

Implementasi Long Short Term Memory dalam Deteksi Serangan DDoS pada Virtual Private Server

Barry Ananda Moses Sihombing¹, Naikson Fandier Saragih², Indra M Sarkis³
^{1,2,3}Fakultas Ilmu Komputer, Universitas Methodist Indonesia

Info Artikel

Histori Artikel:

Received, Nov 20, 2023
Revised, Des 30, 2023
Accepted, Jan 15, 2024

Keywords:

Long Short Term Memory, Recurrent Neural Network, DDoS Detection, Intrusion Detection System, Klasifikasi, Confusion Matrix, Virtual Private Server

ABSTRAK

Sistem Deteksi Intrusi (IDS) memegang peranan penting dalam menjaga keamanan sistem, terutama pada server. Serangan DDoS seringkali menjadi ancaman yang harus dihadapi, dimana serangan tersebut bertujuan untuk merusak ketersediaan sistem. Dalam rangka memperkuat IDS yang ada, metode Long Short Term Memory (LSTM), sebuah pendekatan dari Jaringan Saraf Rekuren (RNN), muncul sebagai salah satu solusi yang menjanjikan. LSTM memiliki keunggulan dalam mempertahankan informasi jangka panjang yang dibutuhkan untuk mendeteksi pola serangan, serta mampu mengatasi masalah vanishing gradient yang sering terjadi pada RNN, sehingga mengurangi risiko peningkatan loss saat proses training. Penelitian ini menggunakan dataset hasil serangan yang dilakukan pada Virtual Private Server (VPS), dengan tiga kelas serangan: normal, TCP DDoS, dan UDP DDoS. Jumlah total data adalah 70.343, di mana 80% di antaranya digunakan sebagai data latih (56274 data) dan 20% sebagai data uji (14069 data). Dari tujuh kali percobaan, pengujian dengan tingkat akurasi maksimal sebesar 96,35% diperoleh pada epoch 50 dan batch size 64. Selain itu, dilakukan pula pengujian dengan menggunakan dataset yang berasal dari situs <https://www.unb.ca/cic/dataset/ids-2018.html>, di mana 80% data latih berjumlah 838860 rekaman dan 20% data uji berasal dari serangan DDoS pada VPS sebanyak 14069 data. Hasil dari tujuh kali percobaan pengujian menunjukkan tingkat akurasi maksimal sebesar 50,42% dengan nilai epoch 5 dan batch size 64.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Penulis Koresponden:

Naikson Fandier Saragih
Fakultas Ilmu Komputer,
Universitas Methodist Indonesia, Medan,
Jl. Hang Tuah No.8, Medan - Sumatera Utara.
Email: saragihnaikson@gmail.com

1. PENDAHULUAN

Dengan pesatnya pertumbuhan pengguna internet di Indonesia, jika kejahatan di dunia maya juga semakin meningkat. Ada beberapa kasus serangan yang sering terjadi terutama di web, seperti serangan distributed denial of service (DDoS), SQL injection, cross-site scripting (XSS), dan lainnya. Salah satu jenis serangan dengan intensitas yang cukup besar yaitu serangan Denial of Service. Serangan Denial of Service merupakan percobaan yang dilakukan oleh peretas untuk melumpuhkan sistem target dengan cara menghabiskan jaringan atau sumber daya dari sistem tersebut. Jika serangan ini dilakukan dengan lebih dari satu mesin, maka disebut dengan serangan Distributed Denial of Service (DDoS) [1].

Serangan cyber memang tidak bisa dihindarkan, namun dapat diantisipasi dengan membangun suatu sistem yang dapat mendeteksi kinerja aliran data jaringan agar pengguna dapat terhindar dari segala macam bentuk serangan dan usaha-usaha penyusupan dari pihak yang tidak dikenali. Salah satu cara dari ancaman serangan ini adalah Intrusion Detection System (IDS). Intrusion Detection System (IDS) adalah salah satu cara bagaimana mendeteksi sebuah serangan apabila terjadi serangan pada sebuah server. Sistem teknologi di era zaman sekarang membuat IDS memiliki keterbatasan dalam penerapannya dikarenakan serangan-serangan pada jaringan atau server semakin canggih, sehingga diperlukan pengembangan atau cara lain dalam mendeteksi serangan.

