JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Deteksi Serangan pada Jaringan Kompleks IoT menggunakan Recurrent Neural Network

Eko Arip Winanto[1], Kurniabudi[1,*], Sharipuddin[2], Ibnu Sani Wijaya[2], Dodi Sandra[2]

[1]Fakultas Ilmu Komputer, Sistem Komputer, Universitas Dinamika Bangsa, Jambi, Indonesia
[2]Fakultas Ilmu Komputer, Teknik Informatika, Universitas Dinamika Bangsa, Jambi, Indonesia
Email: [1]ekoaripwinanto@unama.ac.id, [2,*]kbudiz@yahoo.com, [3]sharipuddin@unama.ac.id, [4]ibnu_sw@unama.ac.id, [5]doedy234@unama.ac.id
Email Penulis Korespondensi :kbudiz@yahoo.com
Submitted **08-12-2022**; Accepted **25-12-2022**; Published **30-12-2022**

Abstrak

Jaringan yang kompleks dalam Internet of Things sendiri menjadi tantangan untuk menjaga keamanan dalam jaringan. Dengan kompleksitas jaringan termasuk data, protokol, ukuran, komunikasi, standar, dan lainnya, menjadi sulit untuk menerapkan sistem deteksi intrusi (IDS). Salah satu cara untuk meningkatkan IDS pada jaringan IoT yang kompleks adalah dengan menggunakan metode deep learning untuk mendeteksi serangan yang terjadi pada jaringan kompleks IoT. Recurrent neural network (RNN) adalah metode deep learning yang meningkatkan deteksi jaringan IoT yang kompleks karena memperhitungkan input saat ini dan juga apa yang telah dipelajari dari input yang diterima sebelumnya. Saat membuat keputusan tentang RNN, pertimbangkan masukan saat ini serta apa yang telah dipelajari dari masukan yang diterima sebelumnya. Oleh karena itu, penelitian ini mengusulkan metode RNN sebagai metode untuk meningkatkan kinerja sistem deteksi serangan pada jaringan IoT yang kompleks. Hasil dari pengujian ini menunjukkan hasil yang memuaskan dengan meningkatkan performa dari sistem deteksi akurasi pada jaringan kompleks IoT yang mencapai 87 %.

Kata Kunci: IDS; deep learning; RNN; Complex networks; IoT

Abstract

The complex network in the Internet of Things is challenging to maintain network security. With network complexity including data, protocols, sizes, communications, standards, and more, it becomes difficult to implement an intrusion detection system (IDS). One way to improve IDS on complex IoT networks is by using deep learning to detect attacks that occur on complex IoT networks. Recurrent neural network (RNN) is a deep learning method that enhances the detection of complex IoT networks because it takes into account the current input as well as what has been learned from previously received inputs. When making decisions about RNNs, consider current information as well as what has been learned from previous input. Therefore, this study proposes the RNN method to improve the performance of attack detection systems on complex IoT networks. The results of this experiment show satisfactory results by increasing the performance of the accuracy detection system in complex IoT networks which reaches 87%.

Keywords: IDS; deep learning; RNN; Complex networks; IoT

1. PENDAHULUAN

Internet of Things (IoT) memasuki periode yang berkembang pesat dengan ekspansi yang luar biasa dalam segi ukuran, ruang lingkup dan kompleksitas dan terhubung ke jaringan serta menjangkau berbagai tingkat organisasi [1]. Kompleksitas pada jaringan IoT *(complex network)* tidak hanya terbatas dari perbedaan perangkat tetapi juga dari services, protocol, jalur komunikasi, data, tipe jaringan dan lainnya. Dikarenakan kompleksitas dari jaringan IoT ini maka salah satu tantangan pada jaringan komplek IoT adalah isu tentang keamanan.

Deteksi serangan pada jaringan kompleks IoT perlu dilakukan karena jaringan IoT yang kompleks seringkali terdiri dari banyak perangkat yang terhubung ke internet dan saling terkoneksi satu sama lain. Hal ini membuat jaringan tersebut rentan terhadap serangan cyber, seperti serangan malware, phishing, dan serangan Denial of Service (DoS). Serangan tersebut dapat merusak perangkat IoT, mengakses data pribadi yang tersimpan di perangkat, atau bahkan memanfaatkan kelemahan keamanan dari perangkat untuk masuk ke jaringan lain. Oleh karena itu, penting untuk mengadakan deteksi serangan pada jaringan kompleks IoT agar dapat mengidentifikasi dan menangani serangan sebelum mereka dapat merusak sistem atau mengakses data yang tidak seharusnya.

