InComTech: Jurnal Telekomunikasi dan Komputer
vol.13, no.3, Desember 2023, 189-195
http://publikasi.mercubuana.ac.id/index.php/Incomtech
P-ISSN: 2085-4811 E-ISSN: 2579-6089

Pengembangan *Intrusion Detection System* (Ids) Berbasis *Machine Learning*

Ady Suryadi [1*], Marza Ihsan Marzuki [2]

Program Studi Magister Teknik Elektro, Universitas Mercu Buana
Jl. Meruya Selatan, Jakarta 11650, Indonesia
Email : 55419120016@student.mercubuana.ac.id[1*],
marza.ihsan@mercubuana.ac.id[2]

Abstract:
Penggunaan internet yang terus meningkat memerlukan sistem deteksi serangan yang handal agar penyusup atau *cracker* yang hendak melakukan *cyberattacks* dapat terdeteksi dengan cepat. Mitigasi dan pertahanan dari ancaman serangan *cyber* menjadi sangat penting mengingat masyarakat sudah mulai ketergantungan pada teknologi internet yang bisa mengancam setiap saat. Ketika sejumlah besar paket datang, maka perlu dideteksi apakah paket tersebut paket data normal atau paket data serangan. *Intrusion Detection System* (IDS) dapat digunakan untuk mendeteksi setiap serangan pada jaringan atau sistem informasi. Deteksi anomali adalah jenis IDS yang mendeteksi serangan anomali pada jaringan berdasarkan probabilitas statistik. Pada penelitian ini deteksi serangan dilakukan dengan menggunakan metode *Knowledge Discovery in Databases* (KDD) berbasis *machine learning* untuk menganalisis serangan berdasarkan 2 (dua) sumber *dataset yaitu* UNSW-NB15 dan CICIDS2017. Algoritma J48, naïve bayes dan AdaBoostM1 digunakan untuk melakukan klasifikasi serangan. Pemrosesan data menggunakan *tools* WEKA. Seleksi jumlah atribut dilakukan menggunakan metode CFs-*Greedystepwise* untuk memilih atribut yang sangat berpengaruh terhadap pendeteksian serangan untuk efisiensi. Hasil pengujian menunjukkan algoritma J48 menghasilkan akurasi tertinggi sebesar 99.839%.

Key Words:
Data mining;
Intrusion Detection System;
Cyberattacks;
Algoritma Machine Learning;
WEKA;

Riwayat Artikel:
Diserahkan 14 Maret, 2022
Direvisi 10 November, 2023
Diterima 10 November, 2023

DOI:
10.22441/incomtech.v13i3.15118

1. PENDAHULUAN

Keamanan merupakan aspek penting dari masalah internet (terutama jaringan komputer)[1]. Jaringan komputer harus dapat memastikan bahwa informasi atau *data* pribadi

tidak dapat diakses oleh penyusup (*attacker*)[2], sehingga memberikan rasa aman bagi akses pengguna[3]. Penyusup *(attacker)* dapat dicegah dengan membangun sebuah *intrusion detection system* (IDS). Untuk mendeteksi intrusi, sistem komputer atau jaringan harus terus dipantau dan dianalisis untuk kejadian yang dapat mengindikasikan pelanggaran atau upaya pelanggaran kebijakan keamanannya, kebijakan penggunaan yang dapat diterima, atau praktik keamanan normal yang dikenal sebagai deteksi intrusi [4]. Menurut [5], [6], [7] dan [8] Sebagai akibat dari jumlah data yang besar, deteksi anomali atau deteksi serangan menjadi sulit. Pengembangan aplikasi dan perangkat dipengaruhi oleh jumlah data yang tersedia untuk pengembang. Jumlah data yang dihasilkan karena dapat terhubung ke jaringan sangat besar. Salah satunya adalah kemampuan untuk mengidentifikasi jenis serangan tertentu yang sedang terjadi.

Browsing yaitu salah satu tindakan paling umum yang dilakukan orang setiap hari. Akses internet tersedia untuk anak-anak dan orang dewasa, berapa pun usia mereka [9]. Pengguna internet, di sisi lain, tidak menyadari bahwa internet dapat menjadi bahaya jika sistem keamanan jaringan diserang [10]. Sebuah IDS (*Intrusion Detection System*) diperlukan untuk mengawasi lalu lintas jaringan untuk setiap tanda-tanda aktivitas yang tidak biasa. Ketika ada banyak serangan yang masuk, IDS yang tidak dilengkapi dengan *machine learning* tidak dapat menanganinya dengan baik, yang dapat menyebabkan aktivitas normal di jaringan terlihat sebagai serangan dari peretas atau bisa jadi client biasa yang sedang mengakses server dianggap sebagai serangan. Acuan data pada IDS harus diperbaharui sesuai dengan perkembangan teknologi dan akan lebih baik jika dilengkapi dengan sistem deteksi serangan berbasis *machine learning*[5].

