

Analisa Perbandingan Algoritma CNN Dan LSTM untuk Klasifikasi Pesan Cyberbullying pada Twitter

Alifqi Radjavani¹, Theopilus Bayu Sasongko²

alifqiradjavani@students.amikom.ac.id, theopilus.27@amikom.ac.id

Universitas Amikom Yogyakarta

Informasi Artikel	Abstrak
Diterima : 18 Jul 2023 Direview : 5 Ags 2023 Disetujui : 27 Ags 2023	Dengan meningkatnya penggunaan sosial media, cyberbullying telah mencapai titik puncak sepanjang masa. Anonimitas pada internet membuat cyberbullying sangat merusak, dikarenakan korban akan merasa jika tiada jalan keluar dari pelecehan tersebut. Setiap individu harus selalu waspada terhadap cyberbullying dan dihindari untuk selalu melindungi diri sendiri beserta orang lain dari hal ini. Pada kasus ini, penulis membuat model yang secara otomatis akan menandai tweet yang berpotensi membahayakan serta memecah pola pesan kebencian tersebut. Dataset yang disediakan oleh penulis berisi sekitar 48.000 tweet yang telah dilabeli sesuai dengan jenis dan data-data tersebut telah diseimbangkan dan berisi sekitar 8000 data. Penelitian ini membandingkan algoritma Convolutional Neural Network dengan Long Short-Term Memory untuk menentukan algoritma terbaik untuk dataset pada penelitian ini. Berdasarkan hasil penelitian yang sudah dilakukan disimpulkan jika Long Short-Term Memory adalah algoritma terbaik dengan f1-score 83.09%.
Kata Kunci	
Cyberbullying, Merusak, Membahayakan, Convolutional Neural Network, Long Short-Term Memory	

Keywords	Abstrak
Cyberbullying, Damaging, Harmful, Convolutional Neural Network, Long Short-Term Memory	<i>With the increasing use of social media, cyberbullying has reached an all-time high. The anonymity of the internet makes cyberbullying particularly damaging, as victims may feel that there is no escape from the harassment. Individuals should be aware of cyberbullying and to protect themselves and others from it. In this case, the authors built a model that automatically flags potentially harmful tweets and breaks down the pattern of hate messages. The dataset provided by the author contains about 48,000 tweets that have been labeled according to type and the data has been balanced and contains about 8000 data. This research compares the Convolutional Neural Network algorithm with Long Short-Term Memory to determine the best algorithm for the dataset in this study. Based on the results of the research that has been performed, it is concluded that Long Short-Term Memory is the best algorithm with an f1-score of 83.09%.</i>

A. Pendahuluan

Penggunaan sosial media telah mendapatkan momentum dalam beberapa tahun ini yang memungkinkan penggunaanya dapat berbagi ide, sudut pandang, atau memberikan komentar terhadap berbagai masalah [1]. Namun, kemudahan dalam akses pada sosial media ini menimbulkan resiko bagi mereka yang terlibat didalamnya, terutama bagi anak-anak hingga remaja dan *cyberbullying* adalah salah satunya [2].

Cyberbullying dapat diartikan sebagai tindakan pengucilan atau perundungan sosial yang dilakukan secara daring seperti mengirimkan pesan menyinggung secara sengaja, tidak sesuai fakta, dan menyebarkan informasi pribadi tanpa seizin pemiliknya [3]. Perilaku *cyberbullying* yang sering dijumpai meliputi penghinaan, pelecehan, pencemaran nama baik, menirukan identitas, menipu, hingga pengucilan [4]–[6].

Berdasarkan data dari badan pusat penelitian *cyberbullying*, 36,5% siswa sekolah menengah atas melaporkan bahwa mereka pernah menjadi korban *cyberbullying* [7]. Penelitian serupa juga pernah dilakukan sebelumnya di Indonesia, dimana 68,6% siswa sekolah menengah atas pernah mengalami *cyberbullying* [8].

Klasifikasi teks merupakan proses mengelompokkan teks ke dalam kelas-kelas yang berbeda berdasarkan lingkup teks [9]. Istilah kategorisasi teks dan klasifikasi teks juga sering digunakan untuk menggambarkan proses memprediksi kategori atau lingkup yang telah ditentukan sebelumnya pada dokumen tertentu [10].

