Klasifikasi URL Berbahaya dengan Neural Network

PROPOSAL TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Program Studi Informatika

Diajukan oleh:
Mathys Jorge Alberino Seilatu
205314714

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2023

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

ABSTRAK

Tautan keamanan web semakin bertambah setiap harinya, dan deteksi serta analisis halaman web yang berbahaya semakin penting dan sulit dilakukan, *Phishing* adalah salah satu bentuk kejahatan dengan cara mendapatkan informasi sensitif dengan meniru pengirim yang terpercaya di dalam saluran komunikasi, jika dalam dunia internet berarti meniru *domain name* dari sebuah *website* sehingga pengguna internet tertipu dan akhirnya memberikan data pribadi mereka.

Dikarenakan perkembangan teknologi dan membludaknya *URL* (tautan) di internet maka kita tidak dapat melakukan pengecekan satu persatu terhadap tautan tautan yang ada, maka dibangunlah sistem untuk membantu dalam hal tersebut.

Prediksi *URL* berbahaya dibangun menggunakan *Deep Learning* dengan metode campuran *Long Short term Memory* (LSTM) dan *Convolutional Neural Network* (CNN). Pelatihan dan pengujian model tersebut dilakukan pada platform *Kaggle Notebook.* Setelah selesai, model akan dievaluasi akurasi dan f1 skornya.

Kata kuci : tautan berbahaya, *neural network, phishing, fraud detection, kaggle notebook.*

BAB I

PENDAHULUAN

1.1.Latar Belakang

Dengan perkembangan teknologi internet yang terus berlangsung, tautan keamanan *web* semakin bertambah setiap harinya, dan deteksi serta analisis halaman *web* yang berbahaya semakin penting dan sulit dilakukan [1]. *Phishing* adalah salah satu bentuk kejahatan dengan cara mendapatkan informasi sensitif dengan meniru pengirim yang terpercaya di dalam saluran komunikasi. *Phishing* sering digunakan untuk mendapatkan informasi atau uang dengan cara yang tidak benar. Biasanya, pesan yang dikirimkan mengandung *software* atau tautan berbahaya. Lebih mudah untuk membuat laman palsu daripada membobol keamanan sistem. Selain itu, *phishing* dapat diluncurkan dengan biaya yang relatif murah dari mana saja di dunia karena internet yang terbuka dan anonim. Laman *phishing* berisi logo tipe dan teks yang ditujukan untuk meniru laman yang asli [2].

Dikarenakan maraknya hal tersebut menjadikan jenis *fraud* ini mulai berkembang dan diterapkan pada halaman *website* dan ketika diakses melalui URL, halaman dengan *phising* tersebut bisa menyebabkan masalah bagi pengguna, mulai dari menginfeksi komputer pribadi dengan *malware*, mengunduh perangkat lunak mata-mata, mencuri kredensial, hingga pencurian uang dari kartu bank [3]. Salah satu cara paling baik untuk mengategorikan URL yang berbahaya atau tidak adalah dengan *Machine Learning*.

Jaringan Syaraf Tiruan (Neural Network) sering disebut juga *Deep Learning* dan semakin populer dalam penelitian saat ini khususnya dalam klasifikasi teks. *Neural Network* telah menunjukkan kinerja yang sangat baik dalam klasifikasi teks [4]. Bukti yang kuat dapat ditemukan melalui beberapa penelitian yang menunjukkan bahwa penggunaan *Neural Network* dapat menghasilkan akurasi yang sangat tinggi, bahkan melebihi 95% [5]. Sehingga *Neural Network* dapat digunakan untuk mendeteksi URL berbahaya dengan cukup akurat.

1.2.Rumusan Masalah

1. Dapatkah *Neural Network* digunakan untuk mengklasifikasikan url dengan baik?
2. *Variable* apa yang mempengaruhi akurasi dari model yang telah dibuat?

1.3.Batasan Masalah

1. Metode yang digunakan adalah *Deep Learning Neural Network* dengan komposisi layer *word embedding, convolutional, max pooling*, LSTM(*Long Short-Term Memory*), kemudian dilakukan modifikasi *hyper-parameter* dan komposisi *layer* untuk mencari model dengan akurasi yang tinggi.
2. Data yang diguanakan adalah data dari *website* pencatat *link* berbahaya atau *website* penyedia data dan kemudian dikombinasikan dengan

website yang masuk dalam *white list* atau *website* yang sering kita gunakan seperti google.com, youtube.com, dan sebagainya.

1.4.Tujuan Penelitian

1. Untuk mengetahui komposisi *layer Neural Network* apa yang dapat digunakan untuk mengklasifikasikan *URL* berbahya dengan baik.
2. Untuk mengetahui *hyper-parameter* yang paling optimal untuk klasifikasi *URL*.