Istilah *data mining* mengacu pada tindakan menemukan pola dalam data yang dapat digunakan untuk menyimpulkan kesimpulan [11]. Penelitian yang dilakukan oleh [12] menggunakan dataset NSLKDD dengan *tools rapidminer*, Dengan akurasi 96.371 persen, algoritme C4.5 digunakan buat mengkategorikan data log dari sistem IDS, yang menunjukkan kalau model tersebut bisa dipakai untuk menentukan apakah suatu aktivitas merupakan bagian dari serangan siber. Namun menurut [13] dan [14] *dataset* NSLKDD merupakan lalu lintas normal dan serangan digabungkan bersama dalam lingkungan simulasi. Kumpulan data ini mempunyai sejumlah besar catatan yang berlebihan dan dipenuhi oleh kerusakan data yang menyebabkan hasil pengujian yang tidak tepat.

Pada penelitian [8] menggunakan *tools* WEKA dan dilakukan seleksi fitur *Information Gain, Gain Ratio, CFs-BestFirst* dan *CFs-PSO Search*. Namun penelitian tersebut hanya menggunakan 20% *dataset* CICIDS2017 dengan algoritma Naive Bayes, k-NN dan J48. Menghasilkan Algoritma J48 mempunyai akurasi tertinggi dengan menggunakan fitur seleksi *information gain* untuk memilih fitur dengan tingkat akurasi 99,81% karena mempengaruhi kinerja Naive Bayes, k-NN, dan algoritma lainnya. Hasil tersebut adalah dari 20% dataset yang digunakan, perlu dilakukan pengujian kembali menggunakan 100% serangan pada *dataset* CICIDS2017.

Seleksi fitur dan *tools* WEKA juga dilakukan pada penelitian [15] menggunakan dataset UNSW-NB15 dengan algoritma naïve bayes, pengukuran performa dilandasi dalam akurasi, presisi, *F-Measure* dan ROC Area. Hasil seleksi atribut dengan *Correlation-based Feature Subset* (CFS) *Selection*, dengan *search method Greedy Stepwise* meninggalkan 4 atribut didapatkan akurasi 74,8 %.

[16] menggunakan dataset KDDCUP'99 yang mana *dataset* tersebut ialah sebuah *dataset* yang diterbitkan oleh DARPA (*Defense Anvance Research Project Agency*) dalam tahun 1998. Teknik *Naive Bayes Classifiers, Random Forests*, dan SVM digunakan dalam kompetisi data mining dan eksplorasi ilmiah global yang diadakan oleh ACM SIGKDD (*Special Interest Group on Knowledge Discovery and Data Mining*). Selama percobaan ini, pengujian dilakukan untuk mengidentifikasi fitur KDD yang paling menonjol. Pada langkah pertama eksperimen ini, kami akan menerapkan teknik pemfilteran *At*tributeSelection untuk

pemilihan fitur guna menghapus fitur yang tidak perlu menggunakan evaluator *Correlation-based Feature Selection* (CfsSubsetEval). Alat *Select attributes* di aplikasi WEKA digunakan untuk langkah ini. Sebelum dan sesudah pemilihan fitur, algoritma Random Forest penelitian ini memiliki tingkat akurasi 98 persen.

Penelitian yang dilakukan oleh [17] menggunakan *dataset* ISCX2012. *Dataset Information Security Center of eXcellence (ISCX),* Universitas New Brunswick, Fakultas Ilmu Komputer yang mengembangkan *dataset* tersebut dari 2009 hingga *2012. Dataset* ISCX2012 dijelaskan dalam penelitian oleh (Shiravi, Tavallaee, dan Ghorbani, 2011), yang mencakup 157.867 paket dengan 19 fitur dikumpulkan selama 7 hari aktivitas jaringan (yaitu normal dan intrusi). [17] Menggunakan bahasa pemrograman *python* dengan menggunakan *library* bernama *scikit-learn* yang menghasilkan algoritma *Naïve Bayes*, *SVM Linear*, *SVM Polynomial*, dan *SVM Sigmoid* tanpa seleksi fitur. Akurasi tertinggi diperoleh *SVM Polynomial* yaitu 99,999%. Menurut [14] *dataset* ISCX2012 (University of New Brunswick 2012) ini memiliki dua *profil*, *Alpha-profile* yang melakukan berbagai skenario serangan multi-tahap, dan *Beta-profile*, yang merupakan pembangkit lalu lintas normal dan menghasilkan lalu lintas jaringan yang realistis dengan lalu lintas serangan. Ini termasuk lalu lintas jaringan untuk protokol HTTP, SMTP, SSH, IMAP, POP3, dan FTP dengan muatan paket penuh. Namun, ini tidak mewakili protokol jaringan baru karena hampir 70% lalu lintas jaringan saat ini adalah HTTPS dan tidak ada jejak HTTPS dalam kumpulan data ini. Selain itu, distribusi serangan simulasi tidak didasarkan pada statistik dunia nyata.