Untuk mengadakan deteksi serangan pada jaringan kompleks IoT, perusahaan atau organisasi dapat menggunakan berbagai alat keamanan seperti sistem deteksi intrusi (IDS) untuk memonitor aktivitas jaringan dan mengidentifikasi serangan yang mungkin terjadi. *Intrusion detection system (IDS)* adalah sebuah sistem yang sangat penting dalam jaringan komplek IoT yang berfungsi untuk mendeteksi sebuah tindakan kejahatan yang terjadi didalam jaringan [2][3]. Dengan kompleksitas jaringan di dalam IoT juga menjadi tantangan dalam membangun sistem IDS [4] yang efisien dan memiliki performa yang tinggi. Salah satu solusi yang menjanjikan adalah *deep learning* [5].

Pada penelitian [6-7] menunjukkan bahwa penerapan *deep learning* dapat diimplementasikan kedalam klasifikasi lalu lintas jaringan dan intrusion detection systems (IDS). Terdapat beberapa penelitian sebelumnya yang sudah menggunakan deep learning. Pada penelitian [8] mengusulkan pendekatan berbasis *deep belief network* (DBN) untuk mendeteksi serangan di jaringan IoT. Begitu pula [9][10] telah mengusulkan metode DBN pada jaringan IoT. Selain itu pada [11] mengusulkan model hibrid untuk meningkatkan kinerja dari DBN dengan autoencoder, hasil deteksi menunjukan bahwa terdapat peningkatan kinerja dari pada DBN tunggal. Ada pun gap penelitian IDS deep learning yang perlu ditingkatkan adalah akurasi deteksi pada jaringan kompleks IoT.

JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

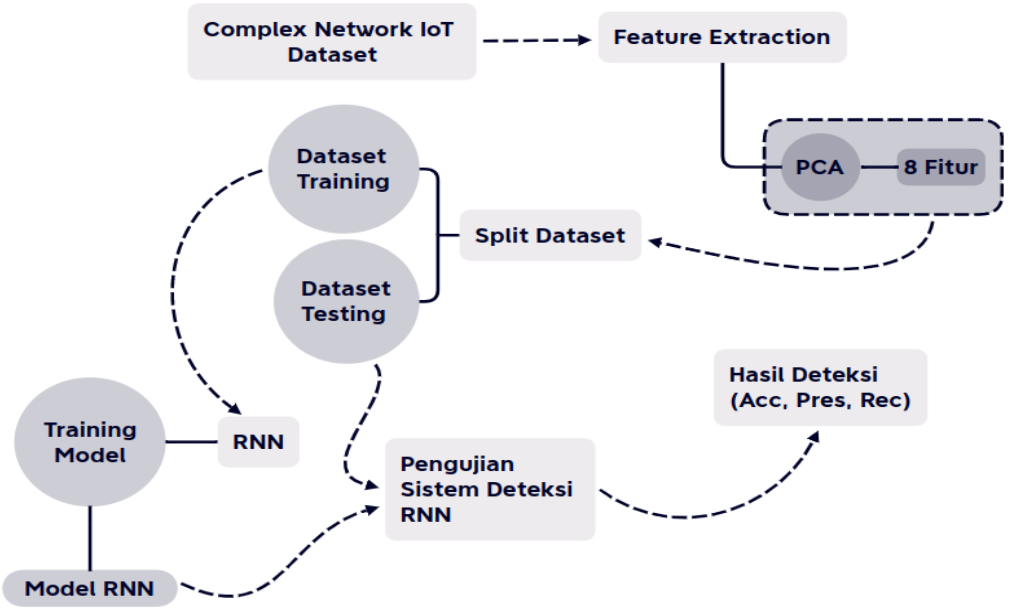
Menurut penelitian [12] salah satu metode *deep learning* yaitu *recurrent neural network (RNN)* memiliki fitur lebih yang dapat digunakan untuk meningkatkan performa dari deteksi serangan yaitu RNN dapat mempertimbangkan input saat ini dan juga apa yang telah dipelajari dari input yang diterima sebelumnya untuk mendeteksi sebuah serangan. Oleh karena itu, penelitian ini melakukan deteksi serangan pada jaringan komplek IoT menggunakan metode *deep learning* yaitu RNN. Tujuan dari penelitian ini adalah untuk meningkatkan kinerja dari sistem deteksi serangan pada jaringan komplek IoT menggunakan RNN. Pemilihan metode RNN ini dikarenakan RNN memiliki karakteristik yang sesuai untuk sistem deteksi serangan pada jaringan komplek IoT [13]. Pada metode RNN perlu adanya pemahaman tentang konsep bagaimana metode ini bekerja untuk deteksi deteksi serangan pada jaringan komplek IoT. Terdapat dua konsep dalam RNN dan metode *deep learning* lainnya yaitu *RNN and feed-forward neural networks* yang sesuai namanya dari cara mereka menyebarkan informasi. Dalam *feed-forward neural networks*, informasi hanya bergerak dalam satu arah dari lapisan input, melalui lapisan tersembunyi, ke lapisan output. Informasi bergerak langsung melalui jaringan. *Feed-forward neural networks* tidak memiliki memori input yang mereka terima dan buruk dalam memprediksi apa yang akan terjadi selanjutnya. Karena *feed-forward neural networks* hanya mempertimbangkan input saat ini dan tidak memiliki fitur tentang urutan waktu. *Feed-forward neural networks* tidak bisa mengingat apapun tentang apa yang terjadi di masa lalu kecuali pelatihannya. Sedangkan dalam RNN, informasi berputar melalui sebuah loop [14]. Ketika membuat keputusan, RNN mempertimbangkan input saat ini dan juga apa yang telah dipelajari dari input yang diterima sebelumnya. Sehingga pada penelitian ini mengusulkan metode RNN sebagai metode untuk meningkatkan performa dari sistem deteksi serangan pada jaringan komplek IoT. Selain itu penelitian ini memiliki beberapa kontribusi adalah : Melakukan ekstraksi fitur pada jaringan kompleks IoT menggunakan PCA dan mengusulkan sistem deteksi pada jaringan komplek IoT menggunakan RNN.