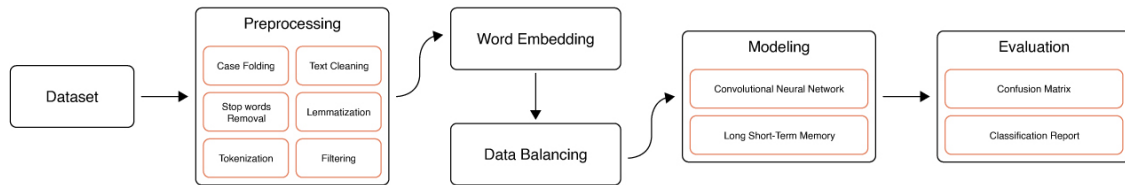
Penelitian yang terkait dengan klasifikasi teks *cyberbullying* yang pernah dilakukan sebelumnya membandingkan beberapa kombinasi *word embedding* dan *classifier model* [11]. Data yang digunakan berupa *dataset open source* sebanyak 48000 data dengan kelas *Not Cyberbullying*, *Age*, *Ethnicity*, *Gender*, *Religion*, dan *Other Cyberbullying* dimana masing-masing kelas memiliki 8000 sampel data yang diambil secara acak. Hasil penelitian menunjukkan bahwa model dengan *Bag of Words* dan *XGBoost* merupakan model terbaik pada dataset utuh (40.000 tweet) dengan akurasi sebesar 94,38% dan F1 Score sebesar 94,44%.

Penelitian terkait lainnya, melakukan perbandingan algoritma *Long Short-Term Memory* (LSTM) dan *Convolutional Neural Network* (CNN) pada analisis sentimen berbasis aspek [12]. Metode yang digunakan pada penelitian tersebut menggunakan arsitektur *Bidirectional Attention Neural Networks* (BANN) dan *Convolutional Attention Neural Networks* (CANN). Dataset berasal dari SemEval pada tahun 2014 yang digunakan untuk mengevaluasi kedua arsitektur. Dari penelitian, disimpulkan bahwa LSTM sangat cocok untuk menyusun informasi dan ketergantungan jarak jauh dengan tingkat akurasi 80,71%, sementara CNN cukup baik dalam mengekstraksi fitur lokal dan invarian dengan akurasi sebesar 78,04%.

Berdasarkan uraian diatas, penelitian ini membandingkan performa dari dua algoritma *Neural Networks* yaitu *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) untuk klasifikasi pesan *cyberbullying* pada media sosial twitter. Hasil dari penelitian ini diharapkan mampu memberikan pandangan yang lebih baik terhadap perbandingan dari kedua jenis algoritma untuk menentukan algoritma yang efektif untuk pembuatan model klasifikasi teks untuk menandai pesan atau *tweet* yang berpotensi membahayakan, dimana model ini nantinya akan digunakan untuk menyelidiki dan mencari tahu pola pada setiap jenis *cyberbullying*.

B. Metode Penelitian

Diagram alur akan analisa perbandingan algoritma untuk klasifikasi pesan *cyberbullying* pada twitter dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Penelitian

1. Dataset

Data yang digunakan merupakan *public dataset* pada penelitian sebelumnya yang dilakukan oleh [11]. Dataset ini kurang lebih berisi 48.000 *tweet* yang telah dilabeli sesuai dengan kelas *cyberbullying* yang terdiri dari *age*, *ethnicity*, *gender*, *religion*, *other bullying*, dan *not bullying*. Data tersebut sudah diseimbangkan dan memiliki nilai sekitar 8000 *tweet* pada masing-masing kelas.

Tabel 1. Informasi *Dataset*

No	Column	Count	Data Type
0	tweet_text	47.692	object
1	cyberbullying_type	47.692	object

Tabel 2. Jumlah Data pada Setiap Kelas

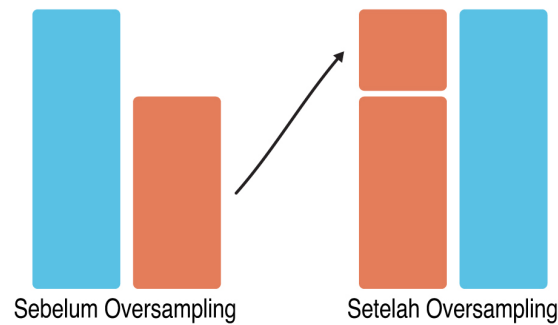
No	cyberbullying_type	Count	Data Type
0	age	7992	int64
1	ethnicity	7961	int64
2	gender	7973	int64
3	not_cyberbullying	7945	int64
4	other_cyberbullying	7823	int64
5	religion	7998	int64

2. Preprocessing

Tahap *Preprocessing* ini sangat penting untuk melakukan analisis, mengidentifikasi korelasi, dan pembuatan model *Natural Language Processing* (NLP) [13]. *Natural Language Toolkit* (NLTK) merupakan salah satu *library open-source* Python yang banyak digunakan untuk keperluan NLP [14]. NLTK menyediakan beberapa teknik *text preprocessing*. Pembersihan data mencakup berbagai proses untuk memastikan data layak untuk dianalisis [15]. *Case folding* mengganti huruf kapital yang terdapat pada paragraf menjadi huruf kecil [16]. *Stop words Removal* menghilangkan kata yang sering muncul namun tidak berarti seperti kata “yang”, “dan”, “itu”, “dari” dan sebagainya [17]. *Stemming* digunakan untuk menghilangkan imbuhan pada suatu teks untuk mendapatkan kata dasarnya [18]. Namun, metode *Lemmatization* menghasilkan kata dasar yang lebih akurat dibandingkan dengan *stemming* dikarenakan menggunakan kamus *Stop words* dan *WordNet* [19]. *Tokenization* memotong teks menjadi bagian yang lebih kecil atau sering dikenal dengan istilah dengan *token* [20].