1.5.Manfaat Penelitian

1. Mengurangi ancaman kejahatan *cyber* bagi pengguna awam.
2. Peningkatan keamanan informasi *online*.

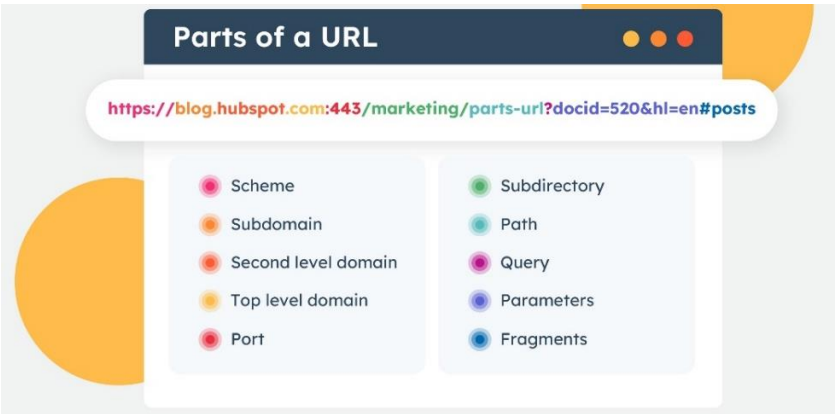
BAB II

TINJAUAN PUSTAKA

2.1. Landasan Teori

2.1.1. URL

URL (*Uniform Resource Locator*) atau tautan atau *link* adalah koneksi antara *resource* satu ke lainya. Awal mula dari tautan adalah World Wide Web (WWW) yang berfungsi untuk melihat multimedia-*based* dokumen yang menampilkan dokumen dengan teks, gambar, animasi atau video. World Wide Web Consortium (W3C) adalah organisasi yang menstandarisasi perihal Hyper-Text Markup Language (XHTML), Cascading Style Sheets (CSS), HyperText Markup Language (HTML) dan Extensible Markup Language (XML). Untuk dapat melihat dokumen yang dipublikan dalam internet adalah dengan cara mengakses alamat URL-nya (URL address), struktur dasar dari URL terdiri dari protokol, *subdomain*, nama *domain*, dan ekstensinya[6].



Gambar 2.1.1 Struktur Tautan (https://blog.hubspot.com/marketing/parts-url)

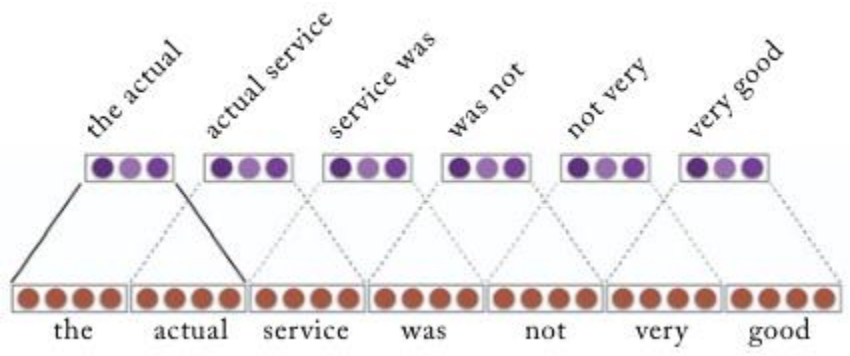
2.1.2. Convolutional Neural Network

CNN adalah tipe lain dari *Deep Neural Network*, jika menggunakan CNN biasanya disertai juga penggunaan *pooling layer*, dimana *pooling layer* tersebut berfungsi untuk membagi region dari *convolutional* menjadi *subregion* agar komputasi menjadi lebih ringan. CNN sangat sukses dalam kasus klasifikasi gambar [7] tetapi bisa juga digunakan dalam natural *language processing* [8].

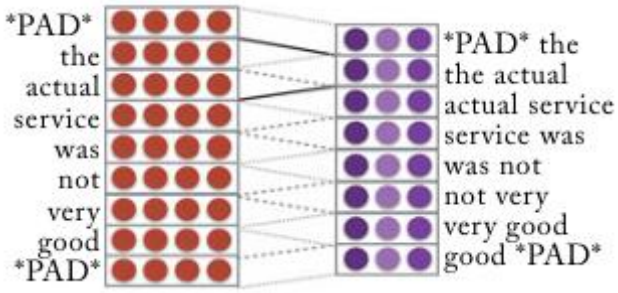
	Recurrent Network	Recursive Neural Tensor Network	Deep Belief Network	Convolution Network	MLP
Text Processing	✔	✔		✔	
Image Recognition			✔	✔	
Object Recognition		✔		✔	
Speech Recognition	✔				
Time Series Analysis	✔				
Classification			✔	✔	✔

Gambar 2.1.2 Use-case tiap Neural Network (T. Oliver, 2020)

Dalam klasifikasi teks, input akan diambil dari *embedding layer* kemudia di proses oleh *convolution layer* dengan perlakuan mirip seperti pemrosesan citra, yaitu dengan melakukan *filtering* pada tiap *window* terhadap *convolutional filter* (*kernel*) [9].



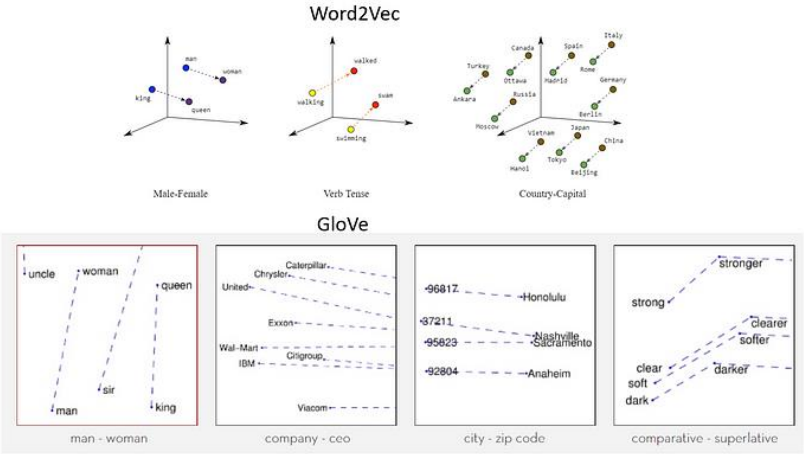
Gambar 2.1.3 CNN 1D (Graeme Hirst, dkk, 2017)



Gambar 2.1.4 CNN 1D 2 (Graeme Hirst, dkk, 2017)

2.1.3. Word Embedding

Word Embedding adalah Layer wajib untuk teks *processing*, karena pada layer ini tiap kalimat akan dikonversi menjadi vector dengan bobot yang dapat diproses oleh *computer* saat *training* [7].



