Faktor Exacta

Vol. 18, No. 2, June 2025, pp. 194~200
eISSN: 2502-339X, pISSN: 1979-276X, DOI: https://doi.org/10.30998/faktorexacta.v18i2.26769

Evaluasi Efektivitas Penggunaan *FastText Embedding* dan LSTM *Networks* dalam Deteksi *Phishing* Email

Sheptianna Healtha Rukiman[1], Alam Rahmatulloh[1]
[1]Program Studi Informatika, Fakultas Teknik, Universitas Siliwangi

Article Info

Article history:

Received Dec 06, 2024
Revised Sept 09, 2025
Accepted Sept 13, 2025

Keywords:

Phishing Email Detection
Cybersecurity
FastText Embedding
Long Short-Term Memory

ABSTRACT

Phishing emails represent a significant cyber threat, necessitating advanced detection methods. This study evaluates a model combining FastText word embedding and a Long Short-Term Memory (LSTM) neural network to identify these threats. Using a public dataset from Kaggle, the model was trained on 80% of the data and tested on the remaining 20%. The methodology included data preprocessing, vectorization with FastText to capture sub-word information, and sequential pattern recognition using the LSTM architecture. Performance was evaluated using accuracy, precision, recall, and F1-Score, with the model achieving a 92% detection accuracy. Key challenges identified include class imbalance and high computational requirements. Future research could focus on model optimization and data augmentation techniques to further enhance detection performance and address these limitations.

Corresponding Author:

Alam Rahmatulloh,
Program Studi Informatika,
Fakultas Teknik,
Universitas Siliwangi,
Jl. Mugarsari, Kec. Tamansari, Kota Tasikmalaya, Jawa Barat 46196.
Email: alam@unsil.ac.id

1. PENDAHULUAN

Phishing merupakan salah satu ancaman siber yang terus bertransformasi dan mengintensifkan dampaknya dalam beberapa tahun terakhir. *Phishing* email, sebagai salah satu bentuk serangan rekayasa sosial, dirancang untuk memanipulasi pengguna agar mengungkapkan informasi sensitif, seperti kata sandi dan data finansial. Teknik rekayasa sosial ini sering kali memanfaatkan *phishing* email yang menyamar sebagai sumber tepercaya untuk mengekstrak data pribadi, seperti kata sandi dan informasi rekening bank, secara ilegal [1]. Email *phishing* sering kali dioptimalkan agar terlihat asli, sehingga memprovokasi penerima untuk mengklik tautan berbahaya atau membocorkan informasi sensitif [2]. Dengan mengonstruksi situs web palsu yang terlihat menyerupai situs asli, penyerang berusaha menyesatkan pengguna untuk memasukkan informasi sensitif dengan alasan keamanan yang keliru [3]. Teknik *phishing* yang terus berkembang dan semakin canggih menantang pengembangan sistem deteksi yang lebih efektif untuk mengantisipasi ancaman yang terus meningkat.

Berbagai metode telah diimplementasikan untuk mengatasi *phishing*, termasuk teknik *rule-based* dan analisis URL yang didesain untuk mengidentifikasi karakteristik *phishing* yang sering digunakan. Analisis URL memanfaatkan pemeriksaan fitur seperti usia domain, panjang URL, dan keberadaan kata kunci tertentu untuk membedakan situs yang sah dari situs *phishing* [4]. Selain itu, teknik *machine learning* dan *deep learning* menawarkan potensi besar dalam mengoptimalkan akurasi deteksi *phishing* dengan mengekstrak fitur secara otomatis dari data yang besar. Teknik *big data* memfasilitasi analisis pola lalu lintas yang masuk, sehingga memperkuat kemampuan mendeteksi upaya *phishing* yang tersembunyi [5]. Pendekatan yang menonjol dalam deteksi *phishing* adalah penggunaan teknik *Natural Language Processing* (NLP) dengan *word embedding* untuk merepresentasikan teks, seperti Word2Vec dan FastText, serta jaringan saraf *Long Short-Term Memory* (LSTM) yang diakui unggul dalam mengolah *sequence data*. Teknik seperti Word2Vec dan FastText mengubah data tekstual menjadi vektor numerik, menggambarkan hubungan semantik. Representasi ini

memungkinkan identifikasi pola yang terkait dengan *phishing* dalam URL dan email, sehingga meningkatkan efektivitas sistem deteksi [6].

