Perbandingan Vektorisasi Deteksi Spam Email Menggunakan Bag of Word, TF IDF, dan Word2Vec pada Multinomial Naïve Bayes

Rony Arifiandy [1*], Hasanul Fahmi [2]

[1,2] 1S2IT - Informatics Study Program, President University, Indonesia
Email: * ronyarifiandy@gmail.com

Abstrak. Penelitian ini akan mencoba menemukan teknik preprocessing teks yang lebih baik untuk mendukung algoritma Multinomial Naïve Bayes dengan 3 kelas (ham, phishing dan fraud), diharapkan hasil dari penelitian ini dapat membantu pengguna dalam mengklasifikasikan email spam dengan lebih akurat. Untuk dapat melakukan hal tersebut, dalam preprocessing data kita perlu melakukan vektorisasi body email agar machine learning dapat melakukan perhitungan. Vektorisasi memungkinkan mesin memahami konten tekstual dengan mengubahnya menjadi representasi numerik yang bermakna. Efektivitas berbagai metode vektorisasi teks, yaitu bag of word, TF-IDF dan word2vec diselidiki untuk deteksi spam email menggunakan Multinomial Naïve Bayes. Makalah ini menyajikan analisis komparatif berbagai metode vektorisasi pada kumpulan data email spam. Makalah ini akan memberikan vektorisasi terbaik dengan Multinomial Naive Bayes.

Kata kunci: *spam email, vectorization, bag of word, TFIDF, word2vec, Naïve Bayes*

1 Pendahuluan

Email telah menjadi sangat populer di kalangan masyarakat saat ini. Faktanya, ini adalah alat komunikasi termurah, populer dan tercepat saat ini. Email juga telah menjadi media komunikasi resmi di dunia bisnis. Kepopuleran email juga dimanfaatkan oleh oknum-oknum yang tidak bertanggung jawab sebagai media pengiriman berita bohong, sebagai media penipuan dan lain sebagainya. Kami menyebut email semacam ini sebagai email spam. Ada email spam yang berbahaya dan tidak berbahaya. Kami akan fokus pada email spam berbahaya, ada 2 jenis email spam berbahaya. Yang pertama adalah email phishing. Phishing adalah istilah yang digunakan untuk mendefinisikan praktik penipuan di mana pelaku spam mencoba mengelabui korbannya. Biasanya para spammer ini berpura-pura menjadi brand terkenal atau berpura-pura menjadi orang terkenal. Tujuan email phishing adalah untuk membuat korban bersedia memberikan informasi penting atau meminta pengguna membuka

Diterima 23 Maret 2024, Direvisi 25 Mei 2024 Diterima untuk publikasi 31 Mei 2024

lampiran file atau mengklik link tanpa mencurigai apa pun. Tanpa disadari pengguna, mereka telah membuka file yang berisi malware atau link yang terhubung ke pusat malware. Dan yang kedua adalah email *fraud*. Email *fraud* merupakan salah satu jenis email spam yang juga patut Anda waspadai. Jika email phishing bertujuan untuk mendapatkan informasi atau data rahasia dari pemilik email, maka email *fraud* dilakukan untuk melakukan penipuan yang berujung pada pemerasan. Pengirim email mengaku sebagai atau atas nama pihak tertentu lalu meminta sejumlah uang untuk ditransfer ke rekeningnya[1].

Email terdiri dari beberapa bagian; bagian yang akan kita periksa adalah badan email. Badan email adalah bagian email yang berisi kalimat atau kata. Isi body email ini akan diperiksa dan ditentukan apakah termasuk dalam kategori email phishing atau tidak. Oleh karena itu diperlukan teknik agar kata atau kalimat dapat diolah. Diperlukan suatu teknik vektorisasi agar kata atau kalimat tersebut dapat diproses dengan algoritma Naive Bayes. Vektorisasi adalah proses mengubah data non numerik menjadi numerik agar data tersebut dapat diproses oleh komputer. Ada banyak teknik yang dapat digunakan untuk melakukan vektorisasi ini. Seperti Glove, FastText [2], Bag of word, TF-IDF dan Word2Vec [2]. Hasil dari proses ini akan digunakan untuk mengklasifikasikan kata. Klasifikasi kata merupakan suatu metode yang digunakan untuk mengelompokkan kata berdasarkan kategori tertentu. Penelitian [3] menyebutkan bahwa Naïve Bayes Classifier memiliki tingkat akurasi yang lebih baik dibandingkan model classifier lainnya. Penelitian [4] menyebutkan bahwa bare NB mempunyai akurasi bobot sebesar 99,475%. Penelitian lain [5] menyatakan bahwa NB menggunakan ekstraksi fitur Pemilihan fitur berbasis korelasi (CFS) menemukan akurasi sebesar 91,13%. Dan penelitian [6] menyatakan NB mempunyai akurasi 97,5% bila menggunakan TF-IDF.