Gambar 2.1.5 Ilustrasi Word Embedding (https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4)

2.1.4. Tokenizing

Tokenization adalah memecah kalimat menjadi kata kemudian melabeli kalimat tersebut dengan nomor agar dapat dikomputasikan [10]. Kemudian setelah dipecah pecah, nomor dari kata tersebut disusun

ulang sesuai dengan susunan kalimat awalnya kedalam list (*teks to siquence*), setelah selesai kalimat kalimat dalam bentuk list tersebut di samakan dimensinya dan dimasukan kedalam list yang sama (*padding*).

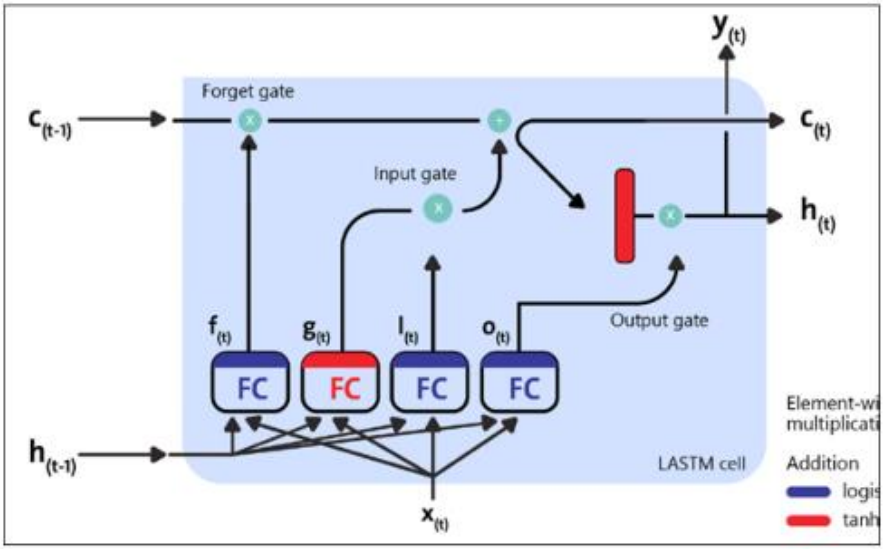
Output:

	I	enjoying	feel	for	freedom	happy	live	love	our	the	we
doc1	1	0	0	0	1	2	0	1	1	1	1	2
doc2	0	1	1	1	0	1	1	0	0	0	0	0

Gambar 2.1.6 Tokenizing (Alexandra George, 2018)

2.1.5. Long Short Term Memory (LSTM)

LSTM adalah salah satu tipe dari RNN yang ideal untuk digunakan dalam prediksi dan kalsifikasi untuk temporal *sequences*. Pada tiap blok LSTM berisikan tiga tipe *gate*: *input gate, output gate*, dan *forget gate* yang mengimplementasikan fungsi tulis, baca, dan reset dalam *cell memory*. *Gate* tesebut bertipe analog dan biasanya digunakan *activation* layer sigmoid [7].



Gambar 2.1.7 Arsitektur LSTM (G. Aurélien, 2019)

2.1.6. Confusion Matrix

Confusion Matrix digunakan untuk mengevaluasi proforma model klasifikasi dengan cara menghitung berapa banyak jumlah kelas A yang di prediksi sebagai kelas B. Rumus yang digunakan adalah:

$$precision = TP * TP + FP \qquad (2.1.6.1)$$

TP = banyak dari benar diprediksi benar

FP = banyak dari benar diprediksi salah.

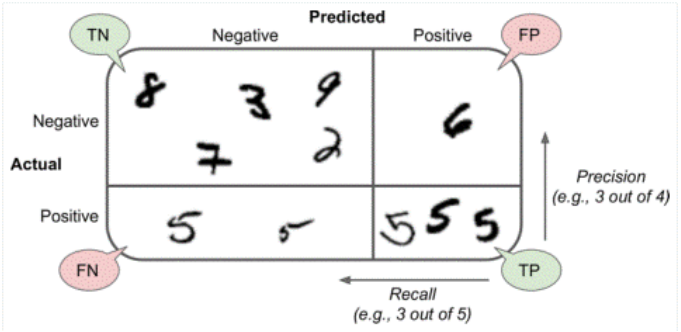
$$recall = TP * TP + FN \qquad (2.1.6.2)$$

FN = banyak dari salah diprediksi benar.

Precicion dan *recall* sering digabung menjadi satu metrik bernama F1 *Score* yang memberikan simplifikasi dengan *harmonic mean* dari *precision* dan *recall* yang berarti F1 *Score* akan tinggi jika nilai dari *precision* dan *recall* juga tinggi [11].

$$F1 = TP * TP + FN + FP * 2 \qquad (2.1.6.3)$$

FP = banyak dari salah diprediksi salah.