Meskipun metode-metode ini menunjukkan hasil yang cukup baik, terdapat kelebihan dan kekurangan yang perlu dievaluasi. Misalnya, Word2Vec mampu menghasilkan representasi kata yang akurat, tetapi cenderung mengabaikan informasi morfologi dan variasinya, sehingga membatasi keefektifannya dalam situasi seperti *phishing* email [7]. Di sisi lain, FastText mengadopsi unit subkata, seperti karakter *n-gram*, yang memungkinkan model untuk menggali pengetahuan semantik dan sintaksis dengan lebih efektif dibandingkan model berbasis kata tradisional [8]. Namun, FastText tetap dihadapkan pada tantangan dalam mengolah data yang sangat tidak terstruktur, seperti email *phishing*. Untuk data berurutan dengan pola teks yang kompleks, jaringan LSTM mampu mengidentifikasi pola tersebut dengan baik. Namun, untuk mencapai hasil yang optimal, LSTM memerlukan jumlah data pelatihan yang besar, pemrosesan data yang cermat, serta pengoptimalan model yang tepat, yang semuanya membebankan waktu dan sumber daya komputasi yang signifikan [9].

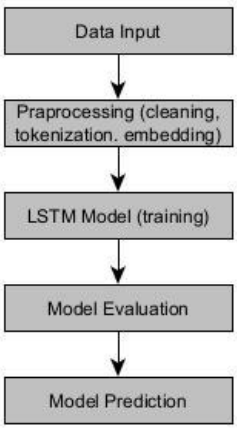
Salah satu tantangan utama dalam deteksi *phishing* email menggunakan metode ini adalah bahwa kombinasi FastText dan LSTM belum mencapai optimalisasi penuh dalam hal kecepatan dan akurasi, terutama ketika dihadapkan pada dataset yang besar dan beragam. Model ini kesulitan mengelola dan mengintegrasikan proses teks yang panjang secara efisien [10]. Selain itu, ketidakseimbangan kelas yang signifikan dalam dataset *phishing* email sering kali menghambat performa deteksi, karena kelas mayoritas dominan dibandingkan dengan kelas minoritas, yang mengarah pada bias dalam prediksi model [11].

Dalam sebuah studi, model FastText diimplementasikan untuk menghasilkan vektor fitur dari konten email, yang kemudian diproses oleh model *Bidirectional* LSTM (BiLSTM), mencapai tingkat akurasi tinggi sebesar 99,12% pada dataset yang tidak seimbang [12]. Selain itu, kerangka kerja lain memanfaatkan FastText untuk menganalisis komponen URL, mengoptimalkan efektivitas deteksi URL *phishing* dengan akurasi yang tinggi dan kebutuhan komputasi yang rendah [13]. Model HLSTMCNN, yang mengintegrasikan LSTM dengan lapisan CNN, terbukti efektif dalam mengidentifikasi pola-pola yang merepresentasikan upaya *phishing*, sehingga memberikan kinerja yang lebih baik dibandingkan metode deteksi *phishing* tradisional [14]. Analisis komparatif menegaskan bahwa model LSTM secara konsisten melampaui arsitektur lain, dengan tingkat akurasi yang mencapai lebih dari 99% [15].