Dimungkinkan untuk meningkatkan metode klasifikasi untuk mendeteksi serangan dengan menggunakan pendekatan pemilihan fitur yang efisien. *Dataset* yang diuji pun harus data yang terbaru dan berdasarkan statistik dunia nyata. Lebih dari beberapa proyek penelitian telah meneliti berbagai metode dan taktik untuk menentukan kualitas mana yang paling berguna. Berdasarkan rujukan pada masing-masing riset diatas, maka dalam penelitian ini akan dilakukan teknik *Correlation-based Feature Subset* (CFS) *Selection*, dengan search method *Greedy Stepwise* pada dataset UNSW-NB15 dan CICIDS2017 menggunakan algoritma Naïve Bayes, J48, AdaBoostM1 guna memperoleh fitur algoritma terbaik dan relevan yang berpartisipasi dalam performa sistem deteksi. Pemilihan algoritma klasifikasi didasarkan atas beberapa pertimbangan, algoritma AdaBoost.M1 yang diusulkan sebagai turunan dari AdaBoost untuk memecahkan masalah multi-klasifikasi [18], Dengan asumsi independen dan kategorisasi selalu benar, Naive Bayes adalah algoritma berbasis probabilitas dasar Teknik klasifikasi *decision tree* lainnya diungguli oleh algoritma j48 [8]. Selain itu penggunaan ketiga algoritma tersebut dimaksudkan untuk melihat perbandingan tingkat akurasi dan memilih algoritma dengan performansi tertinggi dalam mendeteksi serangan.

2. METODE

2.1 RANCANGAN PENELITIAN

1. Model yang dipakai pada riset ini didasarkan pada proses *Knowledge Discovery in Databases* (KDD) pada berbagai fase [19]. Gambar 1 menggambarkan setiap tahapan yang akan diselesaikan.
   Selecting

UNSW-NB15 adalah *dataset public* yang ada sejak tahun 2015 pembuatan *dataset* oleh [20] didasari tidak tersedianya kumpulan data berbasis jaringan yang komprehensif yang dapat mencerminkan skenario lalu lintas jaringan modern. CICIDS2017 merupakan *dataset public Canadian Institute for Cybersecurity* yang dibuat tahun 2017, Berdasarkan penelitian [14] lebih dari sebelas dataset yang tersedia sejak tahun 1998 sudah ketinggalan zaman dan tidak dapat diandalkan untuk digunakan. Beberapa dari kumpulan data ini kurangnya

keragaman serangan dan beberapa di antaranya tidak mencakup berbagai serangan, sementara yang lain dapat dikatakan informasi paket dan muatan yang tidak dapat mencerminkan tren saat ini, atau tidak memiliki kumpulan fitur dan *metadata*.

Dari 2 penelitian tersebut peneliti memutuskan memakai 2 *dataset* ini sebagai penelitian karena data yang tersedia sudah terbaru dan modern.

1.1.1 Preprocessing

Pada *dataset* terpilih akan dilakukan *Correlation-Based Feature Selection* (CFS) adalah tahapan *preprocessing dataset* untuk memilih keterkaitan data dengan class yang ingin diketahui, yang terpilih adalah nilai korelasi yang tinggi pada atribut yang terpilih yang akan diolah selanjutnya menggunakan algoritma. Pada proses ini akan terlihat atribut pada kedua dataset yang memiliki nilai 100% akan diambil untuk selanjutnya diproses menggunakan algoritma.

1.1.2 Transformation

Fitur yang sudah dipilih kemudian disimpan dengan file yang berbeda untuk memudahkan memilih dataset yang sudah melewati tahapan *preprocessing* dengan *Correlation-Based Feature Selection* (CFS).

1.1.3 Machine learning

Metode klasifikasi J48, Naive Bayes, dan AdaboostM1 digunakan untuk mengkategorikan IDS dalam pemodelan data. Yang mana masing-masing *dataset* yang diuji dengan atribut terpilih akan menghasilkan akurasi pada setiap algoritma.

1.1.4 Interpretation / Evaluation

Hasil evaluasi pada proses diatas menghasilkan ketiga algoritma menghasilkan akurasi yang berbeda. Evaluasi dan validasi dari hasil klasifikasi menggunakan *Confusion Matrix*.

Confusion Matrix

Teknik penilaian yang umum dilakukan dalam riset *supervised learning* adalah *confusion matrix*[21][22][23], yang mana metode evaluasi ini menghasilkan dua kelas ialah kelas positif dan kelas negatif [24]. Dasar perhitungan pada metode ini tergambarkan dalam tabel 3.1 berikut ini: :

Tabel 3.1 Matrik Konfusi

		Prediksi	
		a	b
Aktual	a	TP	FP
	b	FN	TN

Penjelasan pada tabel 3.1 yaitu :

***True Positive* (TP)**

Merupakan data positif yang diprediksi benar.

***True Negative* (TN)**

Merupakan data negatif yang diprediksi benar.

***False Postive* (FP)**

Merupakan data negatif namun diprediksi sebagai data positif.

***False Negative* (FN)**

Merupakan data positif namun diprediksi sebagai data negatif.