Gambar 2.1.8 Confusion Matrix (G. Aurélien, 2019)

2.2. Tinjauan Pustaka

Tabel 2.2.1 Tabel tinjauan pustaka

Peneliti, Judul, Penerbit dan Tahun Terbit	Tujuan penelitian	Data dan Hasil
Judul: **URLDeep: Continuous Prediction of Malicious URL with Dynamic Deep Learning in Social Networks** **Peneliti:** **Putra Wanda and Huang Jin Jie** **Penerbit:** **International Journal of Network Security** **Tahun Terbit:** **2019**	Penelitian ini ditujukan untuk menguji teknik baru Neural Network Dynamic CNN yaitu URLDeep untuk membedakan link berbahaya dan tidak, kemudian mendapat titingkat akurasi yang tinggi dalam mendeteksi URL berbahaya secara kontinu dan dinamis di lingkungan jaringan sosial.	**Populasi:** URL dari VirusTotal and PhishTank **Sample:** 16 juta terdiri dari 14,050,275 (Benign) dan 1.049,725 (Malicious) **Pengukuran:** akurasi (accuracy), presisi (precision), area under the curve (AUC), dan waktu komputasi. **Alat Analisis:** Teknik deep learning berbasis LSTM (Long Short-Term Memory) dan transfer learning untuk membangun model prediksi URL berbahaya secara kontinu dan dinamis. Transfer learning merupakan teknik yang digunakan untuk memanfaatkan model yang sudah dilatih sebelumnya pada masalah serupa untuk mempercepat proses pembelajaran pada masalah yang baru. **Hasil:** Hasil penelitianya adalah bahwa pengembangan metode deteksi URL berbahaya yang menggunakan teknik deep learning berbasis LSTM dan transfer learning dengan fitur-fitur dinamis seperti yang disebutkan tersebut mampu mencapai tingkat akurasi yang tinggi dalam mendeteksi URL berbahaya secara kontinu dan dinamis di lingkungan jaringan sosial. Selain itu, metode ini juga mampu mengatasi permasalahan klasifikasi URL yang bersifat dinamis dan terus berubah seiring waktu. Walaupun demikian, tidak disebutkan berapa akurasi yang didapa**t**

Judul: **Malicious URL Detection Based on Improved Multilayer Recurrent Convolutional Neural Network Model** **Peneliti:** **Zuguo Chen, Yanglong Liu, Chaoyang Chen, Ming Lu and Xuzhuo Zhang** **Penerbit:** **Security and Communication Networks** **Tahun Terbit:** **2021**	Penelitian ini ditujukan untuk memberikan gambaran mengenai potensi dan tantangan penggunaan machine learning dalam keamanan siber, privasi, dan keselamatan publik pada aplikasi yang sedang berkembang, serta untuk mengidentifikasi solusi dan strategi untuk mengatasi tantangan tersebut.	**Populasi:** URL **Sample:** 200,000 URL, dibagi menjadi 100,000 normal URL dilabeli “good” dan 100,000 malicious URL dilabeli “bad” training test perbandingan adalah 9:1 **Pengukuran:** akurasi (accuracy), presisi (precision), recall, area under the curve (AUC), dan waktu komputasi. **Alat Analisis:** CNN menggunakan YOLO algorithm kemudian dikembangkan menjadi CSPDarknet neural network model, kemudian menggunakan bidirectional LSTM recurrent neural network algorithm untuk ekstraksi fitur **Hasil:** Hasil penelitianya adalah bahwa menggunakan metode yang disebutkan menghasilkan efektifitas dan kecepatan deteksi yang lebih baik dan mendapatkan high accuracy, high recall rate, and high accuracy dibanding dengan Text-RCNN, BRNN, dan model lainnya. namun demikian masih diperlukan pengembangan untuk menghindari missing information dari URL teks vectorization. akurasi dengan YOLO CSPDarknet menghasilkan 94% dengan nilai loss value 0.19. kemudian dengan traditional bidirectional recurrent neural network menunjukan indikasi over fitting, dengan RCNN akurasinya adalah 92% dengan loss value 0.22, dengan tradisional RNN menghasilkan 90% dengan minimum loss value 0.32, dengan Neural Network models based on fully connected layers menghasilkan 86% dengan loss value 0.33.

Judul: **A Malicious URL Detection Model Based on Convolutional Neural Network** **Peneliti:** **Zhiqiang Wang, Xiaorui Ren, Shuhao Li, Bingyan Wang, Jianyi Zhang, dan Tao Yang** **Penerbit:** **Security and Communication Networks** **Tahun Terbit:** **2021**	Tujuan penelitian adalah untuk mendesain sebuah model deteksi URL jahat berbasis Convolutional Neural Network yang efektif dan sulit dihindari oleh penyerang. Model ini menggunakan metode embedding kata berbasis embedding karakter untuk mempelajari representasi vektor URL dan menggabungkan layer konvolusi dan pooling untuk mengekstraksi fitur URL dengan lebih efisien. Melalui serangkaian eksperimen, penelitian menunjukkan bahwa model ini mampu mendeteksi URL jahat dengan akurasi yang tinggi (98%)	**Populasi:** URL **Sample:** Tidak disebutkan **Pengukuran:** akurasi (accuracy), presisi (precision), recall, F1-score, area under the curve (AUC), dan waktu komputasi. **Alat Analisis:** Dynamic convolutional neural network (DCNN). memodifikasi pooling layer default menjadi k-max-pooling layer. embedding method **Hasil:** Penelitian ini bertujuan untuk merancang model deteksi URL berbahaya berbasis deep learning dengan menggunakan metode word embedding berdasarkan karakter embedding. Hasil eksperimen menunjukkan model deteksi yang dirancang efektif dalam mendeteksi URL berbahaya. Namun, dengan berkembangnya internet dan semakin beragamnya jenis URL berbahaya, model ini harus diperbarui secara berkala agar tetap efektif dalam berbagai skenario aplikasi yang kompleks.