Sebagai solusi, penelitian ini berfokus pada mengevaluasi efektivitas penggunaan embedding FastText yang dikombinasikan dengan jaringan LSTM yang dioptimalkan untuk meningkatkan akurasi deteksi phishing email. Optimalisasi ini melibatkan penyesuaian pada struktur jaringan LSTM atau mengadopsi teknik data augmentation guna mengatasi ketidakseimbangan kelas dalam dataset phishing. Penelitian ini bertujuan untuk memberikan evaluasi menyeluruh terhadap keefektifan integrasi embedding FastText dan jaringan LSTM dalam mengidentifikasi phishing email, sekaligus mengusulkan perbaikan-perbaikan yang diperlukan agar model dapat berfungsi secara lebih efisien dan akurat dalam skenario dunia nyata.

2. METODE

Penelitian ini menggunakan pendekatan kuantitatif dengan memanfaatkan data sekunder berupa *Phishing Emails Dataset* dari Kaggle. Data dibagi dengan rasio 80% untuk pelatihan (*training*) dan 20% untuk pengujian (*testing*). Perancangan penelitian dirancang untuk mencakup serangkaian langkah yang melibatkan seluruh proses, dari tahap awal hingga akhir. Berikut diagram alur penelitian yang menggambarkan tahapan-tahapan tersebut disajikan untuk memberikan visualisasi yang lebih jelas dan terstruktur.



Gambar 1. Diagram Alur Penelitian

Gambar 1 mengilustrasikan diagram alur penelitian penggunaan *embedding* FastText dan LSTM untuk deteksi *phishing* email. Proses dimulai dengan tahap *Data Input*, yaitu mengimpor data mentah berupa teks ke dalam sistem. Selanjutnya, dilaksanakan tahap *Preprocessing*, yang melibatkan pembersihan data dari elemen tidak relevan (*cleaning*), pemecahan teks menjadi token (*tokenization*), dan mengonversi token menjadi representasi numerik (*embedding*) agar dapat diolah oleh model. Setelah proses praproses selesai, data yang telah diproses digunakan untuk melatih model LSTM, di mana model ini memanfaatkan kemampuannya dalam mengidentifikasi pola pada data teks berurutan dan mempertahankan konteks. Setelah pelatihan, model dievaluasi pada tahap *Model Evaluation* dengan menggunakan metrik seperti akurasi atau presisi untuk mengukur performanya. Terakhir, model yang telah dilatih diimplementasikan pada tahap *Model Prediction* untuk mengklasifikasikan data baru, menghasilkan keluaran sesuai tujuan penelitian, seperti klasifikasi teks *phishing*.

2.1 *Data Input*

Proses ini dimulai dengan mengimpor data berupa email *phishing* dan *non-phishing*, yang digunakan sebagai fondasi untuk pelatihan model. Kualitas dan jumlah data pada tahap ini memegang peranan krusial, karena secara langsung memengaruhi performa model dalam mengidentifikasi dan mengklasifikasikan email dengan akurasi yang tinggi.

2.2 *Preprocessing (cleaning, tokenization, embedding)*

Pada tahap ini, data "*Phishing* Email" melalui proses prapemrosesan yang mencakup *cleaning* (normalisasi teks, penghapusan karakter khusus dan konversi teks ke huruf kecil), *tokenization* (konversi teks menjadi token numerik), dan *embedding* (menggunakan FastText 300 dimensi untuk merepresentasikan kata-kata dalam bentuk vektor numerik). *Tokenization* atau tokenisasi, yang mengubah data *input* menjadi token, merupakan langkah penting dalam prapemrosesan pembelajaran mendalam [16]. *Cleaning* atau pembersihan data bertujuan untuk menghilangkan *noise*, seperti tanda baca, kata henti, dan karakter yang tidak relevan, yang dapat memengaruhi hasil analisis secara negatif [17]. Sementara itu, *embedding* digunakan untuk menyimpan karakteristik data dalam bentuk vektor numerik, yang memungkinkan pemodelan hubungan semantik antar kata [18].