3. Data Balancing

Ketidakseimbangan data sering dijumpai pada *dataset* yang memiliki banyak kelas atau biasa disebut dengan *multi-class* [21]. *Dataset* dapat dikatakan tidak seimbang ketika sampel dari salah satu kelas lebih banyak atau lebih sedikit dibandingkan kelas lainnya [22]. Salah satu teknik yang efektif untuk mengatasi masalah data yang tidak seimbang yaitu dengan *oversampling* [23]. *Oversampling* secara acak menduplikat data yang sedikit atau minoritas untuk meningkatkan jumlahnya [24].



Gambar 2. Oversampling

4. Word Embedding

Pembobotan kata mengubah kata menjadi bentuk vektor lalu memperkirakan fitur dari sejumlah teks dengan makna yang hampir mirip dalam konteks yang sama [25]. Tingkat kemiripan diantara kedua teks dapat diukur dengan menggunakan *cosine similarity* [26]. Untuk mencari nilai *cosine similarity* dapat dihitung menggunakan persamaan (1).

$$\cos \theta = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||} \quad (1)$$

Dimana:

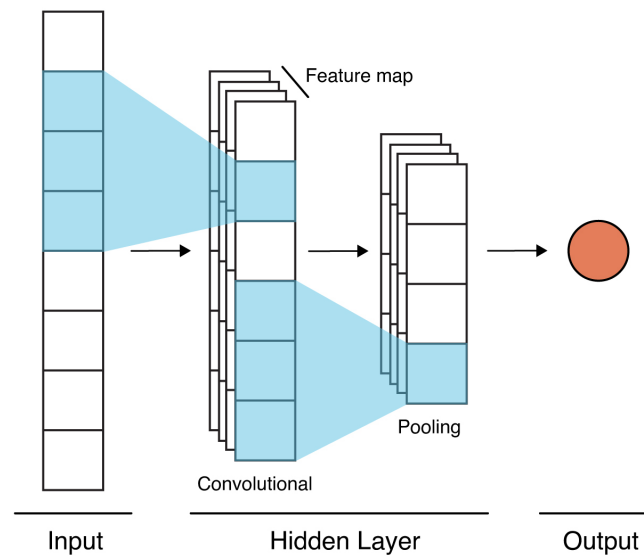
- $\vec{x} \cdot \vec{y}$: hasil perkalian dari vektor x dan y
- $||\vec{x}||$ dan $||\vec{y}||$: panjang dari dua vektor x dan y
- $||\vec{x}|| \times ||\vec{y}||$: hasil perkalian silang dari dua vektor x dan y

Teknik pembobotan kata paling populer yaitu *Word2Vec* [27], [28]. *Word2Vec* mempelajari keseluruhan kata pada dokumen dengan menggunakan *neural networks*, kemudian memetakan kata-kata yang memiliki arti yang sama kedalam vektor yang berdekatan satu sama lain [29].

5. Convolutional Neural Network

CNN merupakan salah satu bentuk *Artificial Neural Network* (ANN) yang sangat cocok untuk menganalisis dan memproses data dengan struktur *grid* seperti halnya gambar. CNN pada dasarnya terdiri dari serangkaian *hidden layer* yang masing-masing melakukan pemrosesan konvolusi pada data input. Operasi konvolusi juga menambahkan *weight* atau *filter* ke data *input* yang menghasilkan *output* [30]. *Transformer* seperti *One-Dimensional Convolutional Neural Network*

(1D-CNN) mulai mendapatkan kepopuleran pada penerapan *Natural Language Processing* (NLP) [31].



Gambar 3. Arsitektur 1D-CNN

Convolutional layer merupakan lapisan inti dimana sebagian besar beban komputasi jaringan berada pada jaringan ini. Lapisan ini melakukan perkalian antara matrik kumpulan *parameter* yang dipelajari atau sering dikenal sebagai *kernel*, dengan matrik *input*. Rumus perhitungan *convolutional layer* dapat dilihat pada persamaan (2).

$$W_{out} = \frac{W - F + 2P}{S} + 1 \quad (2)$$

Dimana:

- W : spasial data *input*
- F : spasial *filter*
- P : *padding* untuk menyesuaikan ukuran data *input*
- S : *stride* atau pergeseran *filter*