Judul: **Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network** **Peneliti:** **Huan-huan Wang & Long Yu & Sheng-wei Tian & Yong-fang Peng & Xin-jun Pei** **Penerbit:** **Appl Intell** **Tahun Terbit:** **2019**	Tujuan penelitian adalah untuk mengusulkan algoritma bidirectional LSTM (CBIR) yang berbasis pada convolutional neural network (CNN) dan independent recurrent neural network (RNN) untuk meningkatkan akurasi deteksi halaman web jahat. Algoritma ini menggunakan fitur "texture fingerprint" dan fitur vektor kata URL yang dihasilkan oleh alat word2vec, dan meningkatkan akurasi deteksi halaman web jahat dengan menggunakan algoritma CBIR yang diusulkan.	**Populasi:** URL dari PhishTank dan known URL **Sample:** 23,652 terdiri dari 10,000 (legitimate) dan 13,652 dari PhishTank (Malicious) **Pengukuran:** akurasi (accuracy) dan waktu komputasi. **Alat Analisis:** convolutional neural network (CNN) dan bidirectional LSTM algorithm (CBIR), digunakan pula teknik word embedding dengan memanfaatkan karakter embedding dan word2vec. **Hasil:** Penelitian ini mendapatkan tingkat akurasi paling besar adalah 97.82%, tetapi setelah menggunakan URL word vector feature akurasinya naik menjadi 98.45%. penggunaan algoritma CBIR yang menggabungkan fitur-fitur seperti texture fingerprint, URL word vector, dan static vocabulary dapat meningkatkan akurasi dalam mendeteksi halaman web yang jahat. Hal ini dapat diobservasi melalui hasil eksperimen yang menunjukkan bahwa metode yang diusulkan dalam penelitian ini memiliki hasil yang lebih baik dibandingkan dengan metode-metode lain yang telah diteliti sebelumnya.

Judul: **Accurate and Fast URL Phishing Detector: A Convolutional Neural Network Approach** **Peneliti:** **Wei Wei, Qiao Ke, Jakub Nowak, Marcin Korytkowski, Rafa l Scherer, Marcin Wo´zniak** **Penerbit:** **Computer Networks** **Tahun Terbit:** **2020**	Penelitian ini bertujuan untuk mengembangkan metode pendeteksian situs phishing dengan menggunakan jaringan saraf konvolusi yang dapat mengidentifikasi URL yang berbahaya dengan hampir 100% akurasi hanya dengan menganalisis teks URL. Dalam eksperimen, metode CNN yang diusulkan dibandingkan dengan pendekatan lain, terutama dengan jaringan LSTM yang diusulkan dalam penelitian sebelumnya.	**Populasi:** URL dari PhishTank dan crawling legitimate URL **Sample:** 21,208 terdiri dari 10,604 (Benign) dan 10,604 (Malicious) **Pengukuran:** akurasi (accuracy) **Alat Analisis:** YOLO algorithm, CNN with Embedding Layer, Teknik deep learning berbasis LSTM (Long Short-Term Memory). **Hasil:** Kesimpulan dari penelitian ini adalah bahwa metode yang diusulkan dapat mengidentifikasi situs phishing dengan akurasi hampir 100% dan mampu mendeteksi serangan zero-day. Selain itu, metode ini lebih cepat dibandingkan dengan pendekatan lain yang menganalisis statistik lalu lintas atau konten web. Metode ini juga dapat digunakan pada perangkat mobile tanpa signifikan mempengaruhi kinerjanya. Jaringan saraf konvolusi yang diusulkan juga lebih mudah dilatih dan lebih cepat konvergennya dibandingkan dengan jaringan LSTM yang dibandingkan dalam penelitian sebelumnya.

Judul: **Detecting Malicious URLs Based on Machine Learning Algorithms and Word Embeddings** **Peneliti:** **Andrei Cris¸an, Gabriel Florea, Lorand Halasz, Camelia Lemnaru, Ciprian Oprisa** **Penerbit:** **IEEE** **Tahun Terbit:** **2020**	Mengembangkan model klasifikasi yang handal untuk mendeteksi URL berbahaya dengan memanfaatkan fitur-fitur yang sesuai, baik itu yang sudah ada maupun yang dibangun khusus untuk domain tertentu.	**Populasi:** URL dari PhishTank dan crawling legitimate URL **Sample:** approximately 500,000 URL terdiri dari 10:1 untuk clean dan URL berbahaya, kemudian ada lagi 8 millions URL dengan perbandingan 1:1 untuk clean dan URL berbahaya **Pengukuran:** Precision, accuracy, dan recall rate. **Alat Analisis:** Kombinasi antara word embeddings dan fitur-fitur domain yang telah dirancang khusus, Teknik oversampling sintetis dan cost-sensitive learning digunakan untuk mengatasi ketidakseimbangan kelas. Teknik-teknik klasifikasi yang berbeda juga dieksplorasi dalam penelitian ini (Cost-Sensitive NN, MLP, Extra Trees).