2.3 LSTM Model (*training*)

Pada tahap LSTM Model *Training*, model diuji dengan beberapa konfigurasi (jumlah unit LSTM, *dropout*, *epoch*) untuk melihat pengaruhnya terhadap akurasi. Data yang telah melalui proses prapemrosesan dimanfaatkan untuk melatih model LSTM. Model ini dirancang untuk mengidentifikasi urutan dan hubungan temporal dalam data teks, yang sangat relevan untuk mendeteksi pola dalam email *phishing*. Data yang telah diproses dibagi menjadi dua bagian: *training set* dan *testing set*. *Training set* digunakan untuk membimbing model dalam mempelajari pola-pola yang terdapat dalam email *phishing* dan *non-phishing*. Sementara itu, *testing set* berfungsi untuk mengevaluasi performa model pada data yang belum pernah dihadapi, guna mengukur kemampuannya dalam menggeneralisasi pola di luar data pelatihan.

2.4 *Model Evaluation*

Setelah proses pelatihan selesai, model dievaluasi menggunakan metrik performa *accuracy*, *precision*, *recall*, dan *F1-score* untuk menilai seberapa baik model dalam mendeteksi email *phishing*. Hasil dibandingkan untuk menunjukkan apakah integrasi FastText–LSTM lebih efektif daripada *baseline* (LSTM tanpa *embedding*). *Accuracy* mengukur seberapa tepat model dalam mengklasifikasikan email secara keseluruhan, sedangkan *precision* mengidentifikasi tingkat ketepatan model dalam mendeteksi email *phishing* yang benar dari seluruh prediksi *phishing*. *Recall* mengevaluasi sejauh mana model berhasil menemukan semua email *phishing* yang benar-benar ada dalam data. Sementara itu, *F1-score*, yang merupakan gabungan dari *precision* dan *recall*, menyediakan gambaran keseimbangan antara kedua metrik tersebut. Evaluasi ini memberikan gambaran menyeluruh tentang kemampuan model dalam membedakan email *phishing* dari *non-phishing*, sekaligus mengidentifikasi bagaimana model mengatasi kesalahan prediksi yang mungkin terjadi.

2.5 *Model Prediction*

Pada tahap *Model Prediction*, model yang telah dilatih dan dievaluasi diimplementasikan untuk memprediksi klasifikasi email baru yang belum pernah dihadapi sebelumnya. Model ini menganalisis teks dari email baru dan, berdasarkan pola serta karakteristik yang telah dipelajari selama pelatihan, menentukan apakah email tersebut merupakan *phishing* atau bukan. Prediksi ini merepresentasikan aplikasi nyata dari model dalam mendeteksi ancaman *phishing* secara otomatis. Dengan kemampuan ini, model dapat diterapkan dalam sistem keamanan email untuk membantu melindungi pengguna dari

membuka atau merespons email yang berbahaya. Hasil prediksi ini berperan krusial dalam memitigasi risiko serangan *phishing*, sehingga meningkatkan keamanan di dunia nyata.

