

Bidirectional GRU dengan Attention Mechanism pada Analisis Sentimen PLN Mobile

Bidirectional GRU with Attention Mechanism on Sentiment Analysis of PLN Mobile

Moh. Ainur Rohman¹, Suhartono², Totok Chamidy³

^{1,2,3}Magister Informatika, Universitas Islam Negeri Maulana Malik Ibrahim Malang

E-mail: ¹210605210005@student.uin-malang.ac.id, ²suhartono@ti.uin-malang.ac.id,

³to2k2013@ti.uin-malang.ac.id

Abstrak

PLN Mobile adalah aplikasi ponsel *customer self-service* yang terintegrasi dengan Aplikasi Pengaduan dan Keluhan Pelanggan (APKT) dan Aplikasi Pelayanan Pelanggan Terpusat (AP2T). Mulai awal tahun 2021 sampai sekarang PLN mengencangkan sosialisasi PLN Mobile pada masyarakat sehingga jumlah ulasan PLN Mobile pada google playstore meningkat drastis. Untuk mengetahui kepuasan pelanggan tidak bisa hanya dengan melihat dan menganalisis dari kolom ulasan PLN Mobile di google playstore, hal ini dikarenakan data ulasan berbentuk tidak terstruktur. Untuk mengatasi masalah ini dibutuhkan teknik khusus yaitu analisis sentimen. Penelitian ini bertujuan untuk mengusulkan arsitektur analisis sentimen untuk mengatasi ketidakmampuan algoritma *deep learning* seperti LSTM dan GRU dalam menangkap informasi penting. Arsitektur yang diusulkan yaitu mengkombinasikan *Bidirectional GRU* (BiGRU) dengan *attention mechanism* menggunakan *word2vec* sebagai *word embedding*. *Attention mechanism* digunakan untuk menangkap kata yang penting sehingga arsitektur tersebut dapat memahami informasi yang penting. Kemudian, arsitektur yang diusulkan dilakukan perbandingan dengan metode CNN, CNN-GRU, CNN-LSTM, CNN-BiGRU, CNN-BiLSTM dengan menggunakan data ulasan PLN Mobile. Hasil eksperimen menunjukkan bahwa arsitektur analisis sentimen yang diusulkan memiliki akurasi dan f1-score yang lebih tinggi.

Kata kunci: Analisis Sentimen, *Attention Mechanism*, BiGRU, PLN Mobile, Word2vec

Abstract

PLN Mobile is a self-service customer mobile application integrated with the Customer Complaints and Complaints Application (APKT) and the Centralized Customer Service Application (AP2T). From early 2021 until now, PLN has intensified the socialization of PLN Mobile to the public so that the number of PLN Mobile reviews on the Google Play Store has increased drastically. Finding out customer satisfaction cannot only be done by looking at and analyzing the PLN Mobile review column on the google play store; this is because the review data is unstructured. To address this problem, a unique technique is needed, namely sentiment analysis. This research aims to propose a sentiment analysis architecture to overcome the inability of deep learning algorithms such as LSTM and GRU to capture important information. The proposed architecture combines Bidirectional GRU (BiGRU) with an attention mechanism using word2vec as word embedding. Attention mechanisms capture important words so that the architecture can understand the information that matters. Then, the proposed architecture was compared with the CNN, CNN-GRU, CNN-LSTM, CNN-BiGRU, and CNN-BiLSTM methods using PLN Mobile review data. The experimental results show that the proposed sentiment analysis architecture has higher accuracy and f1-score.

Keywords: *attention mechanism, BiGRU, PLN Mobile, sentiment analysis, word2vec*

1. PENDAHULUAN

PLN Mobile adalah aplikasi ponsel *customer self-service* yang terintegrasi dengan Aplikasi Pengaduan dan Keluhan Pelanggan (APKT) dan Aplikasi Pelayanan Pelanggan Terpusat (AP2T). Aplikasi ini menyediakan data terpadu pelanggan dengan tingkat akurasi, keamanan data dan validasi yang dapat dipertanggungjawabkan. Mulai awal tahun 2021 sampai sekarang PLN mengencarkan sosialisasi PLN Mobile pada masyarakat, alhasil jumlah ulasan PLN Mobile pada google playstore meningkat drastis dibanding tahun-tahun sebelumnya. Namun, untuk mengetahui kepuasan dan loyalitas pelanggan, psikologi pelanggan, mana sisi baik dan tidak dari aplikasi tidak bisa hanya melihat dan menganalisis dari kolom ulasan PLN Mobile, hal ini dikarenakan data ulasan berbentuk *unstructured* (tidak terstruktur). Untuk mengatasi masalah ini dibutuhkan teknik khusus, yaitu analisis sentimen. Dengan menerapkan analisis sentimen pada PLN Mobile dapat membantu pihak manajemen untuk mengukur keberhasilan mereka dalam mempromosikan dan mensosialisasikan PLN Mobile juga digunakan untuk evaluasi dan membuat *roadmap* kedepan berdasarkan hasil analisis sentimen.