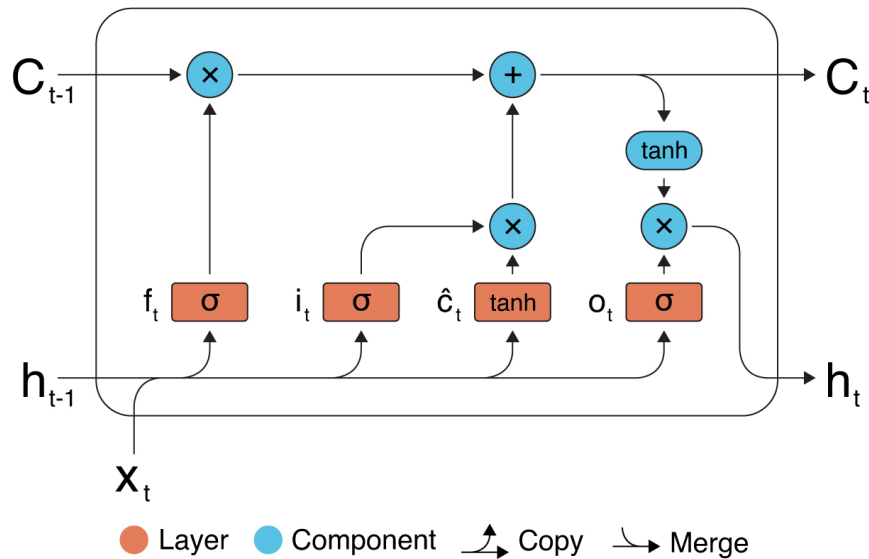
Pooling layer menggantikan output dari jaringan berdasarkan informasi dari output terdekat. *Max pooling* adalah salah satu *kernel* yang populer pada lapisan ini, dimana metode ini menghasilkan *output* maksimal dari *output* sekitar. Rumus *kernel max pooling* dapat dituliskan menggunakan persamaan (3).

$$W_{out} = \frac{W - F}{S} + 1 \quad (3)$$

6. Long Short-Term Memory

LSTM dirancang berdasarkan bentuk umum dari *Recurrent Neural Network* (RNN) untuk menghilangkan masalah dependensi jangka panjang pada algoritma. LSTM terdiri dari sekumpulan *memory cell* unik yang menggantikan *hidden layer* pada RNN dengan keadaan *memory cell* sebagai *key*. Setiap *memory cell* memiliki tiga

sigmoid layer dan satu *tanh layer*. LSTM menyaring informasi menggunakan *gate* untuk mempertahankan dan memperbaharui keadaan *memory cell*. Struktur *gate* meliputi *input*, *forgotten*, dan *output* [32].



Gambar 4. Arsitektur LSTM

Forgotten gate (f_t) berfungsi untuk menyimpan seberapa banyak *cell state* (C_{t-1}) sebelumnya yang disimpan untuk *cell state* (C_t) sekarang dengan menggunakan persamaan (4). Hasil perhitungan akan menghasilkan nilai antara 1 dan 0 dimana nilai 1 berarti *cell state* akan disimpan sedangkan 0 *cell state* akan diabaikan.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (4)$$

Dimana:

- σ : *sigmoid layer*
- W_f : *weight matrix*
- h_{t-1} : *output dari memory sebelumnya*
- x_t : *input saat ini*
- b_f : *bias dari forgotten gate*

Input gate menentukan seberapa banyak *input* (x_t) saat ini yang disediakan ke dalam *cell state* (C_t) dan mencegah informasi yang kurang penting untuk masuk ke *memory cell*. *Input gate* memiliki dua fungsi. Fungsi pertama yaitu menemukan keadaan *cell state* yang harus diperbaharui dengan menggunakan persamaan (5). *Vector* (\hat{c}_t) yang dibuat melalui *lanh layer* digunakan untuk mengendalikan seberapa banyak informasi yang akan ditambahkan dihitung menggunakan persamaan (6). Fungsi kedua memperbaharui informasi kedalam *cell state* melalui persamaan (7).

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (5)$$

$$\hat{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (6)$$

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \quad (7)$$

Dimana:

W_σ : weight dari *sigmoid layer*
 b_i : bias dari *input*
 \tanh : *tahn layer*
 W_c : weight dari *cell state*
 b_c : bias dari *cell state*

Output gate (O_t) menentukan seberapa banyak *cell state* yang akan dibuang. Informasi ini ditentukan oleh *sigmoid layer* yang kemudian diproses oleh *tanh layer* dan dikalikan dengan *output* dari *sigmoid layer* untuk mendapatkan nilai *output* sementara menggunakan persamaan (8). Untuk hasil akhir *output* dapat didefinisikan dengan persamaan (9).

$$O_t = \sigma(W_\sigma \cdot [h_{t-1}, x_t] + b_o) \quad (8)$$

$$h_t = O_t \times \tanh(C_t) \quad (9)$$

Dimana:

b_o : bias dari *output*

7. Evaluation

Performa dari algoritma digambarkan dengan menggunakan *confusion matrix* yang dilengkapi dengan *classification report*. *Confusion Matrix* terdiri dari baris dan kolom dimana setiap baris berisi kelas yang sebenarnya, dan setiap kolom berisi kelas yang diprediksi [33].

