neural network, yang memiliki isi cell lebih kompleks. Dari pengertian sebelumnya, dapat diketahui yang membedakan LSTM dengan RNN adalah gate function yang ada pada LSTM dan hal tersebut meningkatkan kemampuan LSTM secara signifikan dibandingkan dengan RNN. Dan alasannya mungkin fakta bahwa gate function yang digunakan dalam LSTM dapat memungkinkannya untuk menangkap ketergantungan jangka panjang lebih baik daripada RNN. LSTM paling cocok untuk data urutan (Raharjo, 2022). LSTM dapat memprediksi, mengklasifikasikan, dan menghasilkan data urutan. Urutan berarti urutan pengamatan, bukan serangkaian pengamatan. Contoh urutan adalah rangkaian pengujian di mana stempel waktu dan nilai berada dalam urutan (kronologis) dari urutan. Contoh lain adalah video, yang dapat dianggap sebagai rangkaian gambar atau rangkaian klip audio. Dapat dilihat pada gambar 2 yang merupakan dari gambaran metode LSTM



Gambar 2. LSTM

III. METODOLOGI

Diagram alur metode penelitian dapat dilihat pada gambar 3 berikut.

Gambar 3. Diagram Alur SKKNI No. 229 Tahun 2020

Metode penelitian disesuaikan dengan pendekatan Standar Kompetensi Kerja Nasional KepMen Ketenagakerjaan No.299 Tahun 2020 . Namun, penelitian ini akan dilakukan sampai tahapan pengembangan model yaitu *model evaluation*. Adapun penjelasan tahapan-tahapan metode penelitian adalah sebagai berikut:

1. *Business understanding*, pada tahapan ini akan dilakukan penentuan objek bisnis, tujuan teknis, dan membuat rencana proyek. Pada penelitian ini akan dilakukan penentuan masalah yang akan dipecahkan, yaitu bagaimana mengidentifikasi serangan low-rate DDOS.
2. *Data Understanding*, pada tahapan ini akan dilakukan pengumpulan data, penelaahan dan validasi data. Pada penelitian ini akan dilakukan pengumpulan data. Data yang akan digunakan adalah dataset CIC-Ddos-2017 CIC-Ddos-2018.
3. *Data Preparation*, pada tahapan ini akan dilakukan pemilahan data, pembersihan data,konstruksi data, menentukan label data, dan integrasi data. Pada proses pemilahan data akan dipilah record yang terpakai dan tidak, atribut yang terpakai dan tidak. Lalu dilakukan pembersihan data yaitu tahapan yang akan meminimalisir noise, yaitu memperbaiki data yang tidak lengkap, dan mendeteksi outlier. Selanjutnya akan dilakukan pemilihan feature dan melakukan transformasi data, dan setelahnya data akan digabungkan.

https://doi.org/10.51967/tanesa.v23i2.1737

4. *Modeling*, pada tahapan ini akan dibangun skenario pengujian dan membangun model. Dalam tahapan ini akan dilakukan pemilihan algoritma. Algoritma yang akan digunakan adalah RNN dan RNN LSTM. Dan dilakukan eksekusi algoritma, pengaturan parameter, dan pengkuran performa. Pada tahapan ini juga nantinya akan dilakukan skenario pemodelan.
5. *Model evaluation*, pada tahapan ini akan dilakukan evaluasi hasil model dan review proses pemodelan. Dalam penelitian ini akan dilakukan evaluasi model dengan cara mengukur performansi capaian terhadap target, dan memilih model terbaik. Setelahnya akan direview untuk mencari kelemahan dan kekurangan dalam pemodelan.

IV. HASIL DAN PEMBAHASAN

A. *Business Understanding*

Pada tahapan ini kita akan menentukan tujuan dari penelitian ini. Tujuan dari penelitian ini adalah dapat mengidentifikasi serangan Low Rate DDOS. Dan model yang digunakan adalah model RNN LSTM.