Email spam yang berbahaya dapat menimbulkan kerugian yang sangat besar, seperti kehilangan uang, kehilangan data bahkan hilangnya kepercayaan karena dianggap tidak hati-hati. Sangat penting untuk membantu pengguna email mengidentifikasi email spam berbahaya.

Machine learning memiliki banyak algoritma yang dapat digunakan, salah satunya adalah algoritma Naïve Bayes yang digunakan untuk klasifikasi. Sebelum proses machine learning menggunakan algoritma ini, perlu adanya proses preprocessing data (teks) agar dapat diproses oleh algoritma Naïve Bayes ini. Proses pemrosesan awal ini sangat penting agar hasil yang diberikan oleh algoritma Naïve Bayes

sangat baik. Sangat penting untuk mengetahui teknik terbaik yang dapat mendukung algoritma Naïve Bayes. Penelitian ini akan mencoba mencari teknik preprocessing teks yang lebih baik untuk mendukung algoritma Multinomial Naïve Bayes (MNB) dengan 3 kelas (ham, phishing dan penipuan) untuk mengklasifikasikan jenis email, diharapkan dapat membantu pengguna dalam mengklasifikasikan email spam dengan lebih akurat. Penelitian [7] menyebutkan bahwa MNB tanpa proses vektorisasi mempunyai akurasi 93%, presisi 100%, recall 74% dan f1-score 85%.

Klasifikasi email spam ini hanya mengandalkan teks di badan email. Terkadang ada email body yang serupa namun bukan spam, hal ini bisa diketahui dari pengirim yang menjadi partner komunikasi pengguna. Jadi selanjutnya klasifikasi harus ditambahkan mitra komunikasi pengguna.

2 Landasan Teori

2.1 Vectorizing

Klasifikasi email spam adalah dengan membaca body email yang berupa teks dan untuk menghitung teks tersebut diperlukan proses vektorisasi. Di bawah ini adalah pengenalan singkat tentang proses vektorisasi teks yang ada.

2.1.1 Bag of Word (BoW)

Para peneliti dalam [8] menyebutkan bahwa BoW merupakan model multiguna yang dapat digunakan sebagai algoritma pemilihan fitur, dan klasifikasi dokumen dan gambar. Dalam klasifikasi dokumen, BoW adalah vektor jumlah kemunculan kata, yang disebut juga histogram dokumen tersebut. Beberapa kata yang tidak informatif, seperti; a, an, the, dan, dll. akan dihapus dari kamus setelah menghitung semua kata dari kamus yang muncul di dokumen.

Misalkan *V* adalah kosakata kata-kata unik di seluruh korpus. Misalkan *n* adalah jumlah kata unik dalam kosa kata. Misalkan d_i adalah kata ke-*i* di *V*. Misalkan f_{di} adalah frekuensi data d_i pada dokumen D. Bag of Words dimisalkan *BoW(D)* dari dokumen D adalah vektor dengan panjang *n*, di mana setiap elemen mewakili frekuensi kata yang terkait dalam kosakata:

$$BoW = (f_{d1}, f_{d2}, \dots, f_{dn}) \quad (1)$$

Representasi vektor kata dengan metode BoW menggunakan scikit learn (sklearn) dengan Python yang ditunjukkan pada Gambar 1.

```
doc1 = 'Game of Thrones is an amazing tv series!'
doc2 = 'Game of Thrones is the best tv series!'
doc3 = 'Game of Thrones is so great'
```

	amazing	best	game	great	series	thrones	tv
0	1	0	1	0	1	1	1
1	0	1	1	0	1	1	1
2	0	0	1	1	0	1	0