Sisa makalah ini disusun sebagai berikut. Bagian 2 secara singkat membahas dataset dan setup eksperimental yang digunakan dalam penelitian ini. Bagian 3 menjelaskan lebih detail tentang eksperimen dan hasil temuan penelitian ini. Akhirnya, Bagian 4 memberikan kesimpulan dan pekerjaan potensial di masa depan.

2. METODOLOGI PENELITIAN

2.1 Experiment Setup

Penelitian ini bertujuan untuk mengusulkan metode RNN untuk mendeteksi serangan yang terjadi pada jaringan kompleks IoT. Ada beberapa tahapan penelitian yang harus dilakukan untuk dapat mencapai tujuan dari penelitian ini, maka perlu dirancang alur penelitian yang ditampilkan pada Gambar 1.



Gambar 1. Konfigurasi Eksperimen

Gambar 1 adalah konfigurasi eksperimen yang didesain pada penelitian ini. Oleh karena itu pada penelitian ini dapat dibagi menjadi tiga tahapan yaitu:

a. Melakukan ekstraksi fitur dari dataset komplek IoT menggunakan metode PCA serta membagi dataset untuk data training dan data testing.
b. Selanjutnya adalah melakukan training model RNN untuk deteksi serangan menggunakan dataset training sehingga diperoleh model RNN yang akan digunakan untuk proses pengujian.
c. Terakhir, melakukan pengujian menggunakan model RNN dengan diujikan menggunakan dataset training serta menghitung keberhasilan seperti akurasi, presisi dan *recall.*

JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

2.2 Dataset

Penelitian ini akan menggunakan dataset IoT yang kompleks dari Comnets Lab Unsri [8][10]. Untuk merepresentasikan jaringan IoT yang kompleks di lingkungan nyata dalam kumpulan data ini, beberapa perangkat keras digunakan: sensor (kelembaban tanah, MQ2, Funduino, DHT22, dll.), perangkat node (PC, Raspy dan Arduino). *Middleware* meliputi: *XBee, w1d D1* dan *Wi-Fi* untuk menghubungkan antara *middleware* dan server. Pada dataset ini terdapat beberapa skenario serangan dan tipe serangan yang jinak, *TCP flood* dan *zbassocflood* pada *Xbee* disajikan pada Tabel 1.

Tabel 1. Jenis jenis database

No	Nama Dataset	Jenis Serangan	Jumlah paket
1	*normalserver*	*Benign*	12792
2	*seranganserver*	*TCP flood, Benign*	3135393
3	*normalnode_xbee*	*Benign*	568
4	*serangannode_xbee*	*Zbassocflood, Benign*	19426
5	*Normal_serangan_node_xbee*	*Zbassocflood, Benign*	22441
6	*normalmid1*	*Benign*	1739
7	*seranganmid1*	*TCP flood, Benign*	1175059
8	*Normal_serangan_server*	*TCP flood, Benign*	1191320
9	*normalmid2*	*Benign*	2102
10	*seranganmid2*	*TCP flood, Benign*	1555706
11	*Normal_serangan_mid2*	*TCP flood, Benign*	1603038
12	*normalnode_wifi*	*Benign*	7806
13	*serangannode_wifi*	*TCP flood, Benign*	2399420
14	*Normal_serangan_node_wifi*	*TCP flood, Benign*	2426599

Pada Tabel 1 adalah jenis dataset yang terdiri dari beberapa jenis dari dataset serangan *TCP flood, benign dan zbassocflood*. Kemudian dataset ini berjumlah 14 file dataset yang terdiri dari paket serangan dan paket normal. Jenis dataset IoT ini memiliki karakteristik yaitu dataset protocol WIFI dan Xbee.