B. *Data Understanding*

Dataset yang digunakan adalah CICIDS2017 dan CICIDS2018. Terdapat 2.830.743 data dan 79 fitur pada CICIDS2017. Sedangkan pada CICIDS2018 terdapat 16.233.002 data dan 80 fitur. Untuk fitur pada CICIDS2017 dan CICIDS2018 sebenarnya hanya memiliki sedikit perbedaan pada penamaan fitur.

Setelah dilihat banyak data dan banyak fitur pada dataset tersebut, maka dilanjutkan dengan melihat jenis dan jumlah serangan pada masing-masing dataset. Jenis serangan terdapat pada fitur Label.

```
[ ]  CICIDS2017['Label'].value_counts()

BENIGN                        2073361
DoS Hulk                       172849
DDoS                           128016
PortScan                        90819
DoS GoldenEye                   10286
FTP-Patator                      5933
DoS slowloris                    5385
DoS Slowhttptest                 5228
SSH-Patator                      3219
Bot                              1953
Web Attack – Brute Force         1470
Web Attack – XSS                  652
Infiltration                       36
Web Attack – Sql Injection         21
Heartbleed                         11
Name: Label, dtype: int64
```

Gambar 4. Jenis dan Jumlah Serangan Dataset CICIDS2017.

Pada gambar 4 di atas dapat dilihat bahwa terdapat 14 buah jenis serangan DDOS dan Low Rate DDOS. Sedangkan Benign adalah bukan serangan. Dari gambar di atas BENIGN memiliki jumlah data yang cukup banyak, disusul dengan DoS HULK.

```
CICIDS2018['Label'].value_counts()

Benign                      10636903
DDoS attacks-LOIC-HTTP        575364
DDOS attack-HOIC              198861
DoS attacks-Hulk              145199
Bot                           144535
Infilteration                 140610
SSH-Bruteforce                 94048
DoS attacks-GoldenEye          41406
DoS attacks-Slowloris           9908
DDOS attack-LOIC-UDP            1730
Brute Force -Web                 555
Brute Force -XSS                 228
SQL Injection                     84
DoS attacks-SlowHTTPTest          55
FTP-BruteForce                    54
Label                              1
Name: Label, dtype: int64
```

Gambar 5. Jenis dan Junlah Serangan Dataset CICIDS2018

Pada gambar 5bdi atas dapat dilihat bahwa terdapat 16 buah jenis serangan. Dan dapat dilihat pula bahwa jumlah Benign dengan serangan sangat berbanding jauh.

Dari gambar 5 dan 6 dapat disimpulkan bahwa jenis serangan yang ada pada dataset tersebut tidak berbeda jauh. Hanya berbeda pada penulisan jenis serangan dan jumlah serangan pada CICIDS2018 sedikit lebih banyak.

C. *Data Preparation*

Setelah melakukan tahapan data understanding, maka dilakukanlah tahapan data preparation. Pada CICIDS2017 terdapat 2 buah fitur yang sama, maka akan coba drop salah satunya. Bisa dilihat pada gambar 6 berikut ini.

```
CICIDS2017=CICIDS2017.drop(' Fwd Header Length.1',axis=1)
CICIDS2017.info()
```

Gambar 6. Drop Fitur Fwd Header Length.1

Setelah mendrop fitur yang duplikat, maka akan diperbaiki penulisan fitur pada dataset CICIDS2017. Dari data understanding, terlihat bahwa penulisan fitur tidak seragam. Beberapa fitur menggunakan spasi di depan. Maka kita akan menghilangkan spasi yang berada di depan nama fitur tersebut. Dapat dilihat pada gambar 7 berkut ini.

```
[ ]  CICIDS2017.columns = [x.strip() for x in CICIDS2017.columns]
```

Gambar 7. Menghapus spasi di depan nama fitur.