Deep Learning adalah bidang ilmu komputer yang menggunakan teknik statistik untuk memberikan kemampuan sistem komputer untuk belajar dari data, tanpa diprogram secara eksplisit. Banyak bidang yang telah menerapkan pembelajaran mesin, salah satunya diterapkan pada masalah IDS dengan harapan dapat meningkatkan tingkat deteksi dan kemampuan klasifikasi.

Salah satu jenis Deep Learning adalah Recurrent Neural Network (RNN) yang belakangan ini telah diterapkan menjadi IDS. RNN adalah salah satu bagian dari keluarga Neural Network untuk memproses data yang bersambung (*sequential data*). Cara yang dilakukan RNN untuk dapat menyimpan informasi dari masa lalu adalah dengan melakukan looping di dalam arsitekturnya, yang secara otomatis membuat informasi dari masa lalu tetap tersimpan. Jenis RNN yang digunakan adalah LSTM (Long Short Term Memory) untuk menutupi kekurangan pada RNN yang tidak dapat menyimpan memori untuk dipilah dan menambahkan *mekanisme Attention* [2].

LSTM adalah arsitektur jaringan saraf yang cukup baik untuk memproses data sekuensial. Cara kerja LSTM adalah dengan mengubah data berupa log jaringan menjadi sebuah teks dan angka untuk digunakan sebagai dataset. Dataset yang didapat dari log jaringan akan diubah menjadi dataset berformat csv dan selanjutnya di proses menggunakan metode LSTM untuk mendapatkan sebuah tingkat akurasi [3].

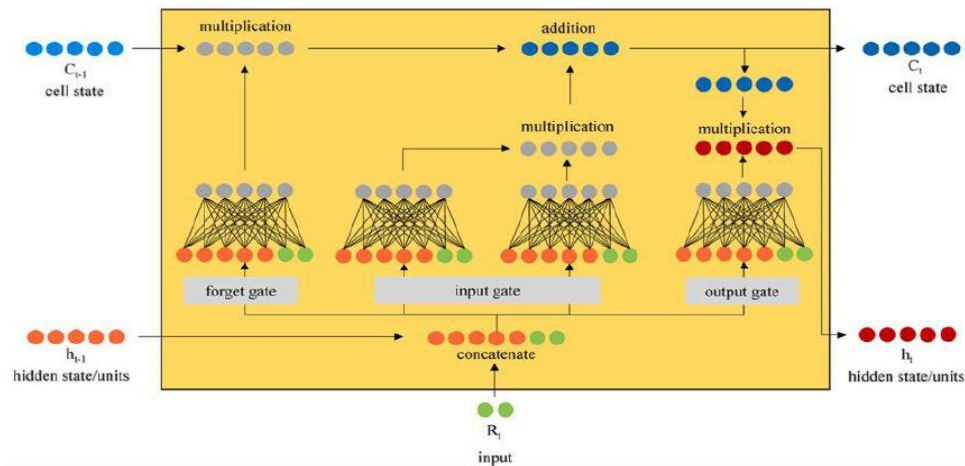
2. METODE PENELITIAN

2.1 Long Short Term Memory (LSTM)

Long Short Term Memory (LSTM) adalah metode yang diusulkan oleh Sepp Hochreiter dan Jurgen Schmidhuber pada tahun 1997. Metode ini merupakan pengembangan dari Recurrent Neural Network (RNN). RNN tidak dapat belajar menghubungkan informasi karena memori lama yang tersimpan akan semakin tidak berguna dan tertimpa dengan memori baru. Sedangkan, LSTM tidak memiliki kekurangan tersebut karena dapat mengatur memori pada setiap masukannya dengan menggunakan memory cells dan gate units [4].