Penelitian ini melakukan klasifikasi pada dataset, pengukuran yang biasa digunakan adalah akurasi, *precision* dan *recall*. Alhasil *precision, recall* dan *accuracy* bisa diukur dengan formula ialah [8] :

a. Precision

Dalam konteks ini, *precision* didefinisikan sebagai perkiraan probabilitas bahwa ramalan positif itu akurat. Rumus *precision* adalah :

$$Precision = \frac{TP}{TP+FP}$$

b. Recall

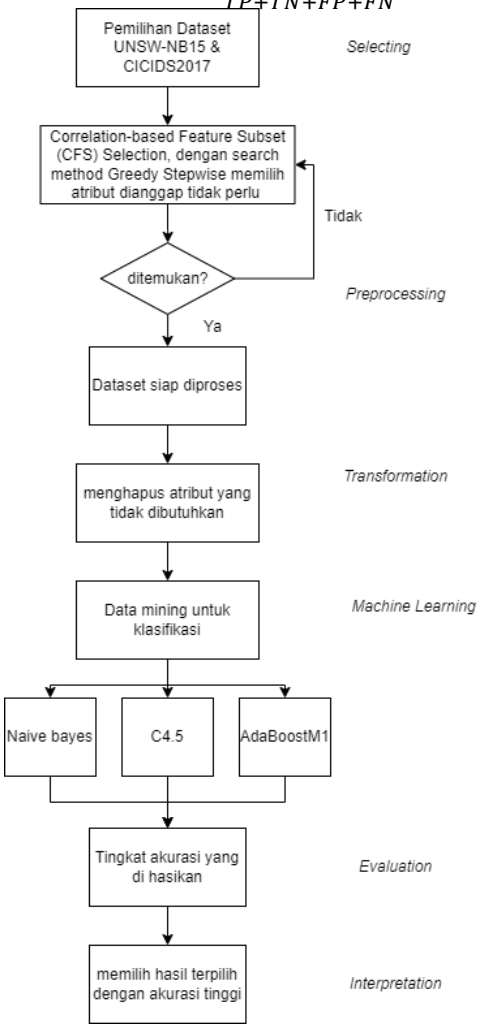
Recall didefinisikan sebagai kelas positif yang sebenarnya adalah kelas positif. Rumus *recall* yaitu :

$$recall = \frac{TP}{TP+FN}$$

c. Accuracy

Accuracy nilai klasifikasi ditentukan oleh seberapa dekat nilainya dengan nilai sebenarnya. Rumus *accuracy* ialah:

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$



Gambar 1. Tahapan penelitian

Annotation step adopts [13] method, scraping data from Zomato and was annotated by 3 reviewers and constructed with three classes such as positive, neutral, and negative with 14740 rows. All of labelled aspect data was aggregated with voting. If voting percentage resulted more than 66,67% then the data is used for experiment.

Data annotation was done manually with four subject annotations as follows: price, food, place, and service. Data was also labelled by three polarity such as: positive, neutral, negative.

Proposed method started with vectorizing sentence using attention model. The result of attention model was used for LSTM and CNN method as shown in Fig. 4. CNN layer used method as [4] proposed with five hyperparameter. This research related to [8] uses the convolutional neural network (CNN) when comparing the linguistic pattern (LP) method and the combination of CNN with LP, which has 7 layers (1 input layer, 2 convolution layers, 2 max grouping, 1 fully connected, 1 output layer). After this step performance and accuracy was evaluated with F-measure.

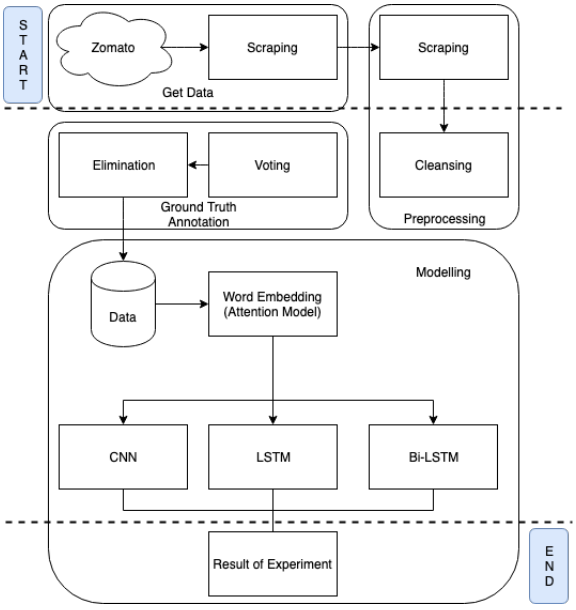


Figure 1. Research Workflow.

In machine learning modelling phase, this experiment stores every information weight of text using word embedding that processed by attention model, and then test to proposed methods such as CNN, LSTM, Bi-LSTM, compared to type of deep learning model combined with attention model.

Features inputted using 60-word inputs, method evaluation aimed for accuracy. Accuracy of all methods used F-measure training from epoch 1 until 200 with 70% training data and 30% testing data. CNN had 8 layers with one input layer, two convolutional layers, two max-pooling layers, one flatten-layer, one fully connected and one output layer. LSTM and Bi-LSTM had 128 units. These methods used ReLu activation function.