CICIDS2017 dan CICIDS2018 akan digabungkan. Fitur pada CICIDS2018 akan mengikuti fitur yang terdapat pada CICIDS2017. Maka dilakukan penghapusan fitur ‘Src IP’, ‘Src Port’, ‘Dst IP’, ‘Protocol’ dan ‘Timestamp’ pada CICIDS2018. Setelah melakukan penghapusan fitur, maka dilakukan pengubahan type data CICIDS2018. Sebelumnya fitur-fitur CICIDS2018 berupa object. Maka akan diubah menjadi numerik.

https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda


```
CICIDS2018=CICIDS2018.drop(['Protocol','Times
CICIDS2018.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 16233002 entries, 0 to 1048574
Data columns (total 78 columns):
 #   Column             Dtype
---  ------             -----
 0   Dst Port           object
 1   Flow Duration      object
 2   Tot Fwd Pkts       object
 3   Tot Bwd Pkts       object
 4   TotLen Fwd Pkts    object
 5   TotLen Bwd Pkts    object
 6   Fwd Pkt Len Max    object
 7   Fwd Pkt Len Min    object
 8   Fwd Pkt Len Mean   object
 9   Fwd Pkt Len Std    object
 10  Bwd Pkt Len Max    object
 11  Bwd Pkt Len Min    object
 12  Bwd Pkt Len Mean   object
 13  Bwd Pkt Len Std    object
 14  Flow Byts/s        object
 15  Flow Pkts/s        object
 16  Flow IAT Mean      object
 17  Flow IAT Std       object
 18  Flow IAT Max       object
 19  Flow IAT Min       object
 20  Fwd IAT Tot        object
 21  Fwd IAT Mean       object
 22  Fwd IAT Std        object
 23  Fwd IAT Max        object
 24  Fwd IAT Min        object
 25  Bwd IAT Tot        object
 26  Bwd IAT Mean       object
```

Gambar 8. Tipe data fitur CICIDS2018 sebelum dirubah.

```
[ ]  CICIDS2018.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 16233002 entries, 0 to 1048574
Data columns (total 78 columns):
 #   Column             Dtype
---  ------             -----
 0   Dst Port           float64
 1   Flow Duration      float64
 2   Tot Fwd Pkts       float64
 3   Tot Bwd Pkts       float64
 4   TotLen Fwd Pkts    float64
 5   TotLen Bwd Pkts    float64
 6   Fwd Pkt Len Max    float64
 7   Fwd Pkt Len Min    float64
 8   Fwd Pkt Len Mean   float64
 9   Fwd Pkt Len Std    float64
 10  Bwd Pkt Len Max    float64
 11  Bwd Pkt Len Min    float64
 12  Bwd Pkt Len Mean   float64
 13  Bwd Pkt Len Std    float64
 14  Flow Byts/s        float64
 15  Flow Pkts/s        float64
 16  Flow IAT Mean      float64
 17  Flow IAT Std       float64
 18  Flow IAT Max       float64
 19  Flow IAT Min       float64
 20  Fwd IAT Tot        float64
 21  Fwd IAT Mean       float64
 22  Fwd IAT Std        float64
 23  Fwd IAT Max        float64
```

Gambar 9. Tipe data CICIDS2018 setelah dirubah

Gambar 8 merupakan gambar fitur yang masih bertipe object yang artinya belum dirubah. Sedangkan gambar 9 merupakan gambar fitur yang tipe datanya sudah dirubah menjadi float. Setelah tipe data sudah dirubah, maka dilakukan pembersihan data dengan menghapus data duplikat pada CICIDS2017 maupun CICIDS2018. Sebelum dihapus, data pada CICIDS2017 sebanyak 2.830.743. Terdapat 331.504 data duplikat pada dataset tersebut. Setelah data duplikat dihapus, maka banyak data pada CICIDS2017 berkurang menjadi 2.499.239

Begitupun dengan CICIDS2018. Banyak data sebelum data duplikat dihapus adalah sebanyak 16.233.002. Jumlah data duplikat pada CICIDS2018 sebanyak 4.243.461 data. Sehingga setelah dibersihkan data CICIDS2018 menjadi 11.989.541 data.