2.3 *Features Extraction* Menggunakan PCA

Pemilihan fitur merupakan proses yang sangat penting dalam sebuah *Intrusion Detection System*, dan performa atau akurasi dari sebuah *Intrusion Detection System (IDS)* akan sangat bervariasi ketika dilengkapi dengan input fitur yang beragam. Selain itu, lalu lintas yang padat di jaringan kompleks di *Internet of Things* dan fitur multidimensi akan mempengaruhi hasil proses klasifikasi [15]. Dengan banyaknya data yang diproses di IDS, maka perlu dilakukan ekstraksi fitur untuk mengurangi biaya komputasi saat memproses data mentah di sistem sensor IoT [16]. Ekstraksi fitur bertujuan untuk mengekstraksi fitur dari fitur asli yang sudah ada dan memodifikasi fitur pada ukuran yang lebih kecil untuk mempercepat proses pelatihan dan meningkatkan akurasi [17][18]. Pada penelitian mengusulkan *Principal Components Analysis* (PCA) untuk mengurangi dimensi dari dataset. Detail dari penggunaan *pseudocode PCA* pada penelitian ini ditampilkan di bawah ini. Pada penelitian ini akan mereduksi dimensi menjadi 8 fitur dan akan digunakan pada proses training dan deteksi.

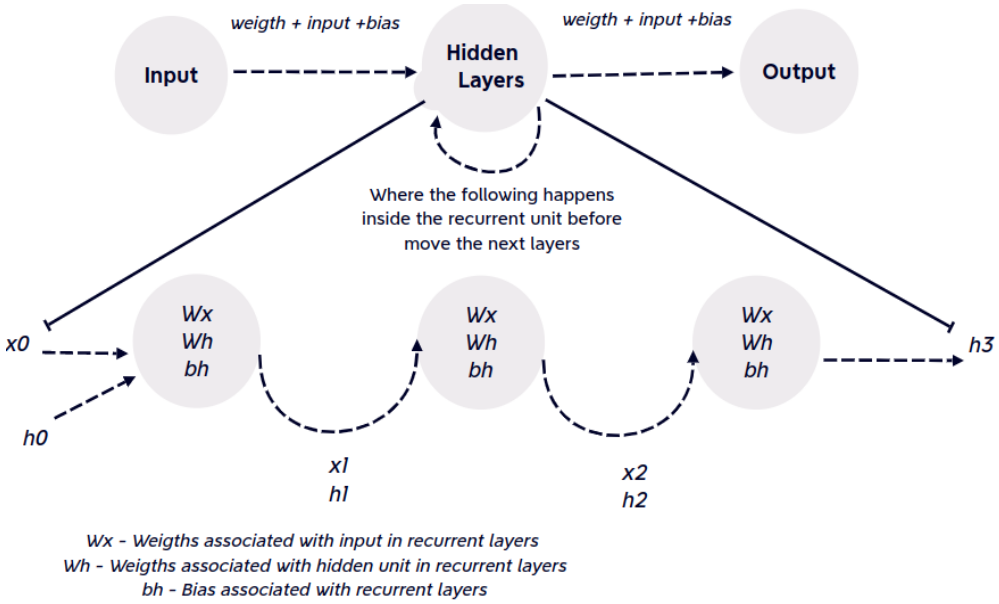
Tabel 2. Pseudocode PCA

Pseudocode PCA
sklearn.decomposition import PCA
dataset ← load dataset
(x_train, y_train),(x_test, y_test) ← dataset_IoT
pca ←PCA(n_componnets=8)
x_pca ← pca.fit_transform((x_train, y_train),(x_test, y_test))

2.4 Sistem Deteksi Menggunakan RNN

Recurrent Neural Network (RNN) adalah salah satu algoritma dari *deep learning* yang lagi trend saat ini. Terdapat beberapa metode *deep learning* yang telah dikenalkan oleh para peneliti sebelumnya seperti *Deep Belief Network (DNN)* [19] , *AutoEncoder* [20], *Convolution Neural Network (CNN)* [21] dan lainnya. RNN memiliki alur tahapan yang terdiri dari tiga layers yaitu input layer, hidden layer dan output layers seperti yang terdapat pada gambar 2. Fitur dari RNN yang membedakan dari metode *deep learning* yang lainnya adalah ada *weights hidden* atau bobot unit dalam hidden layers. Didalam *hidden layers RNN* memungkinkan hasil output dalam *hidden layers* sebelumnya (*Wh)* digunakan untuk proses *training* pada *hidden layers* tersebut, sehingga hasil dari proses training menjadi lebih baik karena menggunakan pengetahuan dari proses training sebelumnya.

JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom



Gambar 2. Desain dari algoritma RNN

Tahapan umum pada algoritma RNN untuk mendeteksi serangan pada jaringan komplek IoT dapat dibagi menjadi tiga tahapan yaitu :

a. Inputs - Di metode RNN inputan akan berupa dataset yang telah di fitur ekstraksi menggunakan PCA dengan recurrent layer, yaitu [x0, x1, x2], …, [x_{n-2 }, x_{n-1}, x_{n}].
b. *Recurrent layer* - dalam *feed-forward neural network*, node tersembunyi akan memiliki dua parameter: bobot dan bias. Namun, *recurrent layer* memiliki tiga parameter untuk dioptimalkan: bobot untuk input, bobot untuk unit tersembunyi, dan bias. hal ini akan tetap menjadi tiga parameter meskipun memiliki jumlah node yang lebih banyak.
c. Training - *feedforward neural network* biasa dilatih menggunakan algoritma backpropagation. Sementara itu, melatih RNN menggunakan versi *backpropagation* yang sedikit dimodifikasi, yang mencakup pembukaan waktu untuk melatih bobot jaringan. Algoritma didasarkan pada komputasi vektor gradien dan disebut *backpropagation* in time atau disingkat BPTT.