LSTM digunakan untuk mengatasi vanishing gradient atau situasi dimana nilai gradient bernilai 0 atau mendekati 0 dengan mekanisme gate. LSTM merupakan cara lain untuk menghitung hidden state. Memori dalam LSTM disebut dengan cells yang mengambil input dari state sebelumnya (h_{t-1}) dan input saat ini (x_t). Kumpulan cells tersebut memutuskan apa yang akan disimpan dalam memori dan kapan yang akan dihapus dari memori. LSTM menggabungkan state sebelumnya, memori saat ini, dan input. LSTM sangat efisien untuk merekam long-term dependencies terlihat[5].

Berikut adalah gambar arsitektur long short term memory.



Gambar 1. Arsitektur Long short term memory
 Sumber : Laras Wiranda,dkk (2019)

Input gate berfungsi mengontrol berapa banyak informasi yang harus disimpan dalam keadaan sel. Ini mencegah sel dari menyimpan data yang tidak perlu. Forget gate berfungsi mengontrol sejauh mana nilai tetap di dalam sel memori. Output Gate berfungsi untuk memutuskan berapa banyak konten atau nilai dalam sel memori, digunakan untuk menghitung output [6].

2.2 Confusion Matrix

Confusion matrix adalah suatu tabel yang berisikan informasi mengenai banyaknya data yang diprediksi dengan benar dan salah. Untuk menghitung performa dari proses pendeteksian serangan pada penelitian ini digunakanlah Confusion matrix. Proses pendeteksian yang dilakukan dengan menganalisis hasil dengan output 0, 1 Dan 2. [7] Confusion Matrix dapat dilihat pada Tabel 1

Tabel 1. Confusion Matrix

TN	FP1	FP2
FN1	TP1	TPf2
FN2	TPf1	TP2

Dalam prediksi yang dilakukan oleh model klasifikasi, Confusion Matrix akan menganalisis seberapa baik model klasifikasi yang dibuat dengan melihat nilai:

1. Accuracy mengukur seberapa akurat model dapat mengklasifikasikan data dengan benar.

$$\text{Accuracy} = (TP+TN) / (TP+FP+FN+TN)$$
2. Precision (Positive Predictive Value) menggambarkan tingkat keakuratan antara data prediksi benar positif yang diminta dengan hasil prediksi yang diberikan oleh model.

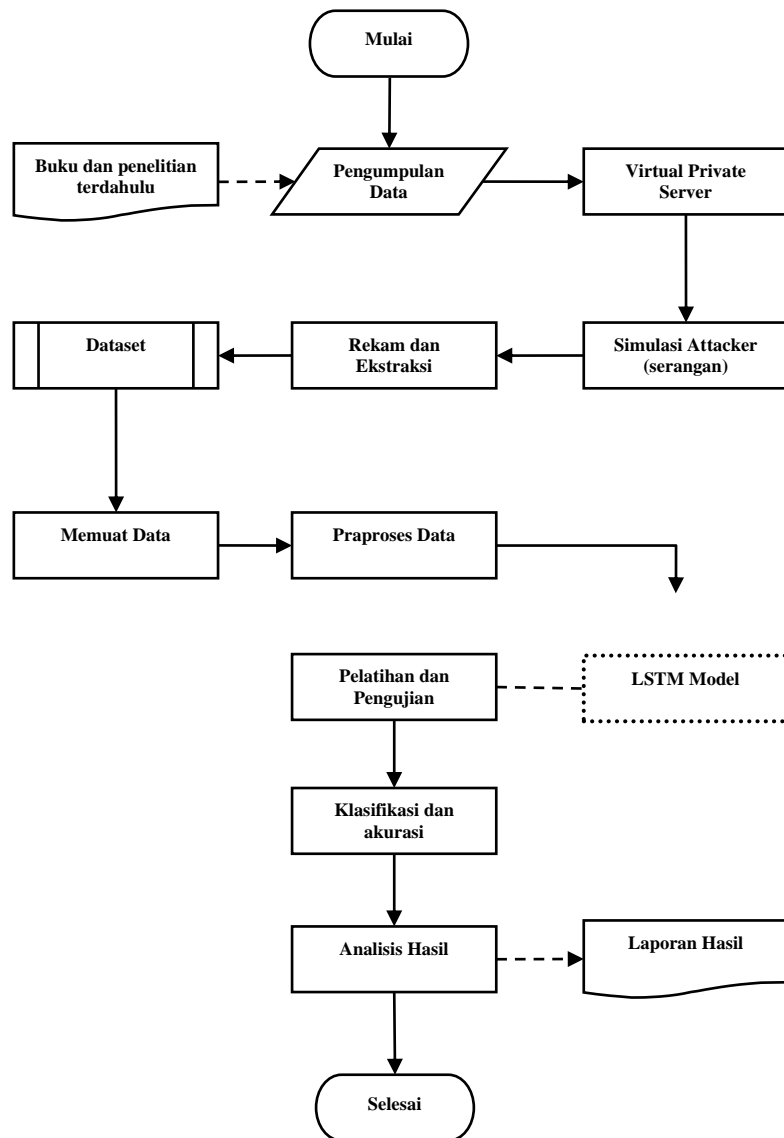
$$\text{Precision} = (TP) / (TP + FP)$$
3. Recall atau sensitivity (True Positive Rate) menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