Pada tahapan data understanding, dapat dilihat bahwa terdapat banyak jenis serangan. Maka akan dilakukan pemilahan jenis serangan yang termasuk Low-rate DDOS. Adapun serangan yang termasuk Low-Rate DDOS adalah DDOS LOIC HTTP, DDOS HOIC, Dos HULK, Dos GoldenEye, Dos Slowloris, Dos SlowHTTPTest (Swe et al, 2021). Maka pada data set CICIDS2017 dan CICIDS2018 akan diambil data yang memiliki serangan seperti yang disebutkan di atas. Setelah difilter berdasarkan jenis serangan low-rate DDOS, maka data CICIDS2017 digabungkan dengan CICIDS2018.

```
CICIDS['Label'].value_counts()

Benign             12710264
DDOS LOIC HTTP       575364
DoS HULK             318048
DDOS HOIC            198861
DoS GoldenEye         51692
DoS Slowloris         15293
DoS SlowHTTPTest       5283
Name: Label, dtype: int64
```

Gambar 10. Jenis serangan pada data CICIDS yang telah digabungkan.

Dapat dilihat pada gambar 10 bahwa selisih antara “benign” dengan serangan low rate DDOS sangat tidak seimbang. Benign dan serangan low-rate DDOS akan dibuat menjadi biner 0 dan 1. 0 menyatakan benign, dan 1 menyatakan serangan low-rate ddos. Maka selain dari pada benign, akan di ubah labelnya menjadi 1.

```
[ ]  clean_df['Attack'] = np.where(clean_df['Label'] == 'Benign', 0, 1)

[ ]  clean_df['Attack'].value_counts()

0    12636706
1     1164538
Name: Attack, dtype: int64
```

Gambar 11. Merubah label menjadi 0 dan 1

Pada gambar 11 di atas, sudah dirubah jenis serangan menjadi 0 dan 1. Setelah dilihat jumlah datanya, dapat dilihat bahwa terjadi imbalance data, yang jumlah data yang berlabel 0 sekitar 10 kali lipat lebih banyak daripada data yang berlabel 1. Setelah itu dilakukan seleksi fitur secara manual, yang akan dicari dihapus fitur yang datanya semua sama. Dapat dilihat pada gambar 12.

https://doi.org/10.51967/tanesa.v23i2.1737

```
column_names = np.array(list(clean_df))
to_drop = []
for x in column_names:
    size = clean_df.groupby([x]).size()
    # check for columns that only take one value
    if (len(size.unique()) == 1):
        to_drop.append(x)
to_drop
```

```
['Bwd PSH Flags',
 'Bwd URG Flags',
 'Fwd Avg Bytes/Bulk',
 'Fwd Avg Packets/Bulk',
 'Fwd Avg Bulk Rate',
 'Bwd Avg Bytes/Bulk',
 'Bwd Avg Packets/Bulk',
 'Bwd Avg Bulk Rate']
```

Gambar 12. Fitur yang isi datanya sama

Setelah menghapus beberapa fitur yang memiliki data yang sama, maka dilakukan pemisahan data, yaitu memisahkan fitur independent dengan label (goal). Setelahnya dilakukan balancing data sehingga data yang berlabel 0 menjadi hanya dua kali lipat berbanding dengan data yang berlabel 1. Dapat dilihat pada gambar 13.

```
[ ] xs = clean_df.drop(['Label','Attack'], axis=1)
    ys = clean_df['Attack']
```

```
from collections import Counter
from sklearn.datasets import make_classification
from imblearn.under_sampling import RandomUnderSampler

# define undersample strategy
undersample = RandomUnderSampler(sampling_strategy=0.5)
# fit and apply the transform
X_over, y_over = undersample.fit_resample(xs, ys)
# summarize class distribution
print(Counter(y_over))
```

```
Counter({0: 2329076, 1: 1164538})
```

Gambar 13. Proses pemisahan dan balancing data.