3. RESULTS AND DISCUSSION

The data was taken from Zomato by scraping the rating and review columns. In addition to the rating column and the revision, the "id" column is formed to facilitate processing, but the id starts from 0 as shown in Table 1.

Table 1. Scraped data from Zomato

id	rating	Review
0	4.0	pas masuk selera saya sate kambing nya lembut dan gurih. sop buntut nya empuk (piring nya kecil bener), bakso segar dan mantabnya banyak kerupuk (buat saya yg fanatik kerupuk sangat luar biasa) sashimi nya mantab walaupun pecking duck nya kurang pas buat saya tapi dessert-nya paling pas.
1	5.0	Makanan ini sangat enak, tempatnya bagus, sajiannya juga luar biasa.

Based on the data, a voting system was carried out on the data. It was found that each selected data had the same number, although a sentence could have more than one aspect. The amount of data that has been tagged in the voting system, among others, Food is 71371 lines, Price is 13777 lines, Place is 19424 lines and Service is 5837 lines.

Once the voting completed, data elimination is done if the sentence has more than one aspect, for example, "This meal is very good, the place is good, the dish is also extraordinary," which has aspects of food and place. Data elimination was also conducted if the aspects do not have the same minimum of two people. During elimination, the data also determined sentence polarity which divided into three categories namely negative, neutral, and positive. Elimination resulted in data reduction and the sentence polarity assignment can be seen in Table 2.

Table 2. Aspect Labelled Percent Agreement

Aspect	**Number of Rows**	**Negative**	**Neutral**	**Positive**
Food	56224	4315	10099	41810
Price	5644	496	985	4163
Place	12160	920	2201	9039
Service	3113	530	584	1999

As seen in Figure 2, the dataset is not balanced. In order to make the dataset more balanced, before continuing into model training, dataset will be balanced by SMOTE method.

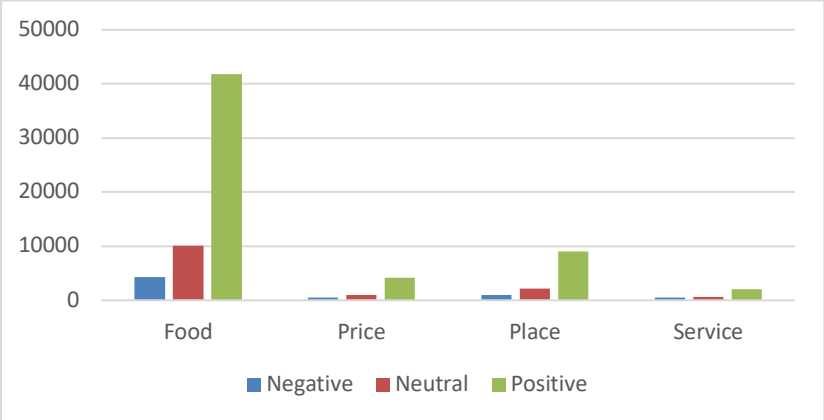


Figure 2. The comparation total of processed data

Graphs in figure 3 show fluctuation in results between the usage of CNN, LSTM, and Bi-LSTM with and without attention model. The highest increase of 0.05% can be seen for LSTM with attention model. Despite the increase with LSTM, the Bi-LSTM method experienced a slight decrease by 1%. However, the best method without using attention model is observed to be Bi-LSTM has gap 2% compared to LSTM. LSTM also proved to be the best method with attention model, having a significant average accuracy difference compared to LSTM by 88%.

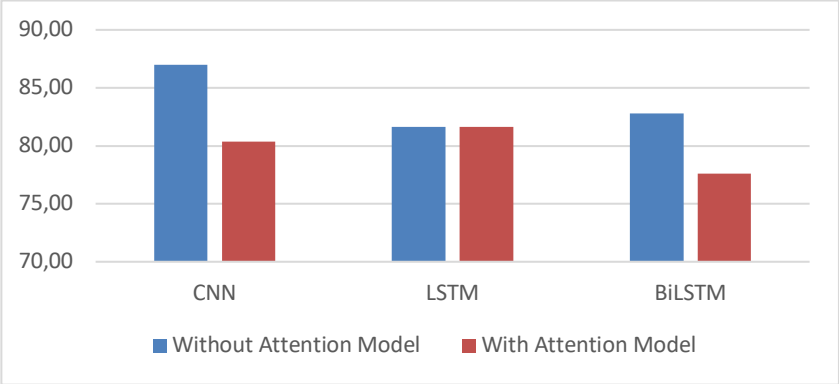


Figure 3. Accuracy of training models

As seen in Figure 4, it was found that the average loss increased and decreased between the attention model than before, with the highest increase that occurred in CNN from 1.0779 to 2.8542. The largest decline occurred in LSTM, decreasing from 1.0942 to 0.875655, while major loss didn't occur for Bi-LSTM that showed decrease only from 0.8415 to 0.8288. Figure 7 also illustrates that the method with the lowest loss before using the attention model was the LSTM showing value of 0.8415, while the highest loss occurred in LSTM by 1.0942. After the use of attention model, the lowest loss remained in Bi-LSTM with the final result of 0.8288, while CNN held the highest loss with 2.8542.