BAB III

METODE PENELITIAN

3.1. Deskripsi Data

Data yang saya gunakan untuk klasifikasi URL berbahaya adalah data teks yang berasal dari *website* PhishTank (untuk *alternative* saya akan menggunakan data dari Kaggle jika website PhishTank sudah tidak menyediakan data lagi)yang kemudian di-*scraping* untuk URL berbahaya. Kemudian untuk URL yang tidak berbahya diambil dari white list milik hackertarget.com yang diambil dari statistik milik amazon alexa yang berisikan URL umum seperti google.com, youtube.com.

Tabel 3.1.1 Contoh Data Mentah (phishtank.com)

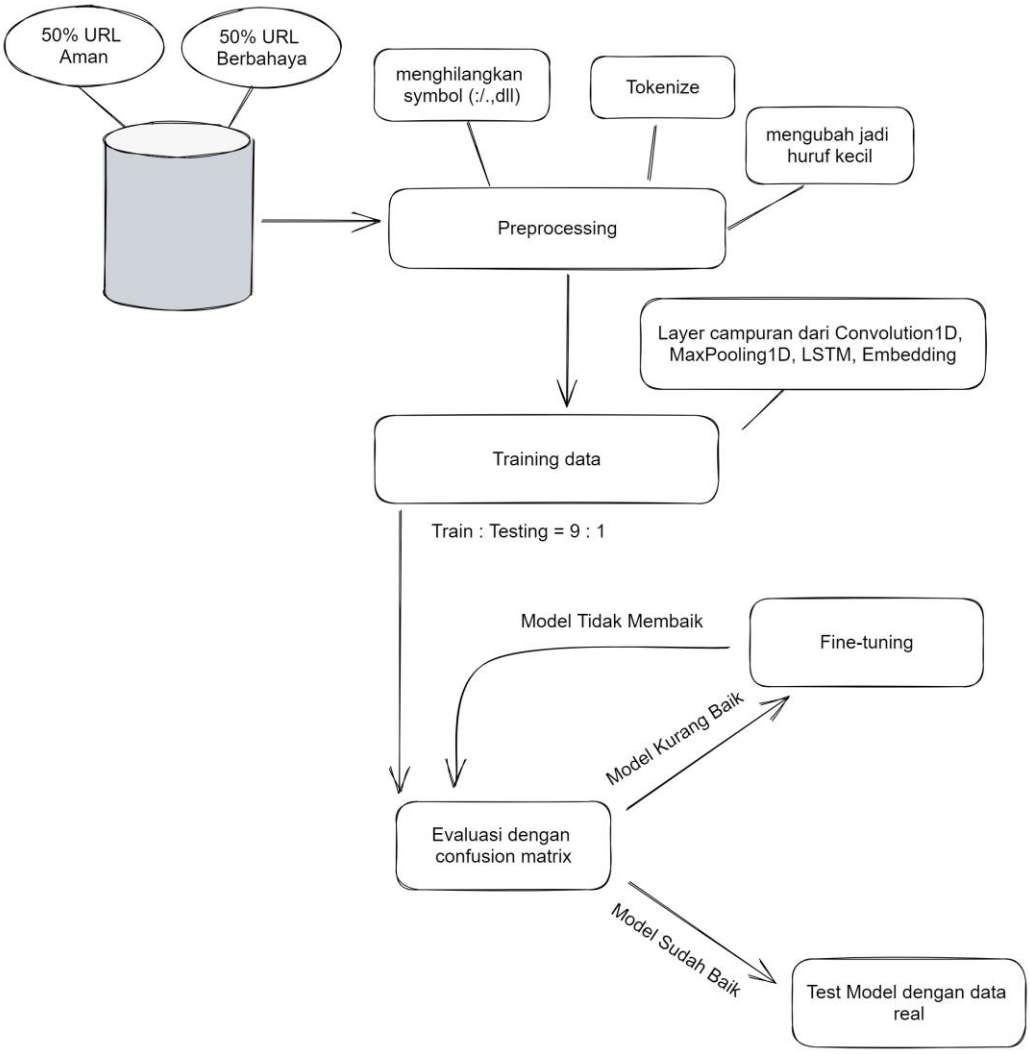
ID	Phish URL	Submitted	Valid?	Online?
8155540	https://kanagawa-u-ac-jp.com/ added on May 25th 2023 7:41 AM	by kubotaa	Unknown	ONLINE
8155539	https://mastersso-kanagawa-u-acweb-jp.com/ added on May 25th 2023 7:40 AM	by kubotaa	Unknown	ONLINE
8155538	https://pepesgifts.vip/ added on May 25th 2023 7:39 AM	by Felix0101	Unknown	ONLINE
8155537	https://paiements-antai-gouv.com/app/pages/index.php... added on May 25th 2023 7:37 AM	by Nameshield	VALID PHISH	ONLINE
8155536	https://paiements-antai-gouv.com/ added on May 25th 2023 7:37 AM	by Nameshield	VALID PHISH	ONLINE
8155535	https://luanajunaiklam.onyx-sites.io/Direct/ujalik/Manage.php... added on May 25th 2023 7:35 AM	by D3Lab	VALID PHISH	ONLINE

Data dari pishtank sudah dila beli dan di uji oleh pengelola *website*-nya dan pelapor *link* untuk *pishing* atau tidaknya, total data yang sudah didapatkan adalah 180.000 data terdiri dari 50% URL berbahaya dan 50% tidak berbahaya