Pada gambar 13 dapat dilihat bahwa jumlah data yang berlabel 0 sebanyak 2.329.076 data dan yang berlabel 1 sebanyak 1.164.538 data. Maka dapat dikatakan bahwa data sudah balance. Setelah itu dilakukan normalisasi menggunakan teknik normalisasi min max. Normalisasi dilakukan agar semua data memiliki rentang range yang sama yaitu antara 0 dan 1. Dapat dilihat pada gambar 14.

```
# Normalise
min_max_scaler = MinMaxScaler().fit(X_over)

# Apply normalisation to dataset
X_over = min_max_scaler.transform(X_over)

# All values between 0 and 1
pd.Series(X_over.flatten()).describe()
```

```
count    2.410594e+08
mean     1.671949e-01
std      3.607672e-01
min      0.000000e+00
25%      0.000000e+00
50%      5.937890e-06
75%      1.200641e-02
max      1.000000e+00
dtype: float64
```

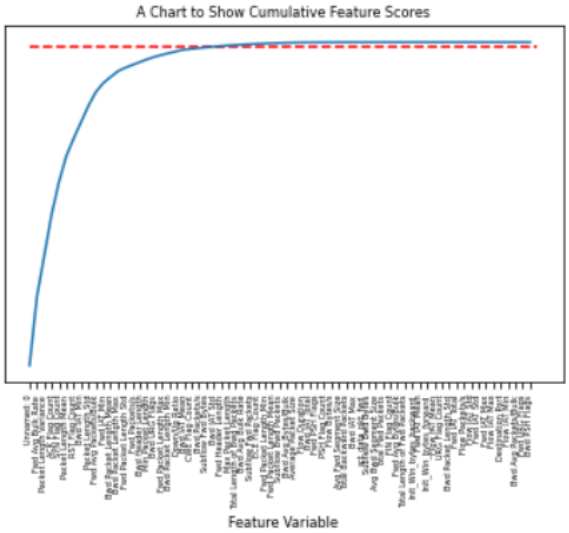
Gambar 14. Proses normalisasi

Pada gambar 14 dapat dilihat bahwa data pada semua fitur sudah ternormalisasi yaitu berada pada rentang 0 dan 1. Lalu dilakukan pemisahan data training dan data testing. Dapat dilihat pada gambar 15.

```
[11] # split dataset - stratified
     x_train, x_test, y_train, y_test = train_test_split(X_over, y_over, test_size=0.3, random_state=0)
```

Gambar 15. Pemisahan data training dan data testing

Data dipisah menjadi 70% data training dan 30% data testing. Lalu dilakukan seleksi fitur menggunakan feature importance.



Gambar 16. Grafik feature importance.

Dari gambar 16 dapat dilihat bahwa terdapat sekitar 30 feature yang baik untuk digunakan dalam kasus ini. Maka setelah mendapatkan jumlah fitur yang diinginkan, dilakukan lagi seleksi fitur menggunakan chi square dengan memasukkan jumlah fitur yang akan digunakan.

```
features = SelectKBest(score_func=chi2, k=30)

#fit features to the training dataset
fit = features.fit(x_train, y_train)
```

```
[ ] x_train = fit.transform(x_train)
    x_test = fit.transform(x_test)
    x_validate = fit.transform(x_validate)
```

```
[ ] x_train.shape
```

```
(2096168, 30)
```

```
[ ] new_features = clean_df.columns[features.get_support(indices=True)]
    new_features
```

```
Index(['Unnamed: 0', 'Total Length of Bwd Packets', 'Fwd Packet Length Max',
       'Fwd Packet Length Std', 'Bwd Packet Length Max',
       'Bwd Packet Length Min', 'Bwd Packet Length Mean', 'Fwd IAT Min',
       'Bwd IAT Mean', 'Bwd IAT Std', 'Bwd IAT Min', 'Bwd URG Flags',
       'Fwd Header Length', 'Bwd Header Length', 'Fwd Packets/s',
       'Bwd Packets/s', 'Min Packet Length', 'Max Packet Length',
       'Packet Length Mean', 'Packet Length Std', 'Packet Length Variance',
       'SYN Flag Count', 'RST Flag Count', 'ACK Flag Count', 'CWE Flag Count',
       'Down/Up Ratio', 'Fwd Avg Packets/Bulk', 'Fwd Avg Bulk Rate',
       'Bwd Avg Bulk Rate', 'Subflow Fwd Bytes'],
      dtype='object')
```

Gambar 17. Proses seleksi fitur dengan chi square.