Gambar 1. Hasil BoW untuk 3 dokumen menggunakan sklearn di python

2.1.2 TF-IDF

Term Frekuensi-Invers Dokumen Frekuensi (TF-IDF) adalah metode yang paling umum digunakan dalam NLP untuk mengubah dokumen teks menjadi representasi matriks vektor. Representasi TF-IDF mencerminkan penonjolan suatu kata dalam kumpulan dokumen terhadap dokumen individual [9]. Pada penelitian ini menggunakan sampel yang sama dengan BoW untuk merepresentasikan model TF-IDF sehingga hasil BoW adalah hasil TF.

	amazing	best	game	great	series	thrones	tv
0	1	0	1	0	1	1	1
1	0	1	1	0	1	1	1
2	0	0	1	1	0	1	0

Gambar 2. Hasil TF dari 3 dokumen

Pada penelitian ini juga menggunakan *sklearn library* sebagai alat untuk mengimplementasikan TF-IDF, dengan rumus IDF adalah:

$$idf(t) = \log_e \frac{n}{df(t)} + 1 \quad (2)$$

Karena *sklearn* menggunakan basis log e maka rumusnya menjadi sebagai berikut:

$$idf(t) = \ln \frac{n}{df(t)} + 1 \quad (3)$$

Dimana *n* adalah jumlah total dokumen dalam kumpulan dokumen, dan *idf*(*t*) adalah jumlah dokumen dalam kumpulan dokumen yang mengandung istilah *t*. Hasil IDF dari 3 dokumen seperti contoh BoW dapat dilihat pada Gambar 3.

amazing	best	game	great	series	thrones	tv
2.0986123	2.0986	1.0	2.0986	1.4055	1.0	1.4055

Gambar 3. Hasil IDF dari 3 dokumen sebagai Figure.1

Rumus TF-IDF sebagai berikut:

$$TF - IDF = tf\ x\ idf \quad (4)$$

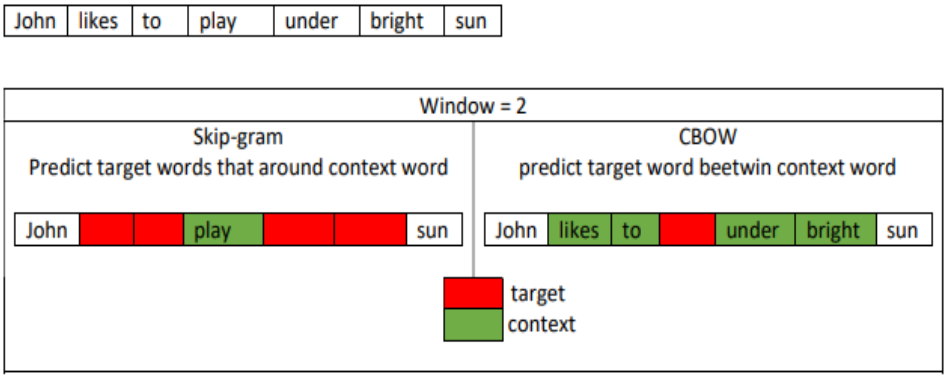
Adapun hasil TF-IDF dapat terlihat di Gambar 4.

	amazing	best	game	great	series	thrones	tv
0	2.098612	0.000000	1.0	0.000000	1.405465	1.0	1.405465
1	0.000000	2.098612	1.0	0.000000	1.405465	1.0	1.405465
2	0.000000	0.000000	1.0	2.098612	0.000000	1.0	0.000000