Tabel 3. Confusion Matrix

Class	Predict	
	Positive	Negative
Actual	Positive	TP FN
	Negative	FP TN

Berdasarkan tabel 3, ada empat kombinasi yang bernama *True Positive* (TP), *False Positive* (FP), *True Negative* (TN), *False Negative* (FN). Empat kombinasi ini digunakan sebagai pengukuran performa klasifikasi, yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Nilai dari pengukuran performa klasifikasi ini dapat dihitung menggunakan persamaan (10-13).

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \quad (10)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (11)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (12)$$

$$F1\ score = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \quad (13)$$

C. Hasil dan Pembahasan

1. Preprocessing

Dataset melalui beberapa tahapan *preprocessing* dengan memanfaatkan algoritma yang ada di *library* NLTK. Tahapan yang pertama yaitu dengan mengubah semua *tweet* menjadi huruf kecil dengan *case folding*. Kemudian dilakukan pembersihan pada *tweet* dengan memisahkan kata yang kurang penting seperti tanda baca, *emoji*, tautan, dan *stop words*. Setelah itu, *tweet* yang sudah melalui tahap pembersihan kemudian dilakukan *lemmatization* untuk mengubah suatu kata menjadi kata dasar dengan memperhatikan konteks dari kata tersebut. Berikut adalah beberapa contoh data sebelum dan sesudah dilakukannya *preprocessing*.

Tabel 4. Sebelum *Preprocessing*

<i>tweet_text</i>	<i>cyberbullying_type</i>
Ever since I met you no one else has even been worth thinki...	not_cyberbullying
Guy who has been making gay rape jokes for years tries his...	gender
But we're Muslim you idiot. And that's mad disrespectful to...	religion
@ProoN i would rather have teeth pulled.	other_cyberbullying
we have a high school bully in our hands	age
@_Valencia123 fuck you, fucking dumb hoe bitch nigger -.-	ethnicity

Tabel 5. Setelah *Preprocessing*

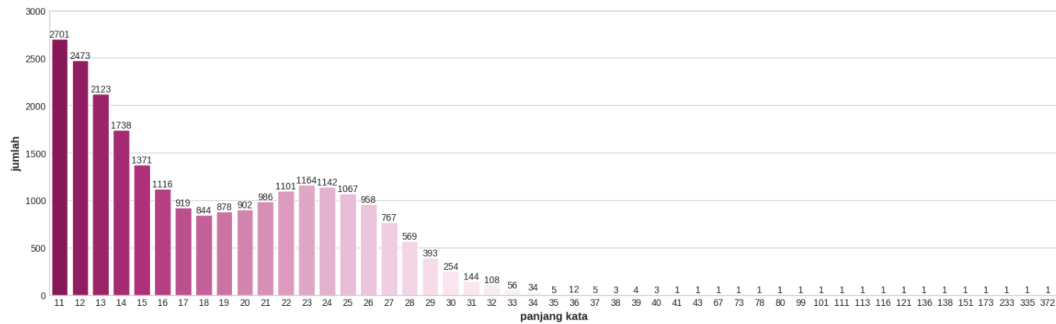
<i>tweet_text</i>	<i>cyberbullying_type</i>
ever since met one else even worth thinking	not_cyberbullying
guy making gay rape joke year try best deflection	gender
muslim idiot mad disrespectful family	religion
would rather teeth pulled	other_cyberbullying
high school bully hand	age
fuck fucking dumb hoe bitch nigger	ethnicity

2. Word Embedding

Setelah *tweet* telah melalui proses *preprocessing*, selanjutnya data akan diubah menjadi bentuk vektor dengan menggunakan *Tokenizer* pada *library* NLTK. Setelah data berubah menjadi vektor, data kemudian dapat dilatih menggunakan *Word2Vec* untuk mengelompokkan kata-kata di setiap data yang memiliki kemiripan antar satu sama lain.

3. Data Balancing

Setelah melalui tahap *preprocessing* dan *word embedding*, penulis memeriksa jumlah data berdasarkan panjang kata yang dimiliki pada setiap data.



Gambar 5. Jumlah *Tweet* Berdasarkan Panjang Kata

Pada Gambar 2, didapati ada beberapa *tweet* dengan kata yang panjang namun datanya sedikit dimana data-data ini dapat diabaikan. Dengan metode pembagian kuantil sebesar 99.5%, data yang digunakan oleh penulis adalah *tweet* dengan maksimal 31 kata. Penghapusan data pada tahap analisis data mengakibatkan data pada setiap kelas menjadi tidak seimbang. Ketidakseimbangan data setiap kelas dapat dilihat pada Tabel 6.

Tabel 6. Jumlah Data Setiap Kelas

No	<i>cyberbullying_type</i>	Count
0	age	4969
1	ethnicity	4709
2	gender	4629
3	not_cyberbullying	3880
4	other_cyberbullying	3138
5	religion	5017

Penulis menggunakan *RandomOverSampler* yang merupakan salah satu teknik *oversampling* pada *library imbalanced-learn*. Cara kerja *RandomOverSampler* cukup sederhana, yaitu dengan menghasilkan sampel baru dengan mengambil sampel secara acak pada sampel yang ada pada dataset.