3.2. Kebutuhan Perangkat Hardware dan Software

a. Spesifikasi Hardware
    i. Intel(R) Core i5-6330U @ 2.40GHz (4 CPUs) vPro
    ii. RAM 8 GB DDR4 Singel-Channel 2133 Mhz
    iii. Intel(R) HD Graphics 520
    iv. SSD
b. Spesifikasi Software
    i. Sistem Operasi Windows 10 Pro 64-bit
    ii. Browser
    iii. Kaggle Notebook (Jupyter)
    iv. Python 3.*
    v. Library Python;
        - TensorFlow dan Keras
        - Pandas
        - Sklearn
        - Numpy
        - Matplotlib

3.3. Perancangan Sistem



Gambar 3.3.1 Diagram Perancangan Sistem

Dari gambar diatas diketahui bahwa data URL *phising* diambil sebanyak 90 ribu dari masing masing kelas (50:50), kemudian data masuk dalam tahap *preprocessing* dengan diberlakukan *cleaning symbol*, mengubah huruf jadi huruf kecil dan terakhir *tokenizing*, setelah itu data masuk kedalam tahap *training* dengan cara pertama *split* data ke *train* dan *test*, kemudian masuk ke model *deep learning* yang terdiri dari *layer Embedding word*, LSTM dan CNN.

3.4. Skema Pengujian

3.4.1. Cleaning dan Normalize

Data yang didapat di *cleaning* dengan menghilangkan *symbol symbol* dan diberlakukan *preprocessing*. Contoh struktur URL:

http://subdomain.domain-name.domain-extension/path-to-resource?parameters

fitur yang digunakan:

- google => domain name
- com => top-lvl domain (domain-extension)
- ../abscz/... => path-to-resource

Sehingga menghasilkan:

Tabel 3.4.1 Data Setelah Normalisasi dan Cleaning

URL	Label
kanagawa u ac jp com	No
mastersso kanagawa u acweb jp com	No
pepesgifts vip	No
paiements antai gouv com app pages index php	Phishing
paiements antai gouv com	Phishing
luanajunaiklam onyx sites io direct ujalik Manage php	Phishing

3.4.2. Tokenize

Melakukan tahapan *tokenizing* pada data set yang sudah bersih dan dinormalisasi Menggunakan *library python*. Menambahkan OOV (*Out Of Vocabulary*) sebagai *index* 1.

<OOV>:1	Kanagawa:2	u:3	ac:4	jp:5	com:6	acweb:7	mastersso:8	vip:9

Gambar 3.4.1 sentence to token

Setelah dilakukan *tokenizing*, mengubah *token* tidak bermakna ke *sequence,* menghasilkan:

kanagawa u ac jp com	mastersso kanagawa u acweb jp com	pepesgifts vip	…
[2, 3, 4, 5, 6]	[8, 1, 3, 7, 5, 6]	[1, 9]	…

Gambar 3.4.2 token to siquence

Setelah selesai, masuk ke tahap selanjutnya yaitu melakukan *padding sequence* untuk mengubah *array* dengan beda ukuran menjadi satu ukuran yang sama. Menghasilkan:

[[0,2,3,4,5,6],

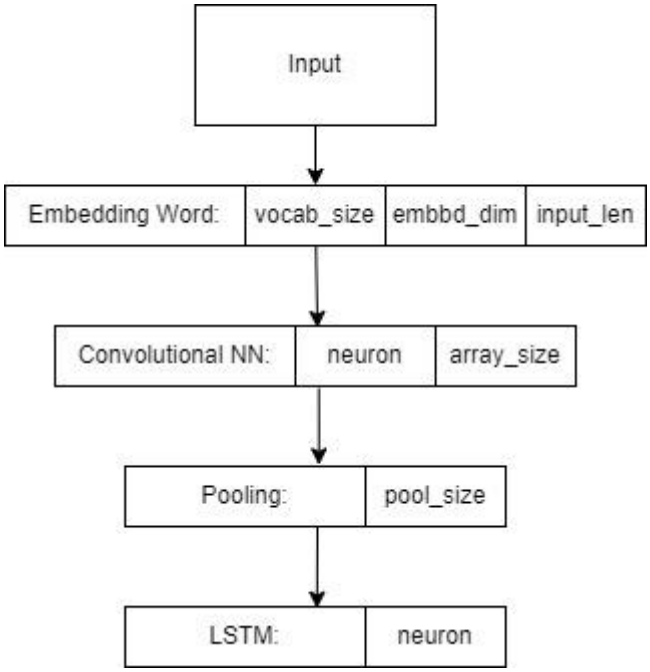
[8,1,3,7,5,6],

[0,0,0,0,1,9]]

Barulah data dapat dimasukan kedalam model.

3.4.3. Pengujian Data dalam Model

Setelah dilakukan *tokenizing*, memasukan data ke model denga *layer* wajib *Embedding Word* dan sisanya dikombinasikan dari CNN dan LSTM.



Gambar 3.4.3 Gambaran Kasar model

Modifikasi yang akan dilakukan untuk menemukan model terbaik adalah dengan mengurangi atau menambah jumlah *neuron*, *embedding dimension*, kemudian menambah atau mengurangi jumlah *layer*, dan *learning rate*. Untuk *epoch* akan diberlakukan teknik *early stopping* dimana akan menghentikan *traning* jika sudah tidak ada perkembangan akurasi atau *loss* pada beberapa epoch kedepan. Untuk neuron pada tiap layer dipergunakan *randomize* diantara [32, 64, 128, 256, 512] dan untuk learning rate dipergunakan randomize diantara [1e-5, 1e-4, 1e-3], kemudian diambil kombinasi yang menghasilkan *validation* akurasi tertinggi.