$$\text{Recall} = TP / (TP + FN)$$
4. F-1 Score menggambarkan perbandingan rata-rata precision dan recall yang dibobotkan.

$$\text{F-1 Score} = (2 * \text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})$$

2.3 Framework Penelitian

Framework penelitian adalah kerangka kerja atau tahapan-tahapan yang jelas untuk membantu mempermudah peneliti. Kerangka kerja ini adalah tahapan-tahapan yang dilakukan dalam penyelesaian masalah yang dibahas.



Gambar 2. Framework Penelitian

3. HASIL DAN PEMBAHASAN

3.1. Normalisasi Data

Pada tahap ini , setiap data akan dinormalisasikan dengan interval 0 dan 1. Pada penelitian ini normalisasi data menggunakan rumus Min Max . Cara kerjanya setiap nilai pada sebuah fitur dikurangi dengan nilai minimum fitur tersebut, kemudian dibagi dengan rentang nilai atau nilai maksimum dikurangi nilai minimum dari fitur tersebut. Teknik normalisasi dengan min max memiliki rumus :

$$X^I = (X - X_{min}) / (X_{max} - X_{min})$$

Dimana:

X' = Data hasil normalisasi

X = Data asli

X_{min} = nilai minimum dari data x

X_{max} = nilai maximum dari data x

3.2. Implementasi Sistem

Implementasi dilakukan setelah perancangan selesai. Tahapan awal dilakukan dengan menyewa jasa layanan VPS dari provider herza.id. Bahasa pemrograman yang akan digunakan adalah bahasa pemrograman Python dengan menggunakan fitur google collab.

3.3 Pengujian Sistem

Untuk mendapatkan dataset LSTM dalam menemukan bukti serangan , peneliti terlebih dahulu melakukan serangan DDoS TCP flooding dan UDP flooding.

- Pengujian Serangan DDoS TCP dan DDoS UDP menggunakan tool LOIC.
- Pengujian aliran Normal

3.4 Tampilan Sistem

1. Dataset Serangan DDoS pada VPS

Pembagian dataset Serangan DDoS VPS antara data latih dan data uji yaitu dengan split 70:30. Berikut ini adalah proses pembagian dataset :

```
[ ] # Split dataset into training set and test set
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=2)

[ ] X_test.shape

(14069, 81)

[ ] X_train.shape

(56274, 81)
```

Gambar 4. Pembagian Data latih dan Data uji.