3. HASIL DAN PEMBAHASAN

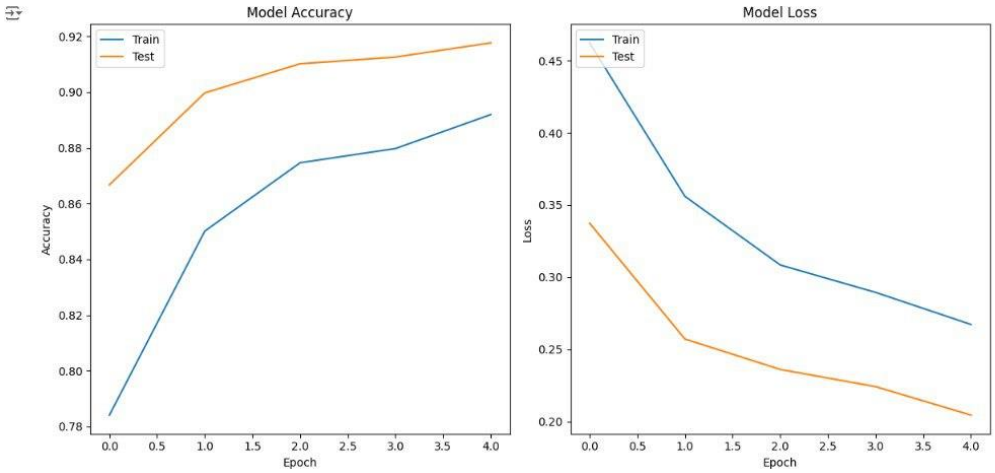
Berdasarkan dataset "*Phishing Emails Dataset*" yang diperoleh dari Kaggle, penelitian ini melakukan deteksi email phishing melalui serangkaian tahap pra-pemrosesan data. Tahap pertama adalah pembersihan (*cleaning*), yang melibatkan penghapusan nilai *non-string* seperti NaN, penghilangan karakter khusus menggunakan ekspresi reguler, normalisasi spasi berlebih, dan konversi seluruh teks menjadi huruf kecil untuk memastikan konsistensi data. Tahap berikutnya adalah tokenisasi, yaitu mengonversi kata-kata dalam teks email menjadi representasi numerik dengan membatasi *vocabulary* sebanyak 5000 kata paling umum dan menambahkan token <OOV> untuk menangani kata-kata di luar *vocabulary*. Tahap terakhir adalah *embedding* menggunakan model FastText pra-latih bahasa Inggris untuk menghasilkan *embedding matrix* berdimensi 300 yang merepresentasikan setiap kata dalam dataset, sehingga data siap diproses oleh model LSTM.

Proses pelatihan model menunjukkan perkembangan yang positif terhadap kemampuan belajar. Berikut adalah ringkasan performa model pada *epoch* pertama dan akhir yang menggambarkan peningkatan akurasi dan konvergensi model yang disajikan dalam Tabel 1.

Tabel 1. Hasil Pelatihan (*epoch*)

Epoch	Akurasi *Training*	*Loss Training*	Akurasi Validasi	*Loss* Validasi
1	0.7841	0.4622	0.8668	0.3374
2	0.8501	0.3560	0.8997	0.2572
3	0.8747	0.3085	0.9102	0.2362
4	0.8798	0.2895	0.9126	0.2242
5	0.8920	0.2673	0.9177	0.2045

Tabel 1 menunjukkan performa model pada *epoch* pertama dan terakhir (*epoch* ke-5) selama proses pelatihan. Terjadi peningkatan yang signifikan pada Akurasi *Training* (dari 0.7841 menjadi 0.8920) dan Akurasi Validasi (dari 0.8668 menjadi 0.9177). Yang sangat penting untuk dicatat adalah akurasi validasi secara konsisten lebih tinggi daripada akurasi *training*, khususnya pada *epoch* pertama. Hal ini merupakan indikasi yang baik yang menunjukkan model tidak mengalami *overfitting* pada data *training* dan mampu menggeneralisasi dengan sangat baik pada data yang belum pernah dilihat.



Gambar 2. Plot Akurasi Model

Berdasarkan Gambar 3, plot akurasi dan loss menunjukkan performa model yang sangat baik. Akurasi pelatihan dan pengujian sama-sama meningkat seiring *epoch*, dengan selisih yang kecil di antara keduanya, mengindikasikan tidak adanya *overfitting*. Akurasi pengujian mencapai titik jenuh sekitar *epoch* ke-2.5, menandakan kemampuan generalisasi yang baik. Sementara itu, nilai *loss* untuk kedua data juga terus menurun, dengan *loss* pengujian yang menurun cepat dan kemudian stabil, memperkuat bahwa model belajar secara efektif.