Gambar 4. Hasil TF-IDF dari 3 dokumen sebagai Figure.1

2.1.3 Word2Vec

Pada tahun 2013[2], tim Google yang dipimpin oleh Tomas Mikolov merilis teknik Word2Vec untuk penyematan kata, yang mencakup dua model: Skip-gram and Continuous Bag of Words (CBOW). Pada model CBOW, word2vec menggunakan kata-kata yang berada sebelum dan sesudah kata target dan dibatasi pada jendela prediksi kata target. Sedangkan skip-gram menggunakan sebuah kata untuk memprediksi kata yang berada sebelum dan sesudah kata yang dibatasi oleh jendela. Jendela adalah jumlah kata di kiri atau kanan kata sasaran. Sebagai contoh, ketika ukuran jendela diberikan 2, maka word2vec akan mempertimbangkan 2 kata sebelum dan 2 kata setelah kata yang terkait dengannya. Ilustrasi dari jendela dapat dilihat pada Gambar 5.



Gambar 5. Ilustrasi Arsitektur CBOW dengan ukuran jendela 2

2.2 Multinomial Naive Bayes (MNB)

Multinomial Naïve Bayes merupakan metode khusus dari Naïve Bayes sebagai metode text mining dalam proses klasifikasi teks menggunakan probabilitas kelas pada dokumen. Prosesnya dimulai dengan memasukkan data latih yang digunakan untuk pembelajaran kemudian menghitung probabilitas munculnya suatu kelas pada data latih.

Misalkan himpunan kelas dilambangkan dengan C. Misalkan N adalah ukuran kosakata kita. Maka MNB menentukan dokumen t_i pada kelas yang mempunyai probabilitas tertinggi Pr($c|t_i$), dengan menggunakan aturan Bayes, seperti pada artikel [10]:

$$\Pr(c|t_i) = \frac{\Pr(c)\Pr(t_i|c)}{\Pr(t_i)}, c \in C \quad (5)$$

Pr($t_i|c$) adalah probabilitas memperoleh dokumen seperti t_i di kelas c dan dihitung sebagai: (Rumus yang digunakan oleh perpustakaan scikit-learn):

$$\Pr(t_i|c) = \frac{N_{ci}+\alpha}{N_c+\alpha n\alpha} \quad (6)$$

Dimana:

$$N_{ci} = \sum_{t\in T} t_i \quad (7)$$

adalah berapa kali fitur i muncul dalam sampel kelas c di set pelatihan T, sedangkan

$$N_c = \sum_{i=1}^{n} N_{ci} \quad (8)$$

adalah jumlah total semua fitur untuk kelas c.

Prioritas pemulusan $\alpha \geq 0$ memperhitungkan fitur yang tidak ada dalam sampel pembelajaran dan mencegah probabilitas nol dalam komputasi lebih lanjut. Pengaturan $\alpha = 1$ disebut pemulusan Laplace, sedangkan$\alpha < 1$ disebut pemulusan Lidstone.

3 Metode Penelitian

Pada penelitian ini dilakukan klasifikasi spam email menggunakan perbandingan vektorisasi antara BoW, TFIDF dan Word2Vec untuk mendukung algoritma Multinomial Naive Bayes. Perbandingan dengan berbagai vektorisasi bertujuan untuk menentukan mana yang cocok untuk mengklasifikasikan email spam dengan menentukan hasil akurasi yang paling besar.

Beberapa tahapan metode yang digunakan dalam penelitian ini dimulai dari pencarian dan pengolahan dataset, preprocessing dataset, vektorisasi dan pembuatan model atau metode pembelajaran, hingga

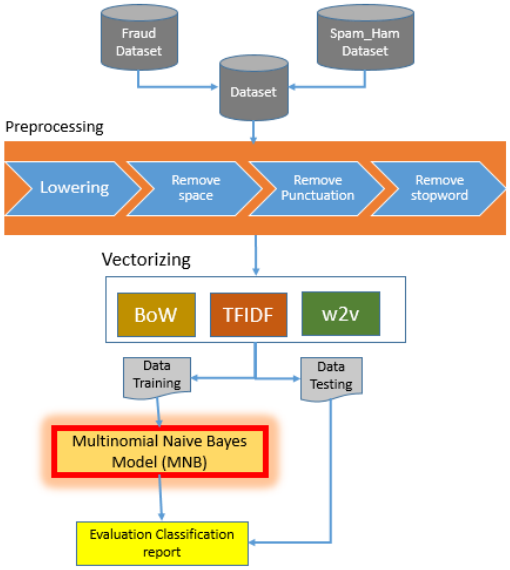
pengujian dataset dengan menggunakan vektorisasi yang berbeda secara bergantian. Hasil pengujian tersebut kemudian dibandingkan untuk menentukan vektorisasi terbaik untuk memperkirakan keakuratan klasifikasi email spam. Penjelasan dari tahap penelitian adalah sebagai Gambar 6.