Tabel 7. Setelah *Oversampling*

No	<i>cyberbullying_type</i>	Count
0	age	5017
1	ethnicity	5017
2	gender	5017
3	not_cyberbullying	5017
4	other_cyberbullying	5017
5	religion	5017

4. Modeling

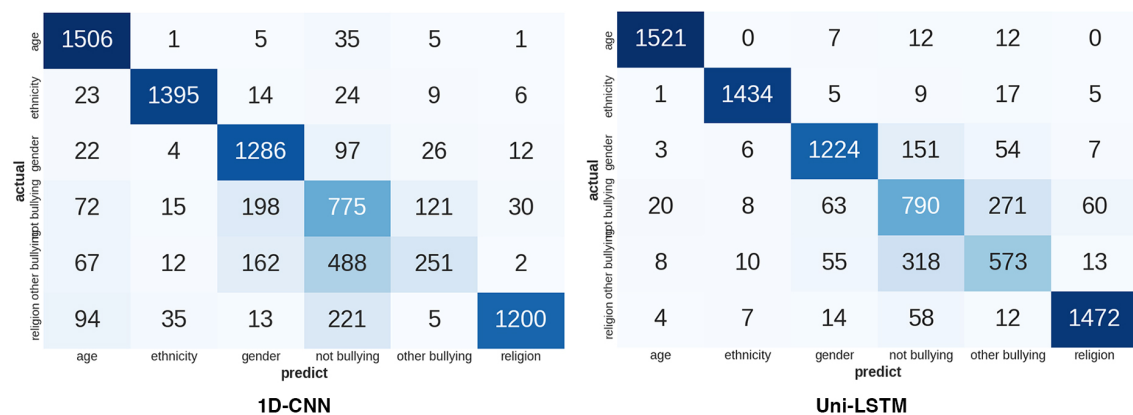
Pembuatan model pada algoritma CNN menggunakan arsitektur *One-Dimensional Convolutional Neural Network* (1D-CNN) sedangkan pada algoritma LSTM menggunakan arsitektur *Unidirectional Long Short-Term Memory* (Uni-LSTM). Parameter yang digunakan pada kedua model dapat dilihat pada Tabel 8.

Tabel 8. Model Summary

<i>Parameter</i>	<i>Value</i>
Split ratio	80 : 20
Embedding	200
Dimension	96
Learning rate	4e-4
Optimizer	Adam

5. Evaluation

Evaluasi dilakukan melakukan *training* dan *testing* pada model sebanyak sepuluh kali kemudian diambil hasil yang terbaik. Hasil klasifikasi pada kedua model disajikan dengan bentuk *confusion matrix* pada Gambar 6 dan *classification report* pada Tabel 9 dan Tabel 10.

**Gambar 6. Confusion Matrix****Tabel 9. Classification Report 1D-CNN**

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
age	0.84	0.97	0.90	1553
ethnicity	0.95	0.95	0.95	1471
gender	0.77	0.89	0.82	1447
not bullying	0.47	0.64	0.54	1211
other bullying	0.60	0.26	0.36	982
religion	0.96	0.77	0.85	1568
accuracy			0.78	8232
macro avg	0.77	0.74	0.74	8232
Weighted avg	0.79	0.78	0.77	8232

Tabel 10. Classification Report Uni-LSTM

	<i>precision</i>	<i>recall</i>	<i>f1-score</i>	<i>support</i>
age	0.98	0.98	0.98	1552
ethnicity	0.98	0.97	0.98	1471
gender	0.89	0.85	0.87	1445
not bullying	0.59	0.65	0.62	1212
other bullying	0.61	0.59	0.60	977
religion	0.95	0.94	0.94	1567
accuracy			0.85	8224
macro avg	0.83	0.83	0.83	8224
Weighted avg	0.86	0.85	0.85	8224

Berdasarkan Tabel 9 dan Tabel 10, penulis menghitung nilai *accuracy* dan rata-rata dari setiap *precision*, *recall*, dan *f1-score* untuk kedua model. Kemudian hasil dari perhitungan tersebut diletakan pada tabel perbandingan yang dapat dilihat pada Tabel 11.

Tabel 11. Perbandingan *Evaluation Metrics*

	<i>accuracy</i>	<i>precision</i>	<i>recall</i>	<i>f1-score</i>
1D-CNN	77.90%	76.64%	74.46%	73.84%
Uni-LSTM	85.28%	83.27%	82.99%	83.09%

Accuracy digunakan sebagai acuan jika hasil klasifikasi memiliki nilai *False Positive* (FP) dan *False Negative* (FN) yang sama, sebagai gantinya penulis menggunakan *f1-score* sebagai parameter penilaian performa dari kedua model yang ada. *F1-score* sendiri merupakan rata-rata kumulatif dari *precision* dan *recall* yang dapat diartikan semakin tinggi nilai *f1-score* maka menandakan bahwa model memiliki *precision* dan *recall* yang baik. Selisih nilai *f1-score* diantara kedua model sebesar 9.25% dan merupakan selisih nilai terbesar diantara semua *evaluation metrics* yang ada.