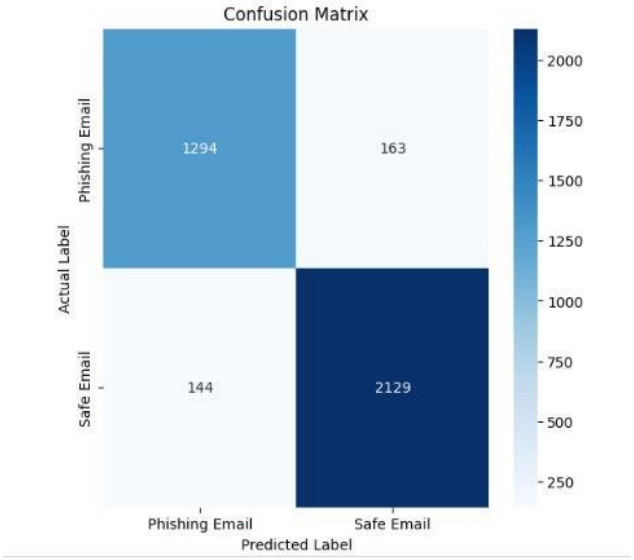
Evaluasi mendalam terhadap performa model menggunakan metrik *precision*, *recall*, dan *F1-score* memberikan gambaran yang lebih detail dibandingkan akurasi secara keseluruhan. Laporan klasifikasi berikut mengukur efektivitas model dalam mengidentifikasi masing-masing kelas, yaitu "*Phishing Email*" dan "*Safe Email*".

Tabel 2. *Classification Report*

Classification Report:	***precision***	***recall***	***F1-Score***	***support***
Phishing Email	0.90	0.89	0.89	1457
Safe Email	0.93	0.94	0.93	2273
accuracy			0.92	3730
marco avg	0.91	0.91	0.91	3730
weighted avg	0.92	0.92	0.92	3730

Berdasarkan Tabel 2, model ini menunjukkan performa yang sangat kuat dengan akurasi keseluruhan sebesar 92% dari 3730 email yang diuji. Untuk kelas "*Phishing Email*", model mencapai *precision* 0.90 yang berarti 90% prediksi *phishing* adalah benar, dan *recall* 0.89 yang menunjukkan kemampuannya mendeteksi 89% dari semua email *phishing* yang ada, dengan *F1-Score* 0.89. Sementara untuk kelas "*Safe Email*", performanya bahkan lebih unggul dengan *precision* 0.93, *recall* 0.94, dan *F1-Score* 0.93. Model ini sangat seimbang (*balanced*) tanpa disparitas performa yang signifikan antara kedua kelas, meskipun kelas "*Safe Email*" sebagai kelas mayoritas menunjukkan angka yang sedikit lebih tinggi.

Untuk menganalisis jenis kesalahan klasifikasi yang dilakukan oleh model, *Confusion Matrix* dapat dilihat pada Gambar 3 berikut. Matriks ini menguraikan jumlah prediksi yang benar berupa *True Positive* dan *True Negative*, serta kesalahan prediksi berupa *False Negative* dan *False Positive*.



Gambar 3. *Confusion Matrix*

Confusion Matrix mengonfirmasi performa model yang telah diuraikan dalam laporan klasifikasi. Matriks ini menunjukkan bahwa model berhasil mengidentifikasi dengan benar 1294 *Phishing Email* (*True Positive*/TP) dan 2129 *Safe Email* (*True Negative*/TN). Di sisi-lain, model melakukan kesalahan dengan mengklasifikasikan 163 *Phishing Email* sebagai *Safe Email* (*False Negative*/FN), yang merupakan jenis kesalahan paling kritis karena mengakibatkan ancaman yang lolos dari deteksi. Selain itu, terdapat 144 *Safe Email* yang salah diklasifikasikan sebagai *Phishing Email* (*False Positive*/FP), yang dapat mengganggu pengguna karena email *legitimate* mungkin tersaring ke folder *spam*. Secara keseluruhan, matriks ini memperkuat kesimpulan bahwa model memiliki akurasi yang tinggi dan seimbang, meskipun masih terdapat sedikit *trade-off* antara *False Negative* dan *False Positive*.

Hasil evaluasi menunjukkan model mencapai akurasi total 92% dengan keseimbangan yang baik antara *precision* dan *recall*. Namun, teridentifikasi kesalahan klasifikasi yang lebih sering terjadi pada kelas minoritas (*phishing*), mengindikasikan adanya pengaruh *imbalance* data dalam performa model.