Pada penelitian ini mengusulkan menggunakan RNN untuk mendeteksi serangan pada jaringan kompleks IoT, untuk lebih detail dari metode RNN yang didesain untuk penelitian ini dapat dilihat pada pseudocode RNN dibawah ini.

Tabel 2. Pseudocode RNN

Pseudocode RNN
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM
(x_train, y_train),(x_test, y_test) ← dataset_IoT
model = Sequential()
model.add(LSTM(10/8,input_shape ← (x_train.shape[1:]), activation ←'relu', return_sequences ←True))
model.add(Dropout(0.2))
model.add(LSTM(128, activation ← 'relu'))
model.add(Dropout(0.1))
model.add(Dense(32, activation ← 'relu'))
model.add(Dropout(0.2))
model.add(Dense(10, activation ←'softmax'))
opt ← tf.keras.optimizers.Adam(lr ← 0.001, decay ←1e-6)
model.compile(loss←'sparse_categorical_crossentropy', optimizer←opt, metrics←['accuracy'])
model.fit(x_train, y_train, epochs←3, validation_data=(x_test, y_test))
classification_report(y_test,y_predict,target_names=serangan)
confusion_matrix(y_test, y_predict)

2.5 *Environment Setup*

Semua *experiment* di dalam percobaan ini dilakukan pada komputer yang menjalankan sistem operasi Ubuntu 20 LTS, dengan spesifikasi prosesor Intel Core i7 2,60 GHz dan RAM 12 GB. Untuk keperluan analisis, digunakan python dengan perangkat lunak seleksi fitur, scikit belajar, *tensorflow* dan Keras untuk RNN.

JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

3. HASIL DAN PEMBAHASAN

Bagian ini membahas hasil percobaan yang dilakukan. Pembahasan meliputi hasil reduksi fitur dan pengujian kinerja algoritma RNN.

3.1 Hasil Features Extraction

Pada tahap ini dataset harus dikonversi ke dimensi yang lebih kecil. Tujuannya adalah untuk mengurangi beban komputasi dan meningkatkan kinerja sistem deteksi pada jaringan kompleks IoT. Penelitian ini mengusulkan penggunaan metode PCA untuk mereduksi dimensi atribut dataset tanpa menghilangkan karakteristik datanya.

Tabel 3. Hasil PCA

Jumlah fitur	Hasil PCA
8	0.03906421, -0.03521358, -0.01257638, -0.03453761, -0.02599266, -0.00253226, 0.06773333, -0.01350402
8	-0.07097025, -0.04464164, 0.03906215, -0.03371358, -0.01257658, -0.06450761, -0.06832974, -0.09220405 0.04445121,
8	-0.01357658, -0.03450761, -0.07832974, -0.09225405, 0.07090025, -0.74464164, 0.03905621, 0.07010025,

Tabel 3 menunjukkan hasil ekstraksi fitur yang dilakukan dengan metode PCA. Dari tabel hasil dapat diketahui bahwa dalam penelitian ini mengkonversi menjadi 8 atribut. Hasil ekstraksi fitur ini digunakan untuk mengolah data training IDS menggunakan RNN. Pada proses PCA, record diubah menjadi nilai dengan range 0 sampai 1. Setelah konversi, nilai dari fitur menjadi range yang lebih kecil. Tujuannya adalah untuk mengurangi sumber daya IDS menggunakan metode RNN.

3.2 Hasil Pengujian Sistem Deteksi

Dalam studi ini, dua langkah harus diambil untuk menggunakan RNN untuk deteksi serangan di jaringan IoT yang kompleks. Langkah pertama adalah melakukan pembelajaran untuk mendapatkan bobot dan bias hirarki jaringan RNN. Hasil pembelajaran DBN berupa nilai bobot dan bias yang digunakan dalam proses penemuan jaringan IoT yang kompleks. Berikut adalah proses deteksi yang digunakan RNN untuk mendeteksi serangan pada jaringan IoT yang kompleks.

Pengujian selanjutnya adalah pengujian IDS pada jaringan IoT yang kompleks dengan input dataset hasil ekstraksi fitur menggunakan PCA. Hasilnya berasal dari pengujian yang menunjukkan bahwa RNN dapat mendeteksi serangan dengan tingkat keberhasilan yang cukup tinggi pada catatan TCP dengan kesalahan deteksi yang rendah. Hal ini dapat diketahui dari nilai FP dan FN terkecil pada tabel 3 yaitu konfusi matriks. Saat menguji dataset Xbee, paket normal memiliki beberapa kesalahan deteksi. Ini terjadi untuk kumpulan data yang berisi data normal dan serangan.