Pada gambar 4. Menunjukkan proses pembagian data latih dan data uji dengan perbandingan 80:20 dengan jumlah data latih sebanyak 56274 baris data dan data uji sebanyak 14069 baris data.

```
# try using different optimizers and different optimizer configs
from tensorflow import keras
model.compile(loss='categorical_crossentropy',optimizer=keras.optimizers.SGD(learning_rate=0.001),metrics=['accuracy'])
csv_logger = CSVLogger('training_set_inanalysis.csv',separator=',', append=False)
history = model.fit(X_train, y_train, batch_size=64, epochs=50, validation_data=(X_test, y_test),callbacks=[csv_logger])
model.save("kddresults/lstm1layer/fullmodel/lstm1layer_model.hdf5")

Epoch 1/50
880/880 [=====] - 12s 9ms/step - loss: 1.0610 - accuracy: 0.6803 - val_loss: 1.0428 - val_accuracy: 0.7323
Epoch 2/50
880/880 [=====] - 4s 5ms/step - loss: 1.0266 - accuracy: 0.7059 - val_loss: 1.0129 - val_accuracy: 0.7411
Epoch 3/50
880/880 [=====] - 3s 4ms/step - loss: 0.9986 - accuracy: 0.7150 - val_loss: 0.9871 - val_accuracy: 0.7448
Epoch 4/50
880/880 [=====] - 3s 4ms/step - loss: 0.9735 - accuracy: 0.7308 - val_loss: 0.9628 - val_accuracy: 0.7468
Epoch 5/50
880/880 [=====] - 3s 3ms/step - loss: 0.9491 - accuracy: 0.7447 - val_loss: 0.9386 - val_accuracy: 0.7482
Epoch 6/50
880/880 [=====] - 3s 3ms/step - loss: 0.9250 - accuracy: 0.7503 - val_loss: 0.9136 - val_accuracy: 0.7498
Epoch 7/50
880/880 [=====] - 4s 4ms/step - loss: 0.8997 - accuracy: 0.7546 - val_loss: 0.8870 - val_accuracy: 0.7510
Epoch 8/50
880/880 [=====] - 3s 3ms/step - loss: 0.8724 - accuracy: 0.7581 - val_loss: 0.8581 - val_accuracy: 0.7524
Epoch 9/50
880/880 [=====] - 3s 3ms/step - loss: 0.8428 - accuracy: 0.7616 - val_loss: 0.8269 - val_accuracy: 0.7549
Epoch 10/50
880/880 [=====] - 3s 4ms/step - loss: 0.8110 - accuracy: 0.7698 - val_loss: 0.7932 - val_accuracy: 0.7605
Epoch 11/50
880/880 [=====] - 3s 4ms/step - loss: 0.7760 - accuracy: 0.7882 - val_loss: 0.7570 - val_accuracy: 0.7748
Epoch 12/50
880/880 [=====] - 3s 4ms/step - loss: 0.7408 - accuracy: 0.8106 - val_loss: 0.7190 - val_accuracy: 0.8020
```