4. KESIMPULAN

Penelitian ini berhasil membuktikan efektivitas integrasi *embedding* FastText dengan arsitektur LSTM dalam mendeteksi email *phishing*, dengan mencapai akurasi sebesar 92%. Model tidak hanya akurat tetapi juga konsisten dalam membedakan antara email *phishing* dan *non-phishing*, yang ditunjukkan oleh nilai *precision*, *recall*, dan *F1-score* yang tinggi dan seimbang untuk kedua kelas. Keunggulan utama dari pendekatan ini adalah kemampuan FastText dalam menangani kata-kata di luar *vocabulary/Out of Vocabullary* (OOV) dan menangkap informasi morfologis, yang kemudian diperkuat oleh LSTM dalam memodelkan ketergantungan *sequential* dalam teks email. Namun, implementasi ini masih menghadapi beberapa tantangan, seperti ketidakseimbangan jumlah sampel antar kelas serta kebutuhan sumber daya komputasi yang relatif tinggi untuk proses pelatihan.

Sun, Yuwei, et al. (2022), dalam penelitiannya yang berjudul "Federated Phish Bowl: LSTM-Based Decentralized Phishing Email Detection", mengusulkan metode *Federated Phish Bowl* (FPB) berbasis jaringan LSTM untuk deteksi email *phishing*. Pendekatan ini memungkinkan akuisisi pengetahuan dari kumpulan data email yang beragam sekaligus menjaga privasi pengguna melalui mekanisme federasi. Hasil penelitian menunjukkan bahwa FPB mencapai akurasi deteksi sebesar 83%, dengan tetap melindungi privasi komunikasi email yang sensitif [19]. Untuk meningkatkan tingkat akurasi, dapat dilakukan penggabungan beberapa model *machine learning* dan *deep learning* serta memanfaatkan teknik *Natural Language Processing* (NLP). Kombinasi berbagai metode ini dapat menghasilkan performa deteksi yang lebih optimal, seperti pada penelitian ini yang menunjukkan akurasi mencapai 92%. Hal ini menunjukkan bahwa pengembangan metode berbasis LSTM dan NLP memiliki potensi besar untuk meningkatkan akurasi deteksi *phishing* email.

Berdasarkan temuan dan keterbatasan dalam penelitian ini, beberapa saran dapat dipertimbangkan untuk pengembangan di masa depan. Pertama, penggunaan lebih dari satu dataset dari sumber yang beragam sangat dianjurkan untuk menguji kemampuan generalisasi model secara lebih komprehensif dan mengurangi bias. Kedua, untuk mengatasi masalah ketidakseimbangan kelas (*imbalance data*), teknik *data augmentation* pada teks atau metode *sampling* seperti SMOTE dapat diterapkan. Ketiga, mengeksplorasi dan membandingkan arsitektur model lain seperti GRU, Transformer, atau *hybrid* CNN-LSTM dapat dilakukan untuk mengetahui apakah terdapat arsitektur yang lebih efisien dan akurat. Terakhir, penelitian lebih lanjut dapat diarahkan pada pengembangan sistem deteksi *real-time* yang dapat diintegrasikan langsung ke dalam layanan email, sehingga memberikan nilai praktis dan *immediat impact* dalam melindungi pengguna dari ancaman *phishing*.

DAFTAR PUSTAKA

[1] M. Sharma, Sunil; Sharma, Rahul; Sharma, "Phishing Email Detection in Cyber Security," 2024.

[2] X. Xia and G. Mogos, "Kalinux and Gophish analysis of phishing emails," *J. Contents Comput.*, vol. 5, no. 2, pp. 777–794, 2023.

[3] M. Sanap, "PHISHING ATTACKS DETECTION USING MACHINE LEARNING APPROACH," 2024, pp. 137–140. doi: 10.58532/V3BAAI6P8CH1.