Dari gambar 17 dapat dilihat bahwa fitur telah terseleksi sebanyak 30 fitur.

https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda

D. Modelling

Dalam model LSTM dilakukan reshape atau perubahan bentuk data menjadi array tiga dimensi.

```
[ ]  x_train_1 = np.reshape(x_train, (x_train.shape[0],x_train.shape[1],1))
     x_test_1 = np.reshape(x_test, (x_test.shape[0],x_test.shape[1])),1
```

Gambar 18. Proses reshape data training dan testing.

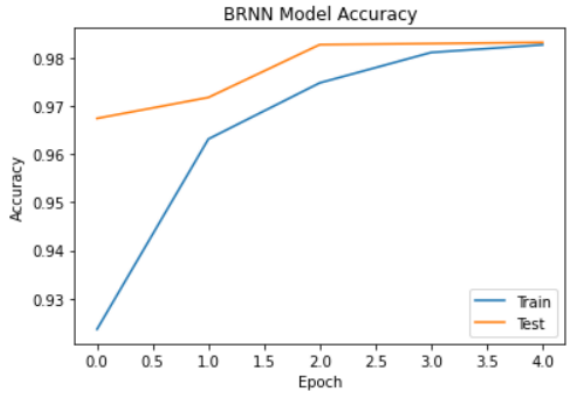
Setelah dilakukan reshape data, maka dibuatlah model LSTM dengan inputan sebanyak 30 dan diulang untuk training sebanyak 5 kali.

```
regressor = Sequential()
regressor.add(Bidirectional(LSTM(units=30, return_sequences=True, input_shape = (x_train.shape[1],1) ) )) #çıktı uzayının boyutu
regressor.add(Dropout(0.2))
#regressor.add(LSTM(units= 10 , return_sequences=True))
#regressor.add(Dropout(0.2))
#regressor.add(LSTM(units= 10 , return_sequences=True))
#regressor.add(Dropout(0.2))
#regressor.add(LSTM(units= 10))
#regressor.add(Dropout(0.2))
regressor.add(Dense(units = 1,activation='sigmoid'))
regressor.compile(optimizer='adam', loss='binary_crossentropy',metrics=['acc'])
history=regressor.fit(x_train_1, y_train, epochs=5,batch_size=64, validation_split=0.2,verbose=1)
```

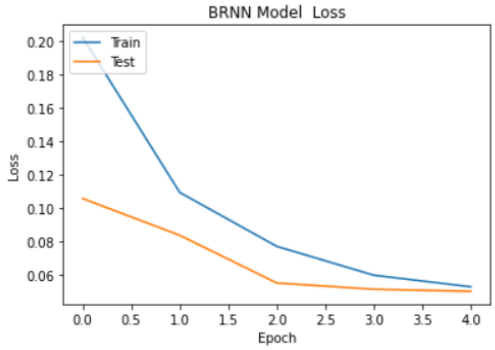
```
Epoch 1/5
30570/30570 [==============================] - 768s 25ms/step - loss: 0.2023 - acc: 0.9237 - val_loss: 0.1055 - val_acc: 0.9674
Epoch 2/5
30570/30570 [==============================] - 654s 21ms/step - loss: 0.1092 - acc: 0.9631 - val_loss: 0.0834 - val_acc: 0.9717
Epoch 3/5
30570/30570 [==============================] - 602s 20ms/step - loss: 0.0768 - acc: 0.9748 - val_loss: 0.0548 - val_acc: 0.9827
Epoch 4/5
30570/30570 [==============================] - 592s 19ms/step - loss: 0.0595 - acc: 0.9811 - val_loss: 0.0510 - val_acc: 0.9829
Epoch 5/5
30570/30570 [==============================] - 585s 19ms/step - loss: 0.0525 - acc: 0.9827 - val_loss: 0.0498 - val_acc: 0.9831
```

Gambar 18. Pemodelan LSTM

Pada model di atas digunakan inputan terhadap model sebanyak 30 dan dilakukan training terhadap model sebanyak 5 kali perulangan. Evaluasi model di atas menggunakan data training dan data validasi, yang jumlah data validasi sebanyak 20% dari data training. Pada gambar 19 dapat dilihat bahwa hasil akhir akurasi dari data training dan data validasi sangat tinggi yaitu mencapai 98%. Untuk membuktikan apakah data sudah sesuai maka kita akan membuat grafik terhadap loss dan acuracy.