Tabel 4. Hasil Confusion Matrix Pengujian Sistem deteksi RNN

No	Dataset	TP	TN	FP	FN
1	normalserver	0	5098	0	0
2	seranganserver	1249585	4498	0	0
3	normalnode_xbee	0	202	0	0
4	serangannode_xbee	3622	3667	109	350
5	Normal_serangan_node_xbee	4397	4130	182	413
6	normalmid1	0	749	0	0
7	seranganmid1	374585	95358	0	0
8	Normal_serangan_server	1478604	5211	0	0
9	normalmid2	0	836	0	0
10	seranganmid2	489381	132810	0	0
11	Normal_serangan_mid2	499745	141391	1	0
12	normalnode_wifi	0	3111	0	0
13	serangannode_wifi	895720	63967	0	0
14	Normal_serangan_node_wifi	911032	59553	0	0

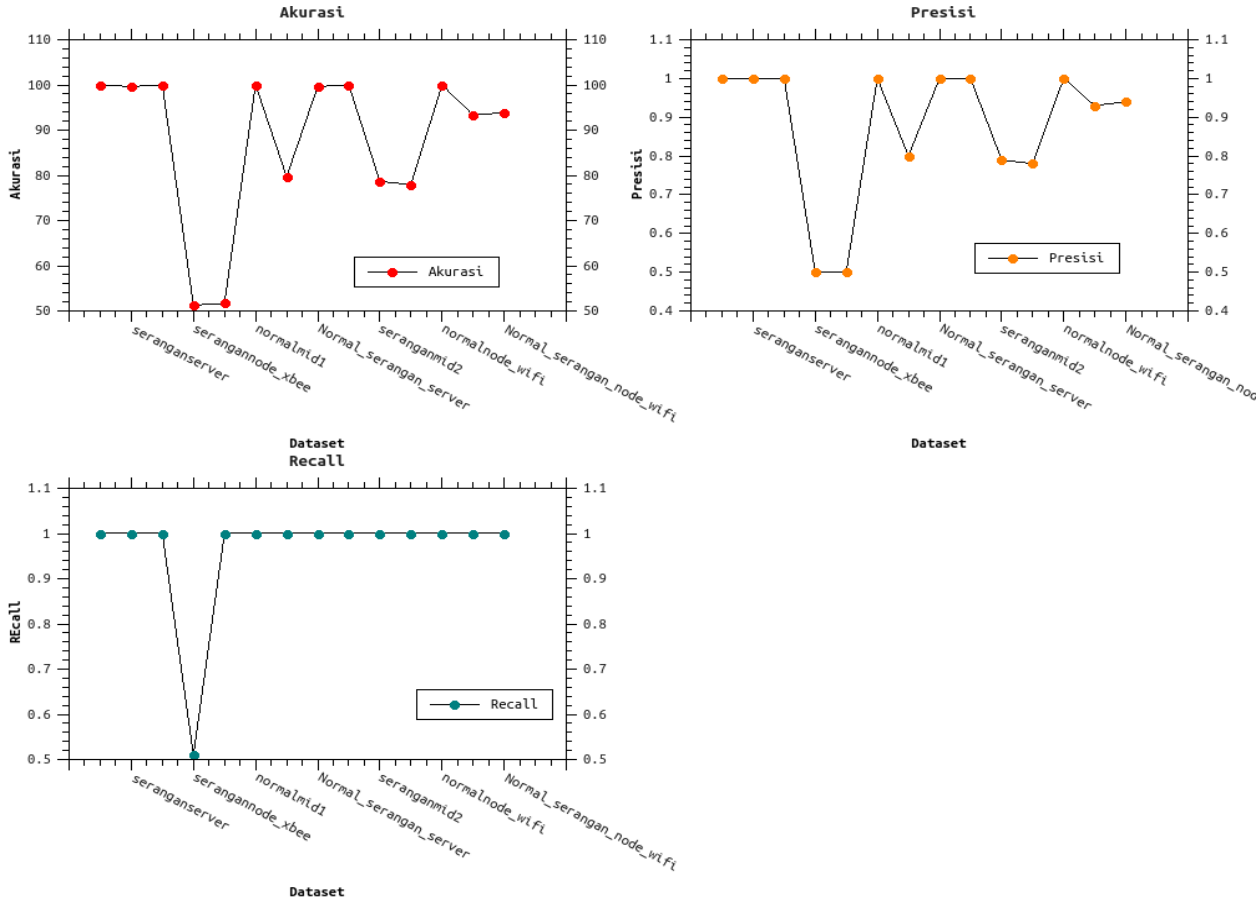
Tabel 4 adalah hasil konfusi matriks yang memberikan hasil yang kurang efektif karena kesalahan deteksi paket normal masih terjadi tetapi masih dalam batas yang dapat ditoleransi. Pada Tabel 4 menampilkan hasil FA dan FN pada dataset normal_serangan_node_xbee dan serangannode_xbee yang cukup tinggi sehingga ini diartikan sebagai hasil deteksi yang kurang optimal.