Gambar 5. Hasil Pelatihan dan pengujian dataset VPS dengan epoch 50

Pada gambar 5. merupakan gambar dari proses pembelajaran mesin dengan menggunakan model LSTM , fungsi loss categorical crossentropy, fungsi optimizer adam, learning rate 0,001 batchsize 64, dan epoch sebanyak 50 . Adapun tampilan hasil akurasi dan tingkat loss nya adalah sebagai berikut :

```
Epoch 40/50
880/880 [=====] - 3s 3ms/step - loss: 0.2182 - accuracy: 0.9643 - val_loss: 0.1917 - val_accuracy: 0.9639
Epoch 41/50
880/880 [=====] - 3s 3ms/step - loss: 0.2163 - accuracy: 0.9643 - val_loss: 0.1886 - val_accuracy: 0.9640
Epoch 42/50
880/880 [=====] - 3s 4ms/step - loss: 0.2118 - accuracy: 0.9643 - val_loss: 0.1858 - val_accuracy: 0.9640
Epoch 43/50
880/880 [=====] - 3s 4ms/step - loss: 0.2100 - accuracy: 0.9646 - val_loss: 0.1832 - val_accuracy: 0.9640
Epoch 44/50
880/880 [=====] - 3s 3ms/step - loss: 0.2071 - accuracy: 0.9647 - val_loss: 0.1808 - val_accuracy: 0.9640
Epoch 45/50
880/880 [=====] - 3s 3ms/step - loss: 0.2053 - accuracy: 0.9650 - val_loss: 0.1786 - val_accuracy: 0.9641
Epoch 46/50
880/880 [=====] - 3s 4ms/step - loss: 0.2020 - accuracy: 0.9650 - val_loss: 0.1766 - val_accuracy: 0.9642
Epoch 47/50
880/880 [=====] - 4s 4ms/step - loss: 0.2000 - accuracy: 0.9650 - val_loss: 0.1747 - val_accuracy: 0.9642
Epoch 48/50
880/880 [=====] - 3s 3ms/step - loss: 0.1987 - accuracy: 0.9651 - val_loss: 0.1730 - val_accuracy: 0.9642
Epoch 49/50
880/880 [=====] - 4s 5ms/step - loss: 0.1956 - accuracy: 0.9653 - val_loss: 0.1714 - val_accuracy: 0.9643
Epoch 50/50
880/880 [=====] - 4s 4ms/step - loss: 0.1946 - accuracy: 0.9652 - val_loss: 0.1699 - val_accuracy: 0.9645

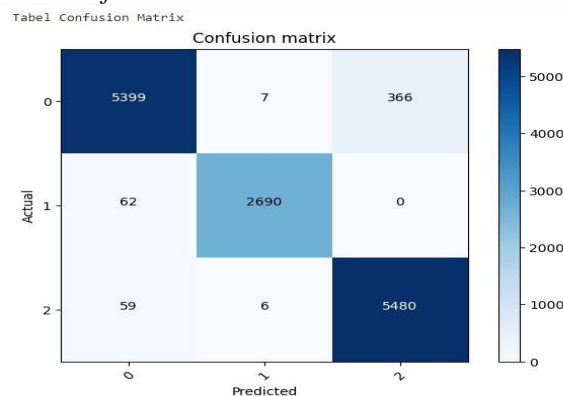
[27] loss, accuracy = model.evaluate(X_test, y_test)
print("\nLoss: %.2f, Accuracy: %.2f%%" % (loss, accuracy*100))

440/440 [=====] - 1s 2ms/step - loss: 0.1699 - accuracy: 0.9645

Loss: 0.17, Accuracy: 96.45%
```

Gambar 6. Hasil akurasi dan loss dataset VPS

Pada gambar 6 menunjukkan bahwa hasil pengujian dataset VPS menggunakan model LSTM adalah 96,45 % . Berikut ini adalah hasil *Confusion Matrix* :



Gambar 7. Hasil Confusion Matrix dataset VPS

Dapat dilihat pada gambar 7 hasil confusion matrix pendeteksian klasifikasi serangan DDoS pada dataset VPS sebanyak 5399 data untuk kelas Normal (Benign), 2690 data untuk kelas serangan DDoS TCP Flood, dan 5480 data untuk kelas seranga UDP Flood.

2. Dataset pada IDS 02-21-2018 dan Dataset serangan DDoS pada VPS

Pembagian dataset IDS 02-21-2018 sebagai Latih dan dataset serangan DDoS VPS sebagai uji yaitu dengan split 80% : 20%. Berikut ini adalah gambar pembagian jumlah komposisi data latih dan data uji pada program python dengan menggunakan fitur *google collab*:

```
[21] datatrain = datatrain.sample(frac=0.8, random_state=25)

[22] datatrain.shape

(838860, 79)

[23] datatest = datatest.sample(frac=0.2, random_state=25)

[24] datatest.shape

(14069, 79)
```