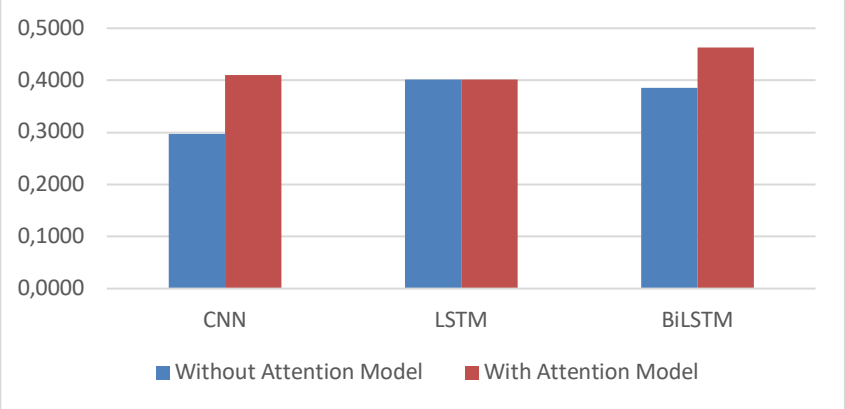


Figure 4. Average validation loss of training models

From Table 3, the best evaluation model is found in the LSTM with an accuracy value of 75.9%, followed by Bi-LSTM with attention model with accuracy value of 79.35%. But the smallest loss was obtained by CNN with attention with loss value of 0.58 followed by Bi-LSTM with attention model of 0.60. Convergent

between machine learning model has slightly different between CNN with attention model than others.

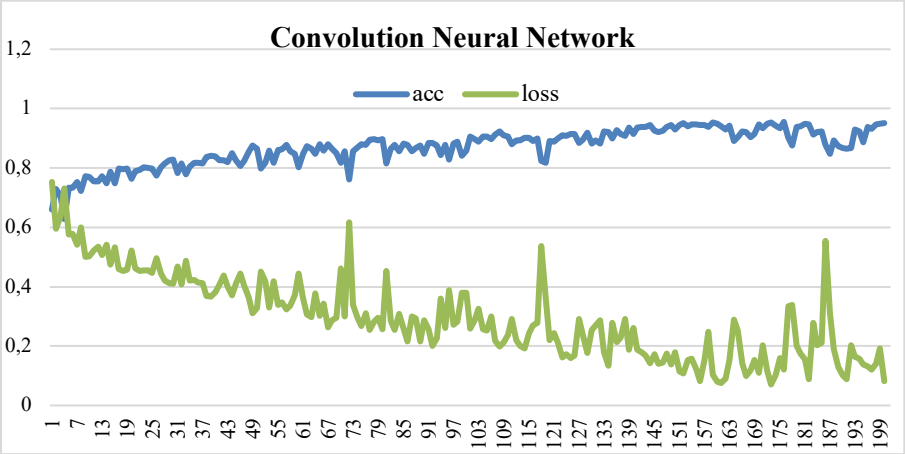


Figure 5. CNN without attention model training accuracy and loss

Meanwhile CNN method as can be seen in figure 5, has no convergent from start training until epoch iteration finished. Compared to figure 6, with attention model CNN method can get convergence at 183 epochs. This case not be found on LSTM and Bi-LSTM with or without attention model.

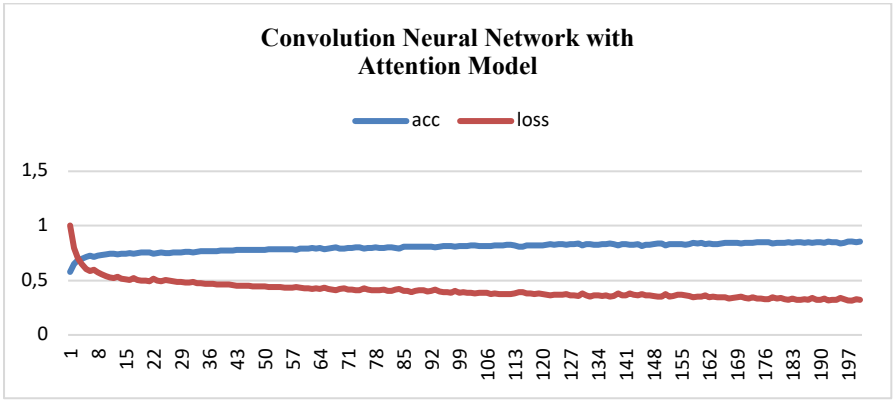


Figure 6. CNN without attention model training accuracy and loss

Table 3. Accuracy and Loss Evaluation Models

Model	**Loss**	**Accuracy (%)**	***Convergent***
CNN	1.12	70.00	-
CNN Attention Model	0.58	77.48	183
LSTM	0.77	79.35	186
LSTM Attention Model	2.03	77.19	186
Bi-LSTM	0.63	71.64	186
Bi-LSTM Attention Model	0.60	77.51	186

Evaluation models that can be seen in Figure 7 showed increases and decreases in accuracy. The decrease in accuracy occurred in CNN by 5.1%, while the highest increase in accuracy of 4.9% can be seen in Bi-LSTM. Before using attention model, the best method was shown to be CNN while Bi-LSTM was displayed to be the method with worst accuracy. After attention model was implemented, the method with the best precision turned out to be Bi-LSTM with CNN being the least accurate method.