3.1 Data Collection

Dataset yang digunakan dalam penelitian ini menggunakan data publik dari Kaggle yang berjumlah 5.572 email dengan menggunakan bahasa Inggris. https://www.kaggle.com/datasets/ashfakyeafi/spam-email-classification. Dan gabungkan dengan 5.592 email penipuan menggunakan bahasa Inggris yang diunduh dari https://doi.org/10.5281/zenodo.8339691. Total dataset menjadi 11.164 record dan dataset disimpan dalam format csv.

3.2 Proposed Method

Metode yang diusulkan untuk mengklasifikasikan email dan menentukan vektorisasi yang paling tepat digunakan dengan melihat hasil akurasi yang paling besar. Figure 6 menjelaskan alur metode yang diusulkan dalam penelitian ini.



Gambar 6. Metode yang diusulkan

3.3 Data Preprocessing

Dataset ini berada pada tahap preprocessing melalui lima proses yaitu:

1. Menurunkan huruf, proses ini akan mengubah semua huruf kapital menjadi huruf kecil.
2. Hapus spasi, ini akan menghilangkan spasi putih tambahan.
3. Hapus tanda baca, pada proses ini kami akan menghapus tanda baca pada teks email.
4. Hapus stopwords, proses ini akan menghilangkan setiap kata yang tergolong stopwords atau kata-kata yang kurang penting dalam teks email.

Preprocessing ini akan menghasilkan dataset baru yang siap digunakan untuk proses selanjutnya.

3.4 Experiment and Testing Method

Dalam penelitian ini, eksperimen dimulai dari menggabungkan dataset kemudian menuju ke tahap preprocessing dan membagi dataset akhir menjadi dua bagian, data latih, dan data uji. Data latih akan dipelajari pada proses vektorisasi dan kemudian tahap metode pembelajaran menggunakan algoritma MNB, sedangkan data uji digunakan untuk menguji keakuratannya. Proses percobaan dan pengujian menggunakan Google Colab atau Anaconda Python 3.10.12.

Proses vektorisasi untuk BoW dan TFIDF menggunakan perpustakaan sklearn dan untuk word2vec menggunakan perpustakaan gensim. Penelitian ini menggunakan model word2vec Continuous Bag of Words (CBOW). Pembuatan model MNB kami menggunakan *sklearn library* dengan parameter *alpha=1*, itu berarti penerapannya pemulusan *Laplace*.

3.5 Evaluation

Evaluasi pada penelitian ini dengan menghitung nilai presisi, recall dan akurasi. Penelitian ini menggunakan matriks confusion yang terdiri dari nilai true positive (TP), false positive (FP), true negative (TN) dan false negative (FN), di mana nilai true negative dan false negative menunjukkan adanya kesalahan klasifikasi teks pada dataset [11]. Karena

kita menggunakan model klasifikasi kelas jamak, maka rumusnya sebagai berikut [12]:

$$Precision_{micro} = \frac{\sum TP}{\sum TP+FP} x100\% \quad (9)$$

$$Precision_{macro} = \frac{\sum Precision_{class}}{N} x100\% \quad (10)$$

$$Recall_{micro} = \frac{\sum TP}{\sum TP+FN} x100\% \quad (11)$$

$$Recall_{macro} = \frac{\sum Recall_{class}}{N} x100\% \quad (12)$$

dimana N adalah jumlah seluruh kelas, dan *Precision1, Precision2, ..., PrecisionN* dan *Recall1, Recall2, ..., RecallN* adalah nilai presisi dan perolehan untuk setiap kelas.