Gambar 19. Grafik akurasi model



Gambar 20. Grafik loss model

Pada gambar 19 dan 20 membuktikan bahwa baik dari grafik akurasi ataupun grafik loss data training terhadap data validation bagus. Yang artinya tidak underfitting ataupun overfitting.

E. Evaluation

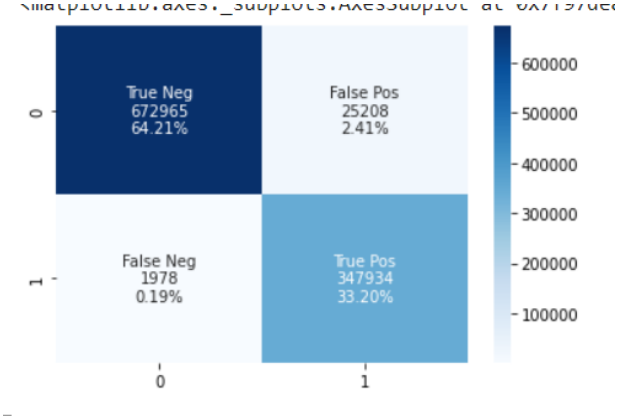
Setelah dirasa menghasilkan hasil yang cukup baik, maka selanjutnya akan dievaluasi terhadap data testing menggunakan confussion matrix.

```
[84] from sklearn.metrics import classification_report
     print(classification_report(y_test, yhat_probs))
```

	precision	recall	f1-score	support
0	1.00	0.96	0.98	698173
1	0.93	0.99	0.96	349912
accuracy			0.97	1048085
macro avg	0.96	0.98	0.97	1048085
weighted avg	0.98	0.97	0.97	1048085

Gambar 21. Classification matrix

Dari gambar 21 dapat dilihat bahwa hasil untuk memprediksi data testing adalah baik, dengan akurasi sebesar 0,97%.



Gambar 22. Heatmap confussion matrix.

Pada gambar 22 dapat diilihat bahwa model memprediksi True negatif sebesar 64% dan true positif sebesar 33%. Sedangkan false positif hanya 2,4% dan false negatif 0,19%.

V. KESIMPULAN

Dari hasil penelitian yang sudah dilakukan, maka dapat disimpulkan bahwa model RNN LSTM bisa digunakan untuk mengklasifikasikan serangan low-rate DDOS dengan menggunakan seleksi fitur. Data yang digunakan adalah CICIDS2017 dan CICIDS2018, yang harus dilakukan pembersihan data serta balancing data. Karena jumlah data benign dan data serangan sangat berbanding jauh, yaitu sekitar 10 kali lipat lebih banyak data berlabel benign. Setelah dilakukan pemodelan dan evaluasi model, akurasi yang didapat cukup tinggi. Akurasi data training terhadap data validasi sekitar 98% dan setelah dilakukan visualisasi data terhadap akurasi

https://doi.org/10.51967/tanesa.v23i2.1737 © 2022 Politeknik Pertanian Negeri Samarinda

dan loss, dapat disimpulkan bahwa model tersebut cukup baik, alias tidak terjadi underfitting ataupun overfitting. Sedangkan akurasi yang didapatkan terhadap data testing adalah sebesar 0,97%. Dari hasil tersebut maka dapat disimpulkan bahwa model LSTM dapat digunakan untuk mengklafisikan serangan DDOS.

LSTM sebenarnya juga dapat memprediksi data sequential ataupun data yang berurut, maka diharapkan untuk penelitian selanjutnya dapat dilakukan identifikasi serangan low-rate DDOS dengan menggunakan fitur timeseries, dan low-rate DDOS akan diidentifikasi berdasarkan waktunya.

DAFTAR PUSTAKA

Aljuhani, A. (2021). Machine Learning Approaches for Combating Distributes Denial of Service Attacks in Modern Networking Environments. *IEEE Access*.