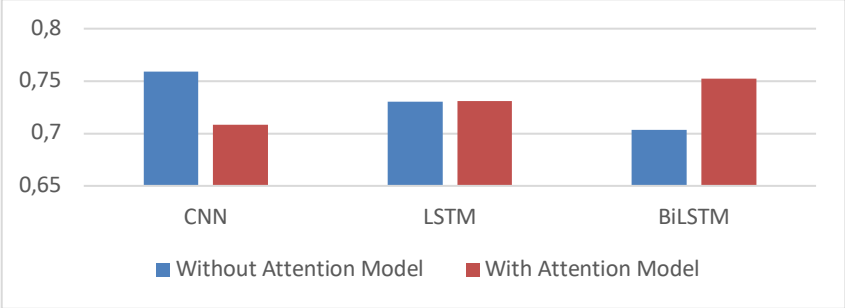


Figure 7. Accuracy of evaluation models

Based on the results of the loss value evaluation model that can be seen in Figure 8, it can be found that there was an increase and a decrease between the use of attention models than before. The increase in loss occurred in CNN from 1.1 to 2.5, while the largest decrease occurred in Bi-LSTM from prior value of 1.12 to 0.60. The method with the least loss before using the attention model was CNN, while the method with the highest loss was Bi-LSTM. After implementing attention model, the method with the least loss was Bi-LSTM, while the highest loss result can be seen in CNN method.

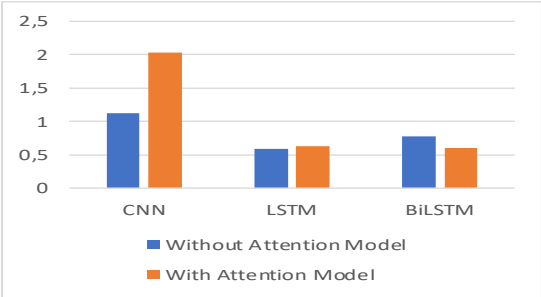


Figure 8. Validation loss of evaluation models

In Figure 9 of the averages of existing steps, it was found that there was a slowdown with the use of attention model compared to without attention model. The lowest deceleration occurred in CNN method where the difference was 900.7 µs, while the highest deceleration occurred in LSTM of 1100 µs. Data observation after using attention model showed that the fastest method is CNN, while the slowest method is Bi-LSTM with an average time of 2000 µs/step. The fastest method after using the attention model is CNN with 998,857,143 µs/step, while the slowest method was Bi-LSTM with attention model with an average time of 300 µs/step.

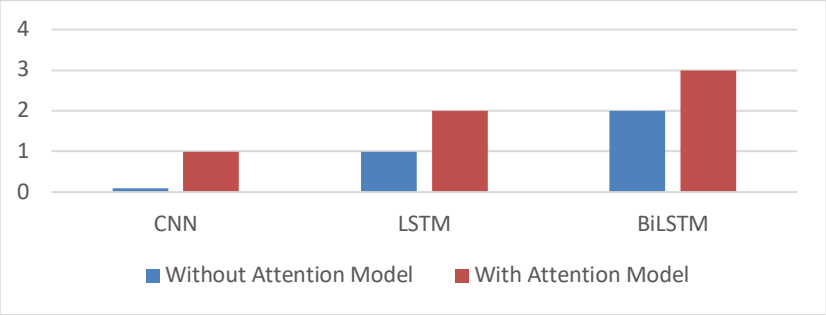


Figure 9. Average Step over Training with 3 Models

Figure 10 showed that there was a slowdown in average processing time after the use of attention model compared to without attention model. The method with the best deceleration was LSTM with a time of 122.18 seconds compared to BiLSTM with a time of 180.75 seconds. Without attention model, the fastest method was CNN with the average time of 15.6 s, while the slowest method was BiLSTM with average time 311.55 s. The fastest method with attention model was CNN with average time of 157.52 s, while the slowest method was BiLSTM with attention model, consuming average processing time of 492.6 s.