Gambar 8. Pembagian Dataset IDS dan VPS

Pada gambar 8 menunjukkan pembagian dataset dengan perbandingan 80:20, dimana jumlah dataset IDS (*training*) sebanyak 838860 baris data sedangkan dataset VPS (*testing*) sebanyak 14069 baris data.

```
[48] # try using different optimizers and different optimizer configs
from tensorflow import keras
model.compile(loss='categorical_crossentropy',optimizer=keras.optimizers.SGD(learning_rate=0.001),metrics=['accuracy'])
csv_logger = CSVLogger('training_set_iranalysis.csv',separator=',', append=False)
history = model.fit(X_train, y_train, batch_size=64, epochs=5, validation_data=(X_test, y_test),callbacks=[csv_logger])
model.save("kddresults/lstm1layer/fullmodel/lstm1layer_model.hdf5")

Epoch 1/5
13108/13108 [=====] - 50s 4ms/step - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.9487 - val_accuracy: 0.5032
Epoch 2/5
13108/13108 [=====] - 43s 3ms/step - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.9442 - val_accuracy: 0.5035
Epoch 3/5
13108/13108 [=====] - 42s 3ms/step - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.9394 - val_accuracy: 0.5037
Epoch 4/5
13108/13108 [=====] - 43s 3ms/step - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.9347 - val_accuracy: 0.5039
Epoch 5/5
13108/13108 [=====] - 43s 3ms/step - loss: 0.0026 - accuracy: 0.9994 - val_loss: 0.9304 - val_accuracy: 0.5042
```

Gambar 9. Hasil setiap epoch pada dataset IDS 2018 (*training*) dan DDoS VPS (*testing*)

Pada gambar 9 telah dilakukan proses pelatihan dan pengujian menggunakan model LSTM, fungsi loss categorical crossentropy, fungsi optimizer adam, metrics accuracy, batchsize 64, dan epoch sebanyak 5. Adapun gambar nilai akurasi adalah sebagai berikut :

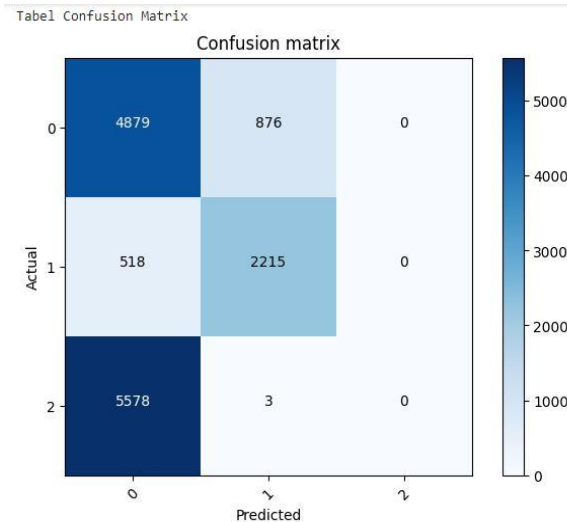
```
[49] loss, accuracy = model.evaluate(X_test, y_test)
print("\nLoss: %.2f, Accuracy: %.2f%%" % (loss, accuracy*100))

440/440 [=====] - 1s 1ms/step - loss: 0.9304 - accuracy: 0.5042

Loss: 0.93, Accuracy: 50.42%
```

Gambar 10. Hasil Akurasi Dataset IDS 2018 dan Dataset DDoS VPS

Pada gambar 13 Hasil nilai akurasi yang didapat dari proses pelatihan dan pengujian model LSTM adalah 50,42%. Berikut gambar menunjukkan hasil dari Confusion Matrix deteksi serangan DDoS menggunakan LSTM:



Gambar 11. Hasil Confusion matrix dataset IDS 2018 dan DDoS VPS

Dapat dilihat pada gambar 14 hasil confusion matrix dataset IDS serangan DDoS 02-21-2018 sebagai training 80% dan dataset serangan DDoS pada VPS sebagai testing 20% hasil pendeteksian klasifikasi prediksi benar didapat sebanyak 2215 data untuk kelas serangan DDoS dan sebanyak 4879 data untuk kelas Normal (Benign).