Tabel 5 adalah hasil pengujian dari pengujian RNN pada IDS jaringan kompleks IoT. Hasil pengujian terdiri dari 3 parameter yaitu akurasi, presisi dan recall. hasil pengujian diperoleh akurasi yang cukup memuaskan untuk beberapa dataset. Hasil akurasi terbaik diperoleh pada dataset yang memiliki variasi paket yang bervariasi. Hasil rata-rata akurasi pada pengujian deteksi menggunakan RNN pada jaringan kompleks IoT mencapai 87.55%. Selain itu pada parameter presisi dan recall memperoleh hasil yang cukup memuaskan. Hasil yang kurang memuaskan diperoleh pada dataset xbee yang hanya mencapai 51% serta pada beberapa dataset mid yang bertipe paket WIFI.

JURIKOM (Jurnal Riset Komputer), Vol. 9 No. 6, Desember 2022
e-ISSN 2715-7393 (Media Online), p-ISSN 2407-389X (Media Cetak)
DOI 10.30865/jurikom.v9i6.5298
Hal 1996−2002
http://ejurnal.stmik-budidarma.ac.id/index.php/jurikom

Tabel 5. Hasil Pengujian Sistem deteksi RNN

No	Dataset	Akurasi	Presisi	Recall
1	normalserver	100	1.00	1.00
2	seranganserver	99.64	1.00	1.00
3	normalnode_xbee	100	1.00	1.00
4	serangannode_xbee	51.26	0.50	0.51
5	Normal_serangan_node_xbee	51.70	0.50	1.00
6	normalmid1	100	1.00	1.00
7	seranganmid1	79.72	0.80	1.00
8	Normal_serangan_server	99.64	1.00	1.00
9	normalmid2	100	1.00	1.00
10	seranganmid2	78.65	0.79	1.00
11	Normal_serangan_mid2	77.94	0.78	1.00
12	normalnode_wifi	100	1.00	1.00
13	serangannode_wifi	93.33	0.93	1.00
14	Normal_serangan_node_wifi	93.86	0.94	1.00



Gambar 3. Hasil Pengujian Akurasi, Presisi dan *Recall*

Gambar 3 adalah hasil pengujian deteksi serangan pada jaringan komplek IoT menggunakan RNN yaitu akurasi, presisi dan recall. Dari gambar ini dapat dilihat bahwa perbandingan performa dari hasil pengujian pada setiap dataset menggunakan RNN. Hasil akurasi dan presisi menunjukkan performa yang hampir mirip dari pola grafiknya yang mencapai 99 sampai 80 persen. Pada grafik recall menunjukkan kinerja yang setabil yaitu mencapai 1 atau 100 persen. Kemudian pada grafik akurasi dan presisi menunjukkan hasil performa yang kurang stabil untuk setiap dataset akan tetapi masih dalam ambang batas yang bagus. Dari pengujian ini dapat disimpulkan bahwa RNN berhasil mendeteksi serangan yang terjadi pada jaringan kompleks IoT.

4. KESIMPULAN

Jaringan kompleks IoT seperti perbedaan *protokol*, *end-device*, sensor, data yang terkoneksi kedalam jaringan internet maka akan meningkatkan isu dari keamanan. Salah satu solusi yang menjanjikan adalah mengusulkan menggunakan deep learning pada IDS di jaringan kompleks IoT. Pada penelitian ini menggunakan RNN untuk metode deteksi pada jaringan

kompleks IoT. Fokus dari pekerjaan ini adalah meningkatkan performa dari sistem deteksi jaringan kompleks IoT menggunakan RNN. Hasil dari penelitian ini berhasil mendeteksi serangan pada jaringan kompleks IoT. Pada penelitian ini menggunakan tiga parameter pengukuran yaitu akurasi, presisi dan recall. Dari hasil perhitungan akurasi, presisi dan *recall* menunjukkan performa dari RNN dengan *feature extraction (PCA)* berhasil dalam mendeteksi serangan pada jaringan kompleks IoT dengan akurasi mencapai 87.55%. Penelitian kedepan adalah mengusulkan penggunaan RNN untuk jaringan kompleks IoT lebih baik lagi. Salah satunya mengusulkan menggunakan metode *deep learning* yang lain seperti CNN dan fitur seleksi lainnya untuk mengidentifikasi serangan pada jaringan kompleks IoT.

REFERENCES

[1] W. Li, S. Tug, W. Meng, and Y. Wang, "Designing collaborative blockchained signature-based intrusion detection in IoT environments," *Futur. Gener. Comput. Syst.*, vol. 96, pp. 481–489, 2019, doi: 10.1016/j.future.2019.02.064.