[4] D. Urlamma, M. Supriya, D. Lavanya, and A. Hari Priya, "Detection Of Phishing Websites Using Gradient Boosting Classifier Based On URL," *Iarjset*, vol. 11, no. 3, pp. 116–121, 2024, doi: 10.17148/iarjset.2024.11318.

[5] B. B. Gupta, A. Gaurav, J. Wu, V. Arya, and K. T. Chui, "Deep Learning and Big Data Integration with Cuckoo Search Optimization for Robust Phishing Attack Detection," in *ICC 2024-IEEE International Conference on Communications*, IEEE, 2024, pp. 1322–1327.

[6] E. Blancaflor, A. H. Calpo, S. J. Cebrian, and F. Siquioco, "A Comprehensive Review of Neural Network-Based Approaches for Predicting Phishing Websites and URLs," in *2024 5th International Conference on Industrial Engineering and Artificial Intelligence (IEAI)*, IEEE, 2024, pp. 96–101.

[7] F. A. O. Santos, H. T. Macedo, T. D. Bispo, and C. Zanchettin, "Morphological skip-gram: Replacing fasttext characters n-gram with morphological knowledge," *Intel. Artif.*, vol. 24, no. 67, pp. 1–17, 2021, doi: 10.4114/intartif.vol24iss67pp1-17.

[8] M. R. Vivek and P. Chandran, "Analysis of Subword based Word Representations Case Study: Fasttext Malayalam," in *2022 IEEE 19th India Council International Conference (INDICON)*, IEEE, 2022, pp. 1–6.

[9] R. Siringoringo, J. Jamaluddin, R. Perangin-angin, E. J. G. Harianja, G. Lumbantoruan, and E. N. Purba, "Model Bidirectional Lstm Untuk Pemrosesan Sekuensial Data Teks Spam," *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 7, no. 2, pp. 265–271, 2023, doi: 10.46880/jmika.vol7no2.pp265-271.

[10] J. Stremmel, B. L. Hill, J. Hertzberg, J. Murillo, L. Allotey, and E. Halperin, "Extend and explain: Interpreting very long language models," *Proc. Mach. Learn. Res.*, vol. 193, no. Ml, pp. 218–258, 2022.

[11] J. Srivastava and A. Sharan, "Malicious website detection using BorderlineSMOTE2NCR sampling and cost-sensitive ensemble learning," in *International Conference on Data Science and Big Data Analysis*,

Springer, 2023, pp. 665–675.
[12] R. Wolert and M. Rawski, "Email Phishing Detection with BLSTM and Word Embeddings," *Int. J. Electron. Telecommun.*, vol. 69, no. 3, pp. 485–491, 2023, doi: 10.24425/ijet.2023.146496.
[13] K. Mangalam and B. Subba, "PhishDetect: A BiLSTM based phishing URL detection framework using FastText embeddings," in *2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS)*, IEEE, 2024, pp. 637–641.
[14] N. A. Nijhum, Q. Li, and T. Yang, "HLSTMCNN: A Hybrid Deep Learning Model to Detect Phishing Email," in *2023 3rd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, IEEE, 2023, pp. 61–66.
[15] C. K. Truong, P. Hao Do, and T. Duc Le, "A comparative analysis of email phishing detection methods: a deep learning perspective," IET, 2023.
[16] R. Friedman, "Tokenization in the Theory of Knowledge," *Encyclopedia*, vol. 3, no. 1, pp. 380–386, 2023, doi: 10.3390/encyclopedia3010024.
[17] S. Kumari, "Text mining and pre-processing methods for social media data extraction and processing," in *Handbook of research on opinion mining and text analytics on literary works and social media*, IGI Global, 2022, pp. 22–53.
[18] B. Dai, X. Shen, and J. Wang, "Embedding learning," *J. Am. Stat. Assoc.*, vol. 117, no. 537, pp. 307–319, 2022.
[19] Y. Sun, N. Chong, and H. Ochiai, "Federated Phish Bowl: LSTM-Based Decentralized Phishing Email Detection," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 2022-October, pp. 20–25, 2022, doi: 10.1109/SMC53654.2022.9945584.