Darryl, P., & Subali, M. (2021). Perbandingan Algoritma SVM dan Algoritma KNN dalam Menghasilkan Klasifikasi DDoS dan Benign. *Jurnal Ilmiah KOMPUTASI, 20*(https://doi.org/http://dx.doi.org/10.32409/jikstik.20.4.2799), 491-500.

Firmansyah, M. R., & al, e. (2020). Klasifikasi Kalimat Ilmiah Menggunakan Recurent Neural Network. *Industrial Research Workshop and National Seminar*.

Gupta, B. B., & Dahiya, A. (2021). *Distributed Denial of Service (DDoS) Attacks.* CRC Press.

Junaedi, R., & Fratelli. (2019). Deteksi serangan DDOS (Distributed Denial of Service) di cloud computing dengan metode rule base.

Li, D., & Li, Z. (2018). A Lightweight Traffic Anomaly Detection Model in SDN Based on Decision Tree. *Atlantis Press: Advances in Computer Science Research*.

Nugraha, F. A., & al, e. (2020). *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning.* Bandung: Kreatif Industri Nusantara.

Nugroho, R. (2022). *Foreign Policy (Implementasi, Evaluasi, dan Manajemen Kebijakan).* PT. Elex Media Komputindo.

Perez-Diaz, J. A., & all, e. (2020). A Flexible SDN-Based Architecture for Identifying and Mitigating Low Rate DDoS Attacks Using Machine Learning. *IEEE Access.*

Puspita, H., & al, e. (2022). *Pengantar Teknologi Informasi.* Sukabumi: penerbithaura.com.

Raharjo, B. (2022). *Deep Learning dengan Python.* Semarang: Yayasan Prima Agus Teknik.

Rahmatullah, F. (2022). *DETEKSI DISTRIBUTED DENIAL OF SERVICE (DDOS) DALAM JARINGAN SOFTWARE DEFINED NETWORK DENGAN METODE SUPPORT VECTOR MACHINE.* Yogyakarta: UPN "Veteran".

Riadi, I., Umar, R., & Aini, F. D. (2019). Analisis Perbandingan Detecsion Traffic Anomaly Dengan Metode Naïve Bayes dan Support Vector Machine (SVM). *ILKOM Jurnal Ilmiah*.

Sarno, R., & al, e. (2022). *Machine Learning Deep Learning Konsep dan Pemrograman Python.* Penerbit: ANDI.

Shandi, M. G., & al, e. (2020). Penerapan Long Short Term Memory untuk Memprediksi Flight Delay pada Penerbangan Komersial. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*.

Sudar, K. M., & Deepalakshmi, P. (2022). Flow Based Detection and Mitigation of LowRate DDOS Attack in SDN Environtment Using Machine Learning Techniques. *IOT and Analytics for Sensor Networks, Lecture Notes in Networks and System*, 193.

Sultana, N., & al, e. (2018). Survey on SDN based Network intrusion detection system using machine learning approaches. *Springer Science + Bussiness media*.

Swe, M. Y., & al, e. (2021). A Slow DDOS Attack Detection Mechanism using Feature Weighing and Ranking. *Proceedings of the 11th Annual International Conference on Industrial Engineering and Operations Management* (p. 4500). Singapore: IEOM Society International.

Valdinos, I. A., Perez-Diaz, J. A., Choo, K.-K. R., & Botero, J. F. (2021). Emerging DDOS attack detection and mitigation strategies in software-defined network: Taxonomy, Challenges and Future Directions. *ELSEVIER:Journal of Network and Computer Applications, 187*.

Valdinos, I. A., Perez-Diaz, Y. A., Cho, K.-K. R., & Botero, J. F. (2021). Emerging DDOS attack detection and mitigation strategies in software-defined network: Taxonomy, challenges dan future directions. *ELSEVIER:Journal of Network and Computer Applications*.

Zhijun, W., & al, e. (2020). Low-Rate DoS Attacks, Detection, Defense, and Challenges: A Survey. *IEEE Access*, 43920 - 43943.