Ada perbedaan mendasar antara rata-rata makro dan mikro dalam cara menggabungkan metrik kinerja. Rata-rata makro menghitung metrik kinerja setiap kelas (misalnya, presisi, perolehan) dan kemudian mengambil mean aritmetika di semua kelas. Rata-rata makro memberikan bobot yang sama untuk setiap kelas, berapa pun jumlah instancenya. Rata-rata mikro, mengagregasi jumlah positif sebenarnya, positif palsu, dan negatif palsu di semua kelas, lalu menghitung metrik kinerja berdasarkan jumlah total. Rata-rata mikro memberikan bobot yang sama untuk setiap instance, terlepas dari label kelas dan jumlah kasus di kelas tersebut.

$$Accuracy = \frac{\sum TP+TN}{\sum TP+TN+FP+FN} x100\% \quad (13)$$

Akurasi menunjukkan proporsi dokumen yang diklasifikasikan dengan benar di antara seluruh dokumen.

4 Hasil dan Pembahasan

Eksperimen dimulai dari pengecekan dataset dan ditemukan bahwa dataset tersebut banyak mengandung karakter non utf-8. Ini memicu kesalahan saat kami mencoba membaca data. Setelah kami membersihkan data, kami memiliki 8.901 catatan baik dengan label ham sebesar 4.825, penipuan sebesar 3.329 dan phishing sebesar 747.

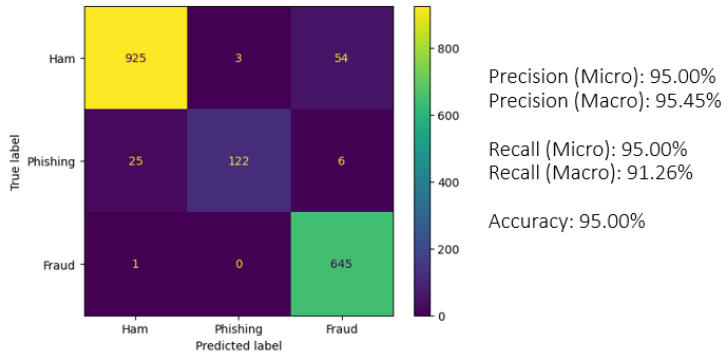
Langkah selanjutnya adalah melakukan langkah preprocessing yaitu menurunkan huruf, menghilangkan spasi berlebih, menghilangkan tanda baca dan menghilangkan stop word. Kemudian dilakukan proses vektorisasi BoW, TFIDF dan W2V. Proses BoW menggunakan *CountVectorizer* dari *sklearn* dengan default parameter. Proses TFIDF menggunakan *TfidfVectorizer* dari *sklearn* dengan parameter *norm=none*, *smooth_idf=False*. Dan w2v process menggunakan *Word2Vec* dari *gensim* dengan parameter : *vector_size*=100, *window*=5, *min_count*=5, *workers*=4, *sg*=0. Kami menggunakan parameter *sg*=0 agar bisa menggunakan Word2Vec dengan jenis CBoW (*Continuous Bag of Words*).

Pada penelitian ini akan menggunakan kumpulan data yang divektorkan untuk melatih model dan menguji model. Kami membagi kumpulan data menjadi 80% untuk data pelatihan dan 20% sebagai data pengujian. Model MNB dibuat menggunakan MultinomialNB dari *sklearn.naive_bayes*. Kami membuat model menggunakan parameter *alpha*=1, artinya mengimplementasikan *pemulusan Laplace*. Dan latih model menggunakan 80% kumpulan data vektor.

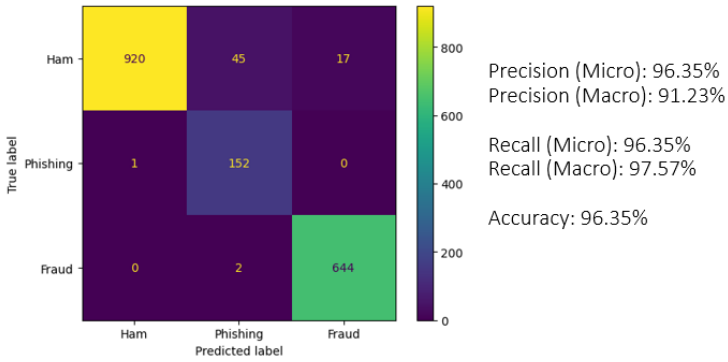
Matriks konfusi hasil masing-masing model dapat dilihat pada figure.7 untuk model MNB dengan dataset vektorisasi BoW, model MNB dengan dataset vektor TFIDF, dan model MNB dengan dataset vektorisasi w2v. Dengan menggunakan matriks ini, kami menghitung data presisi dan perolehan. Rangkuman data tersebut dapat dilihat pada Tabel 1 dan Tabel 2.