4. KESIMPULAN

Hasil Implementasi Long Short Term Memory dalam Deteksi Serangan DDoS pada Virtual Private Server menggunakan metode LSTM memberikan gambaran yang signifikan, seperti berikut:

1. Dengan menggunakan dataset serangan DDoS pada VPS, dengan pembagian data 80:20 untuk training dan pengujian, sebanyak 56274 baris data untuk training dan 14069 baris data untuk pengujian, model LSTM yang diimplementasikan dengan 50 epoch dan batch size 64 berhasil mencapai tingkat akurasi sebesar 96.45% berdasarkan analisis menggunakan confusion matrix. Tingkat akurasi yang mendekati nilai 100% ini menunjukkan kualitas yang sangat baik dalam penggunaan model LSTM untuk deteksi serangan DDoS.
2. Pengujian menggunakan Dataset IDS DDoS tanggal 21 Februari 2018 sebagai data latih, dan dataset serangan DDoS pada VPS sebagai data uji, dengan pembagian data 80:20, menunjukkan hasil yang berbeda. Dengan menggunakan model LSTM yang sama dengan 5 epoch dan batch size 64, tingkat akurasi yang dicapai adalah 50.42% berdasarkan analisis confusion matrix. Hasil ini menempatkan kinerja sistem dalam kategori yang lebih moderat.

Dari kedua hasil percobaan ini, dapat disimpulkan bahwa penggunaan model LSTM dalam deteksi serangan DDoS menunjukkan potensi yang signifikan untuk meningkatkan keamanan sistem, terutama pada penggunaan dataset VPS. Meskipun demikian, performa sistem dapat dipengaruhi oleh sumber data yang digunakan, seperti yang terlihat pada pengujian dengan dataset IDS DDoS, di mana performa model LSTM menunjukkan hasil yang lebih menengah.

REFERENSI

- [1] H. Hafid, "Investigasi Log Jaringan Untuk Deteksi Serangan Distributed Denial of Service (Ddos) Dengan Menggunakan Metode General Regression Neural Network," 2019, [Online]. Available: <http://etheses.uin-malang.ac.id/16609/>
- [2] K. Ivanedra, M. Mustikasari, T. Informatika, U. Gunadarma, R. N. Network, and D. Learning, "Implementasi Metode Recurrent Neural Network Pada Text the Implementation of Text Summarization With Abstractive," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 4, 2019, doi: 10.25126/jtiik.201961067.
- [3] M. R. Firmansyah, R. Ilyas, and F. Kasyidi, "Klasifikasi Kalimat Ilmiah Menggunakan Recurrent Neural Network," *Pros. 11th Ind. Res. Work. Natl. Semin.*, vol. 11, no. 1, pp. 488–495, 2020.
- [4] J. K. Lubis and I. Kharisudin, "Metode Long Short Term Memory dan Generalized Autoregressive Conditional Heteroscedasticity untuk Pemodelan Data Saham," *Prism. Pros. Semin. Nas. ...*, vol. 4, pp. 652–658, 2021, [Online]. Available: <https://journal.unnes.ac.id/sju/index.php/prisma/article/view/44897>
- [5] M. Abdul Dwiyanto Suyudi, E. C. Djamal, A. Maspupah Jurusan Informatika, and F. Sains dan Informatika Universitas Jenderal Achmad Yani Cimahi, "Prediksi Harga Saham menggunakan Metode Recurrent Neural Network," *Semin. Nas. Apl. Teknol. Inf.*, pp. 1907–5022, 2019.
- [6] L. Wiranda and M. Sadikin, "Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma," *J. Nas. Pendidik. Tek. Inform.*, vol. 8, no. 3, pp. 184–196, 2019.
- [7] F. Rahman, "Deteksi anomali pada data web traffic menggunakan long short term memory bertumpuk," *Repository.Uinjkt.Ac.Id*, 2021, [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/56731>