D. Simpulan

Berdasarkan hasil yang diperoleh dari penelitian yang telah dilakukan, dapat disimpulkan bahwa model dengan arsitektur *Unidirectional Long Short-Term Memory* lebih unggul dibandingkan dengan *One-Dimensional Convolutional Neural Network* pada semua *evaluation metrics* yang ada sehingga algoritma *Long Short-Term Memory* lebih cocok untuk mengklasifikasikan jenis *cyberbullying* pada twitter dengan *accuracy* sebesar 85.28%, dengan rata-rata *precision* sebesar 83.27%, rata-rata *recall* 82.99%, dan rata-rata *f1-score* sebesar 83.09% pada klasifikasi *tweet* sebanyak 8224 data. Meskipun algoritma *Convolutional Neural Network* dapat digunakan dengan cukup baik pada penelitian ini, namun terlihat jika CNN kesulitan dalam mengklasifikasikan kelas *not bullying* dan *other bullying*.

E. Referensi

- [1] D. H. Abd, A. R. Abbas, and A. T. Sadiq, "Analyzing sentiment system to specify polarity by lexicon-based," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 1, pp. 283–289, Feb. 2021, doi: 10.11591/eei.v10i1.2471.
- [2] C. Evangelio, P. Rodríguez-González, J. Fernández-Río, and S. Gonzalez-Villora, "Cyberbullying in elementary and middle school students: A systematic review," *Comput Educ*, vol. 176, p. 104356, Jan. 2022, doi: 10.1016/j.compedu.2021.104356.
- [3] M. Rodríguez-Enríquez, M. Bennasar-Veny, A. Leiva, M. Garaigordobil, and A. M. Yañez, "Cybervictimization among secondary students: social networking time, personality traits and parental education," *BMC Public Health*, vol. 19, no. 1, p. 1499, Dec. 2019, doi: 10.1186/s12889-019-7876-9.
- [4] B. S. Yadav and H. Kaur, "A review on the problem of adolescent due to the excessive use of technology: Cyberbully," *Int. J. Sci. Technol. Res*, vol. 9, pp. 3537–3541, 2020.
- [5] W. Cassidy, C. Faucher, and M. Jackson, "Cyberbullying among youth: A comprehensive review of current international research and its implications

- and application to policy and practice," *Sch Psychol Int*, vol. 34, no. 6, pp. 575–612, Dec. 2013, doi: 10.1177/0143034313479697.
- [6] J. Acosta, M. Chinman, P. Ebener, P. S. Malone, A. Phillips, and A. Wilks, "Evaluation of a Whole-School Change Intervention: Findings from a Two-Year Cluster-Randomized Trial of the Restorative Practices Intervention," *J Youth Adolesc*, vol. 48, no. 5, pp. 876–890, May 2019, doi: 10.1007/s10964-019-01013-2.
- [7] T. N. Sergeantanis *et al.*, "Cyberbullying and Obesity in Adolescents: Prevalence and Associations in Seven European Countries of the EU NET ADB Survey," *Children*, vol. 8, no. 3, p. 235, Mar. 2021, doi: 10.3390/children8030235.
- [8] R. Syah and I. Hermawati, "The prevention efforts on cyberbullying case for Indonesian adolescent social media users," *Jurnal Penelitian Kesejahteraan Sosial*, vol. 17, no. 2, pp. 131–146, 2018.
- [9] S. U. Hassan, J. Ahamed, and K. Ahmad, "Analytics of machine learning-based algorithms for text classification," *Sustainable Operations and Computers*, vol. 3, pp. 238–248, 2022, doi: 10.1016/j.susoc.2022.03.001.
- [10] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Inf Process Manag*, vol. 57, no. 1, p. 102121, Jan. 2020, doi: 10.1016/J.IPM.2019.102121.
- [11] J. Wang, K. Fu, and C. T. Lu, "SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection," in *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1699–1708. doi: 10.1109/BigData50022.2020.9378065.
- [12] W. Quan, Z. Chen, J. Gao, and X. T. Hu, "Comparative Study of CNN and LSTM based Attention Neural Networks for Aspect-Level Opinion Mining," in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2018, pp. 2141–2150. doi: 10.1109/BigData.2018.8622150.
- [13] A. Petukhova and N. Fachada, "TextCL: A Python package for NLP preprocessing tasks," *SoftwareX*, vol. 19, p. 101122, Jul. 2022, doi: 10.1016/j.softx.2022.101122.
- [14] V. N. Gudivada and K. Arbabifard, "Open-Source Libraries, Application Frameworks, and Workflow Systems for NLP," 2018, pp. 31–50. doi: 10.1016/bs.host.2018.07.007.
- [15] M. Helal and A. Al-Reyashi, "Peer-Reviewed Journal EXPLORING THE EFFECTIVENESS OF DIFFERENT DATA CLEANING TECHNIQUES FOR IMPROVING DATA QUALITY IN MACHINE LEARNING," *Humanities and Natural Sciences Journal El Reyashi*, vol. 2023, no. 7, p. 4, 2023, doi: 10.53796/hnsj.
- [16] M. A. Rosid, A. S. Fitriani, I. R. I. Astutik, N. I. Mulloh, and H. A. Gozali, "Improving Text Preprocessing For Student Complaint Document Classification Using Sastrawi," *IOP Conf Ser Mater Sci Eng*, vol. 874, no. 1, p. 012017, Jun. 2020, doi: 10.1088/1757-899X/874/1/012017.
- [17] K. Taghandiki and E. R. Ehsan, "Implementation of a noisy hyperlink removal system: A semantic and relatedness approach," *arXiv preprint arXiv:2303.03321*, Mar. 2023, [Online]. Available: <http://arxiv.org/abs/2303.03321>