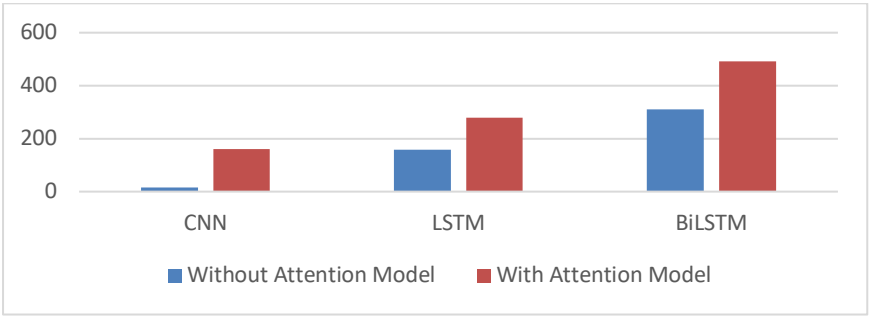


Figure 10. Average Processed over Training with 3 Models

4. CONCLUSION

Based on this study, it can be concluded that attention models relative improved stability model and has trade-off for accuracy because of noise data filtered by it. It was also found that the best accuracy performance after adapting attention model occurred in LSTM method. Although attention model can improve accuracy and loss, but it resulted in deceleration of processing time. Model can adapt with typo and microtext for classification. In the future, there must be comparative method between transformer methods with neural network methods to get insights which model more stable and accurate to do multi-classification sentiment analysis.

REFERENSI

[1] J. Barnes, R. Klinger, and S. S. im Walde, "Assessing State-of-the-Art Sentiment Models on State-of-the-Art Sentiment Datasets," pp. 2–12, 2017.

[2] L. Xu, J. Lin, L. Wang, C. Yin, and J. Wang, "Deep Convolutional Neural Network based Approach for Aspect-based Sentiment Analysis," Adv. Sci. Technol. Lett., vol. 143, pp. 199–204, 2017.

[3] G. Vinodhini and R. Chandrasekaran, "Sentiment Analysis and Opinion Mining: A Survey,"

Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 2, no. 6, pp. 282–292, 2012.
[4] I. Goodfellow, "Deep Learning."
[5] S. Poria, E. Cambria, and A. Gelbukh, "Aspect Extraction for Opinion Miningwith a Deep Convolutional Neural Network," Knowledge-Based Syst., vol. 108, pp. 42–49, 2016.
[6] A. M. Ramadhani and H. S. Goo, "Twitter sentiment analysis using deep learning methods," in 2017 7th International Annual Engineering Seminar (InAES), 2017, pp. 1–4.
[7] S. M. Jiménez-Zafra, E. Martínez-Cámara, M. T. Martín-Valdivia, and L. A. Ur Na-López, "SINAI: Syntactic approach for Aspect Based Sentiment Analysis," Semeval 2015, no. SemEval, pp. 730–735, 2015.
[8] X. Zhu, P. Sobhani, and H. Guo, "Long Short-Term Memory Over Tree Structures," Int. Conf. Mach. Learn., no. Icml, Mar. 2015.
[9] W. Che, Y. Zhao, H. Guo, Z. Su, and T. Liu, "Sentence Compression for Aspect-Based Sentiment Analysis," IEEE/ACM Trans. Audio Speech Lang. Process., vol. 23, no. 12, pp. 2111–2124, 2015.
[10] D. Tang, F. Wei, B. Qin, M. Zhou, and T. Liu, "Building Large-Scale Twitter-Specific Sentiment Lexicon: a Representation Learning Approach," Proc. 25th Int. Conf. Comput. Linguist. (COLING 2014), pp. 172–182, 2014.
[11] H. Lakkaraju, R. Socher, and C. D. Manning, "Aspect Specific Sentiment Analysis using Hierarchical Deep Learning," NIPS WS Deep neural networks Represent. Learn., pp. 1–9, 2014.
[12] L. Zhang and B. Liu, Data Mining and Knowledge Discovery for Big Data, vol. 1. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.
[13] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis," Proc. 2016 Conf. Empir. Methods Nat. Lang. Process., pp. 616–626, 2016.
[14] Bobicev, V., & Sokolova, M. (2017). Inter-Annotator Agreement in Sentiment Analysis: Machine Learning Perspective. Proceedings of Recent Advances in Natural Language Processing, 97–102. https://doi.org/10.26615/978-954-452-049-6_015
[15] Cambria, E., Gelbukh, A., & Thelwall, M. (2017). Affective Computing and Sentiment Analysis. IEEE INTELLIGENT SYSTEMS, 32(6), 74–80. https://doi.org/10.1109/MIS.2017.4531228
[16] Covington, M. A. (1994). Natural Language Processing for Prolog Programmers. New Jersey: Prentice Hall.
[17] Kao, A., & Poteet, S. R. (2007). Natural Language Processing and Text Mining. USA: Springer.
[18] Kemp, S. (2017, August 10). Three Billion People Now Use Social Media. Retrieved from https://wearesocial.com/blog/2017/08/three-billion-people-now-use-social-media.
[19] Lau, J. H., & Baldwin, T. (2016). An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proceedings of the 1st Workshop on Representation Learning for NLP (pp. 78–86). Stroudsburg, PA, USA: Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-1609
[20] Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal, 5(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011
[21] Mitchell, T. M. (1997). Machine Learning. New York: McGraw Hill.
[22] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1–2), 1–135.
[23] Poria, S., Cambria, E., & Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. Knowledge-Based Systems, 108, 42–49. https://doi.org/10.14257/astl.2017.143.41.