Dalam sistem pemfilteran spam, email spam yang salah teridentifikasi tidak seserius email non-spam yang salah teridentifikasi. Dengan kata lain, kesalahan identifikasi email non-spam lebih berisiko dibandingkan kesalahan identifikasi email spam, sehingga presisi harus besar dan recall juga harus besar.

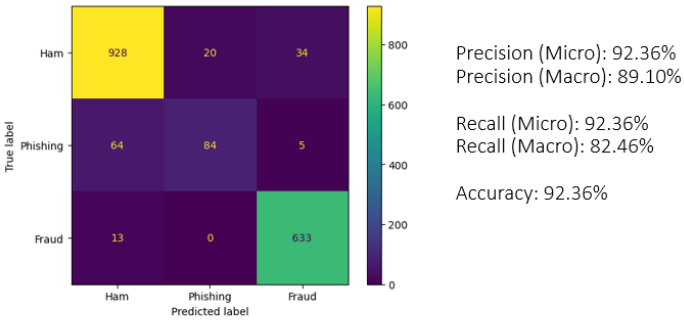
Hasilnya menunjukkan bahwa nilai *recall-makro MNB* dengan *w2v* sangat rendah dan hal ini dapat menyebabkan kesalahan identifikasi non-spam sebagai spam. Daripada *recall-nilai* makro *MNB* dengan *TFIDF* sangat tinggi karena dapat mendeteksi email spam dengan lebih baik.



Confusion matrix of MNB model with BoW vectorized dataset



Confusion matrix of MNB model with TFIDF vectorized dataset



Confusion matrix of MNB model with w2v vectorized dataset

Gambar 7. Matriks konfusi dan akurasi hasil

Tabel.1 Membandingkan ringkasan hasil evaluasi

Evaluation Result	MNB model with vectorizing:		
	BoW	TFIDF	w2v
Precision (Micro)	95.00%	96.35%	92.36%

Precision (Macro)	95.45%	91.23%	89.10%
Recall (Micro)	95.00%	96.35%	92.36%
Recall (Macro)	91.26%	97.57%	82.46%
Accuracy	95.00%	96.35%	92.36%

Tabel.2 Membandingkan hasil evaluasi per kelas

	MNB model with vectorizing :		
evaluation result per class	BoW	TFIDF	w2v
precision fraud	91.49%	97.43%	94.20%
precision ham	97.27%	99.89%	92.34%
precision phishing	97.60%	76.38%	80.77%
recall fraud	99.85%	99.69%	97.99%
recall ham	94.20%	93.69%	94.50%
recall phishing	79.74%	99.35%	54.90%

5 Kesimpulan

Berdasarkan hasil pengujian akurasi ketiga model, vektorisasi *TFIDF* mengungguli *BoW* dan *Word2vec*, akurasinya sebesar 96,35% untuk *MNB* dengan *TFIDF*, 95,00% untuk *MNB* dengan *BoW*, dan 92,36% untuk *MNB* dengan *Word2Vec*. Kinerja percobaan terbaik diperoleh dengan menggunakan vektorisasi *TFIDF*. Namun, pada perbedaan akurasi antara ketiga teknik tersebut secara statistik, menunjukkan bahwa 2 metode yaitu *TFIDF* dan *WoB* memiliki kinerja yang kompetitif dibandingkan metode *Word2Vec*. Keakuratan ketiga model ini bergantung pada kumpulan data yang digunakan, oleh karena itu kumpulan data lain dapat ditambahkan untuk penelitian selanjutnya.