[2] E. A. Winanto, M. Y. Idris, D. Stiawan, and M. S. Nurfatih, "Designing Consensus Algorithm for Collaborative Signature- based Intrusion Detection System," vol. 1, 2020.

[3] E. A. Winanto, M. Y. bin Idris, D. Stiawan, M. S. N. Fatih, and Sharipuddin, "PoAS: Enhanced Consensus Algorithm for Collaborative Blockchain Intrusion Detection System," in *2020 3rd International Conference on Information and Communications Technology (ICOIACT)*, Nov. 2020, pp. 513–518, doi: 10.1109/ICOIACT50329.2020.9332078.

[4] E. A. Winanto, D. Stiawan, and A. Heryanto, "Visualisasi Serangan Remote to Local (R2L) Dengan Clustering K-Means," *Pros. Annu. Res. Semin. 2016*, vol. 2, no. 1, pp. 359–362, 2016.

[5] Y. Li, R. Ma, and R. Jiao, "A hybrid malicious code detection method based on deep learning," *Int. J. Secur. its Appl.*, vol. 9, no. 5, pp. 205–216, 2015, doi: 10.14257/ijsia.2015.9.5.21.

[6] I. Sohn, "Deep belief network based intrusion detection techniques: A survey," *Expert Syst. Appl.*, vol. 167, p. 114170, 2021, doi: 10.1016/j.eswa.2020.114170.

[7] S. Mahdavifar and A. A. Ghorbani, "Application of deep learning to cybersecurity: A survey," *Neurocomputing*, vol. 347, pp. 149–176, 2019, doi: 10.1016/j.neucom.2019.02.056.

[8] S. Sharipuddin *et al.*, "Intrusion detection with deep learning on internet of things heterogeneous network," *IAES Int. J. Artif. Intell.*, vol. 10, no. 3, p. 735, 2021, doi: 10.11591/ijai.v10.i3.pp735-742.

[9] N. Balakrishnan, A. Rajendran, D. Pelusi, and V. Ponnusamy, "Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things," *Internet of Things*, no. xxxx, p. 100112, 2019, doi: 10.1016/j.iot.2019.100112.

[10] S. Sharipuddin *et al.*, "Enhanced Deep Learning Intrusion Detection in IoT Heterogeneous Network with Feature Extraction," *Int. J. Electr. Eng. Informatics*, vol. 9, no. 3, pp. 747–755, 2021, doi: 10.52549/ijeei.v9i3.3134.

[11] A. Bhardwaj, V. Mangat, and R. Vig, "Hyperband Tuned Deep Neural Network With Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud," *IEEE Access*, vol. 8, pp. 181916–181929, 2020, doi: 10.1109/access.2020.3028690.

[12] C. Yin, Y. Zhu, J. Fei, and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017, doi: 10.1109/ACCESS.2017.2762418.

[13] H. HaddadPajouh, A. Dehghantanha, R. Khayami, and K. K. R. Choo, "A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting," *Futur. Gener. Comput. Syst.*, vol. 85, pp. 88–96, 2018, doi: 10.1016/j.future.2018.03.007.

[14] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Inf. Sci. (Ny).*, vol. 513, pp. 386–396, 2019, doi: https://doi.org/10.1016/j.ins.2019.10.069.

[15] B. Yan and G. Han, "Effective Feature Extraction via Stacked Sparse Autoencoder to Improve Intrusion Detection System," *IEEE Access*, vol. 6, no. c, pp. 41238–41248, 2018, doi: 10.1109/ACCESS.2018.2858277.

[16] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*, vol. 17, pp. 219–224, 2019, doi: 10.1109/ICECOS.2018.8605181.

[17] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. Bin Bin Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 Dataset Feature Analysis with Information Gain for Anomaly Detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.

[18] S. Sharipuddin *et al.*, "Features Extraction on IoT Intrusion Detection System Using Principal Components Analysis (PCA)," *Proc. EECSI 2020 - 1-2 Oct. 2020*, pp. 114–118, 2020.

[19] T. Kumar, P. Kumar, M. Nappi, and S. Bakshi, "Satellite IoT Based Road Extraction from VHR Images Through Superpixel-CNN Architecture," *Big Data Res.*, vol. 30, p. 100334, 2022, doi: 10.1016/j.bdr.2022.100334.

[20] Y. Meidan, M. Bohadana, and D. Breitenbacher, "N-BaIoT — Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," *IEEE Pervasive Comput.*, no. September, pp. 12–22, 2018.

[21] T. K. Behera, P. K. Sa, M. Nappi, and S. Bakshi, "Satellite IoT Based Road Extraction from VHR Images Through Superpixel-CNN Architecture," *Big Data Res.*, vol. 30, p. 100334, 2022, doi: https://doi.org/10.1016/j.bdr.2022.100334.