Tabel 3.4.2 Uji Coba

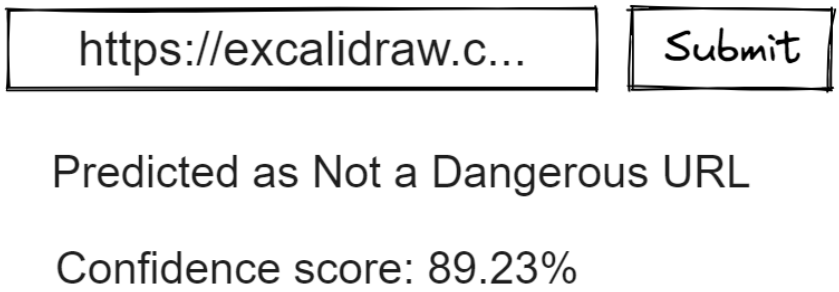
Layer	Learning rate	Neuron
LSTM, CNN	[1e-5, 1e-4, 1e-3]	[32, 64, 128, 256, 512]

LSTM	[1e-5, 1e-4, 1e-3]	[32, 64, 128, 256, 512]
CNN, Pooling	[1e-5, 1e-4, 1e-3]	[32, 64, 128, 256, 512]
LSTM, CNN, Pooling	[1e-5, 1e-4, 1e-3]	[32, 64, 128, 256, 512]

3.4.4. Evaluasi

Pada tahap evaluasi ini akan digunakan *confusion matrix* yang nantinya dijadikan acuan pengambilan keputusan untuk melakukan *re-tuning hyper parameters* dari model atau tidak. Akurasi akan dihitung berdasarkan nilai benar dalam prediksi dibagi nilai sebenarnya dan f1 *score* akan dihitung menggunakan rumus (2.1.6.3)

3.5. Desain User Interface



Gambar 3.5.1 Rancangan Desain GUI

Rancangan desain GUI yang digunakan kurang lebih seperti pada gambar 3.5.1. Untuk cara penggunaanya adalah:

1. *Paste* tautan ke *form* teks
2. Kemudian klik tombol *submit*
3. Tunggu sampai *confidence score* dan prediksi keluar

3.6. Jadwal Penelitian

No.	Bulan	Kegiatan
1	Juni	Collecting dan Processing Data
2	Juli	Model Implementation
3	Agustus	Model Evaluation
4	September	Pengerjaan Laporan Tugas Akhir
5	Oktober	Pengerjaan Laporan Tugas Akhir
6	November	Pengerjaan Laporan Tugas Akhir

DAFTAR PUSTAKA

[1] H. huan Wang, L. Yu, S. wei Tian, Y. fang Peng, and X. jun Pei, "Bidirectional LSTM Malicious webpages detection algorithm based on convolutional neural network and independent recurrent neural network," *Appl. Intell.*, vol. 49, no. 8, pp. 3016–3026, Aug. 2019, doi: 10.1007/s10489-019-01433-4.

[2] W. Wei, Q. Ke, J. Nowak, M. Korytkowski, R. Scherer, and M. Woźniak, "Accurate and fast URL phishing detector: A convolutional neural network approach," *Comput. Netw.*, vol. 178, Sep. 2020, doi: 10.1016/j.comnet.2020.107275.

[3] A. Crisan, G. Florea, L. Halasz, C. Lemnaru, and C. Oprisa, "Detecting Malicious URLs Based on Machine Learning Algorithms and Word Embeddings," in *Proceedings - 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, ICCP 2020*, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 187–193. doi: 10.1109/ICCP51029.2020.9266139.

[4] P. Wanda and H. J. Jie, "URLDeep: Continuous Prediction of Malicious URL with Dynamic Deep Learning in Social Networks," *Int. J. Netw. Secur.*, vol. 21, pp. 971–978, Nov. 2019, doi: 10.6633/IJNS.846.

[5] Z. Wang, X. Ren, S. Li, B. Wang, J. Zhang, and T. Yang, "A Malicious URL Detection Model Based on Convolutional Neural Network," *Secur. Commun. Netw.*, vol. 2021, 2021, doi: 10.1155/2021/5518528.

[6] D. Harvey M, D. Paul J, and R. Nieto T, *Internet and World Wide Web: How to Program*. Prentice Hall, 2001. [Online]. Available: www.theadmin.data.bg

[7] Giancarlo. Zaccone and Md. Rezaul. Karim, *Deep Learning with TensorFlow : Explore neural networks and build intelligent systems with Python, 2nd Edition.* Packt Publishing, 2018.

[8] T. Oliver, *Machine Learning for Absolute Beginners: A Plain English Introduction*. Scatterplot Press, 2020.

[9] G. Hirst and Y. Goldberg, *Neural Network Methods for Natural Language Processing*. Morgan & Claypool, 2017. [Online]. Available: http://store.morganclaypool.com

[10] Alexandra George, *Python Text Mining: Perform Text Processing, Word Embedding, Text Classification and Machine Translation*. BPB Publications, 2022.

[11] G. Aurélien, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019.

LAMPIRAN