- [18] A. Jabbar, S. Iqbal, M. I. Tamimy, S. Hussain, and A. Akhunzada, "Empirical evaluation and study of text stemming algorithms," *Artif Intell Rev*, vol. 53, no. 8, pp. 5559–5588, Dec. 2020, doi: 10.1007/S10462-020-09828-3/METRICS.
- [19] K. Makkar, P. Kumar, M. Poriye, and S. Aggarwal, "Improvisation in opinion mining using data preprocessing techniques based on consumer's review," *International Journal of Advanced Technology and Engineering Exploration*, vol. 10, no. 99, p. 257, 2023.
- [20] B. Khemani and A. Adgaonkar, "A Review on Reddit News Headlines with NLTK tool," *SSRN Electronic Journal*, 2021, doi: 10.2139/ssrn.3834240.
- [21] W. C. Sleeman IV and B. Krawczyk, "Bagging Using Instance-Level Difficulty for Multi-Class Imbalanced Big Data Classification on Spark," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2019, pp. 2484–2493. doi: 10.1109/BigData47090.2019.9006058.
- [22] N. Japkowicz, "Learning from Imbalanced Data Sets: A Comparison of Various Strategies," *Learning from Imbalanced Data Sets: A Comparison of Various Strategies*, vol. 68, Jun. 2000.
- [23] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review," *J Big Data*, vol. 7, no. 1, p. 70, Dec. 2020, doi: 10.1186/s40537-020-00349-y.
- [24] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "KNNOR: An oversampling technique for imbalanced datasets," *Appl Soft Comput*, vol. 115, p. 108288, Jan. 2022, doi: 10.1016/j.asoc.2021.108288.
- [25] S. L. Frisby, A. D. Halai, C. R. Cox, M. A. Lambon Ralph, and T. T. Rogers, "Decoding semantic representations in mind and brain," *Trends Cogn Sci*, vol. 27, no. 3, pp. 258–281, Mar. 2023, doi: 10.1016/j.tics.2022.12.006.
- [26] N. Ritha *et al.*, "Sentiment Analysis of Health Protocol Policy Using K-Nearest Neighbor and Cosine Similarity," in *ICSED TI 2022: Proceedings of the 1st International Conference on Sustainable Engineering Development and Technological Innovation, ICSED TI 2022, 11-13 October 2022, Tanjungpinang, Indonesia*, European Alliance for Innovation, 2023, p. 195.
- [27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Adv Neural Inf Process Syst*, vol. 26, 2013.
- [29] Z. H. Kilimci and S. Akyokus, "Deep Learning- and Word Embedding-Based Heterogeneous Classifier Ensembles for Text Classification," *Complexity*, vol. 2018, pp. 1–10, Oct. 2018, doi: 10.1155/2018/7130146.
- [30] F. Viel, R. C. Maciel, L. O. Seman, C. A. Zeferino, E. A. Bezerra, and V. R. Q. Leithardt, "Hyperspectral Image Classification: An Analysis Employing CNN, LSTM, Transformer, and Attention Mechanism," *IEEE Access*, vol. 11, pp. 24835–24850, 2023, doi: 10.1109/ACCESS.2023.3255164.
- [31] A. Vaswani *et al.*, "Attention Is All You Need," *Adv Neural Inf Process Syst*, vol. 2017-December, pp. 5999–6009, Jun. 2017, Accessed: Jul. 15, 2023. [Online]. Available: <https://arxiv.org/abs/1706.03762v5>

-
- [32] J. Qiu, B. Wang, and C. Zhou, "Forecasting stock prices with long-short term memory neural network based on attention mechanism," *PLoS One*, vol. 15, no. 1, p. e0227222, Jan. 2020, doi: 10.1371/journal.pone.0227222.
 - [33] I. Markoulidakis, G. Kopsiaftis, I. Rallis, and I. Georgoulas, "Multi-Class Confusion Matrix Reduction method and its application on Net Promoter Score classification problem," in *The 14th PErvasive Technologies Related to Assistive Environments Conference*, New York, NY, USA: ACM, Jun. 2021, pp. 412–419. doi: 10.1145/3453892.3461323.