6 Referensi

[1] Cross, C. and Gillett, R. (2020), "Exploiting trust for financial gain: an overview of business email compromise (BEC) fraud", Journal of Financial Crime, Vol. 27 No. 3, pp. 871-884. https://doi.org/10.1108/JFC-02-2020-0026

[2] E. M. Dharma, F. Lumban Gaol, H. Leslie, H. S. Warnars, and B. Soewito, “The Accuracy Comparison Among Word2vec, Glove, And Fasttext Towards Convolution Neural Network (Cnn) Text Classification,” J Theor Appl Inf Technol, vol. 31, no. 2, 2022, [Online].

[3] Daniela XHEMALI, Christopher J. HINDE and Roger G. STONE, Naïve Bayes

vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages, IJCSI International Journal of Computer Science Issues, Vol. 4, No. 1, 2009, ISSN (Online): 1694-0784, ISSN (Print): 1694-0814

[4] Androutsopoulos, Ion; Koutsias, John; Chandrinos, Konstantinos V.; Spyropoulos, Constantine D. (2000). [ACM Press the 23rd annual international ACM SIGIR conference - Athens, Greece (2000.07.24-2000.07.28)] Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00 - An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. , (0), 160–167. doi:10.1145/345508.345569

[5] Rusland, Nurul Fitriah; Wahid, Norfaradilla; Kasim, Shahreen; Hafit, Hanayanti (2017). Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets. IOP Conference Series: Materials Science and Engineering, 226(), 012091–. doi:10.1088/1757-899X/226/1/012091 disebutkan Feature Extraction nya Correlation based feature selection (CFS)

[6] Nadia Anjum, Dr Srinivasu Badugu, A Comparative Study on Classification Algorithms Using Different Feature Extraction And Vectorization Techniques For Text, Turkish Online Journal of Qualitative Inquiry (TOJQI) Volume 12, Issue 7, July 2021: 8216 – 8225

[7] Niken Larasati Octaviani;Eko Hari Rachmawanto;Christy Atika Sari;Ignatius Moses Setiadi De Rosal; (2020). Comparison of Multinomial Naïve Bayes Classifier, Support Vector Machine, and Recurrent Neural Network to Classify Email Spams . 2020 International Seminar on Application for Technology of Information and Communication (iSemantic), (), –. doi:10.1109/iSemantic50169.2020.9234296

[8] Kadam, Sumedh; Gala, Aayush; Gehlot, Pritesh; Kurup, Aditya; Ghag, Kranti (2018). [IEEE 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Pune, India (2018.8.16-2018.8.18)] 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) - Word Embedding Based Multinomial Naive Bayes Algorithm for Spam Filtering. , (), 1–5. doi:10.1109/ICCUBEA.2018.8697601

[9] Anita Kumari Singh, Mogalla Shashi. "Vectorization of Text Documents for Identifying Unifiable News Articles." International Journal of Advanced Computer Science and Applications Vol.10, No.7, 2019: 305-310. ISSN: 21565570, 2158107X

[10] Webb, Geoffrey I.; Yu, Xinghuo (2005). [Lecture Notes in Computer Science] AI 2004: Advances in Artificial Intelligence Volume 3339 || Multinomial Naive Bayes for Text Categorization Revisited. , 10.1007/b104336(Chapter 43), 488–499. doi:10.1007/978-3-540-30549-1_43

[11] Surya, Prabha PM; Seetha, Lakshmi V; Subbulakshmi, B (2019). [IEEE 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) - Coimbatore, India (2019.6.12-2019.6.14)] 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) - Analysis of user emotions and opinion using Multinomial Naive Bayes Classifier. , (), 410–415. doi:10.1109/ICECA.2019.8822096

[12] Helmi Setyawan, Muhammad Yusril; Awangga, Rolly Maulana; Efendi, Safif Rafi (2018). [IEEE 2018 International Conference on Applied Engineering (ICAE) - Batam, Indonesia (2018.10.3-2018.10.4)] 2018 International Conference on Applied Engineering (ICAE) - Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot. , (), 1–5. doi:10.1109/INCAE.2018.8579372