Analisis sentimen adalah ilmu yang digunakan untuk menganalisis pendapat, sentimen, sikap dan emosi orang terhadap entitas seperti layanan, produk, organisasi, peristiwa dan lain-lain [1]. Pada ranah bisnis, analisis sentimen memainkan peran penting untuk mendapatkan *insight* dari *feedback* pelanggan untuk meningkatkan strategi dan produk mereka. Saat ini budaya bisnis menggunakan prinsip '*customer-based*' ini artinya pelanggan dijadikan sebuah mercusuar untuk *role-model* bisnis mereka, oleh karena itu memahami pelanggan adalah hal yang terpenting [2].

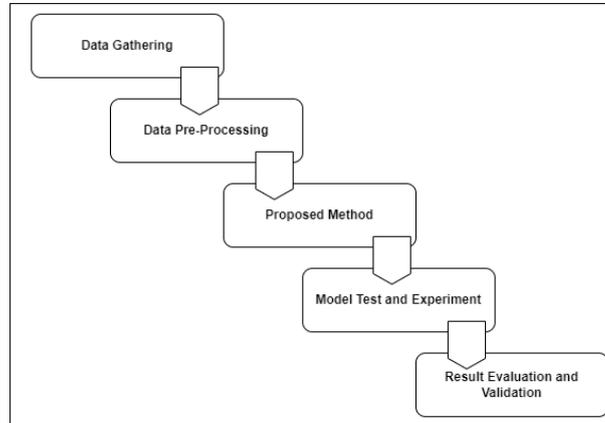
Beberapa penelitian terkait analisis sentimen telah diusulkan. Pada penelitian [3] mengusulkan model CNN-BiLSTM, model ini mengkombinasikan CNN dan *Bidirectional LSTM* untuk *document-level* analisis sentimen dan untuk *word embedding* menggunakan jenis *Doc2vec*. Hasil dari penelitian ini mencapai akurasi sebesar 90.66%. Penelitian [4] mengusulkan model kombinasi antara GRU dan LSTM dengan *word embedding* menggunakan GloVe, penelitian ini bertujuan untuk mengatasi kekurangan pada RNN yaitu tidak bisa mempelajari *long-term information* pada teks. Hasil dari penelitian ini mencapai akurasi sebesar 87.10%. Penelitian [5] mengusulkan model kombinasi *word vector* dengan *sentiment information* dan TF-IDF, untuk algoritma *sequence learning*-nya menggunakan Bi-LSTM. Hasil dari penelitian ini mencapai F1-Score 92.18%.

Berdasarkan penelitian [3]–[5] algoritma *deep learning* telah berhasil dalam menghasilkan *context information* dalam *sequence learning*. Namun, algoritma *deep learning* tidak dapat memfokuskan kata-kata yang paling berkontribusi, akibatnya tidak mampu menangkap informasi penting pada *context information*. Untuk mengatasi masalah ini, penulis mengusulkan arsitektur yang diberi nama “Glo-att-bigru+word2vec” yang merupakan singkatan dari (*Bidirectional GRU* menggunakan *Global Attention* dan *Word2vec*). Arsitektur ini menggunakan *word2vec* untuk *word embedding*, *Bidirectional GRU* (BiGRU) untuk *sequence learning*. GRU dipilih karena dapat mengatasi *vanishing* dan *exploding gradient* pada RNN dan lebih sederhana dari *Long-Short Term Memory* (LSTM) [4] sedangkan Bi-GRU merupakan kombinasi dari dua *hidden layer* yaitu *forward* dan *backward*, dengan begini dapat mengakses konteks sebelumnya dan selanjutnya. *Global attention* digunakan untuk menangkap informasi penting dari *context information*.

Ide dasar dari arsitektur yang diusulkan adalah, *Global attention* digunakan untuk memberikan fokus (*attend*) yang berbeda dari *context information* yang dihasilkan oleh Bi-GRU dengan cara memberikan *weight* (bobot) yang berbeda ke dalam *variable-length* pada *sequence learning*. Kemudian fitur yang diekstraksi dari *global attention* digabungkan dan diklasifikasikan menggunakan *softmax classifier* sehingga model dapat menangkap informasi penting dari *sequence*. Model yang diusulkan akan diujikan menggunakan *dataset* ulasan PLN Mobile, kemudian dilakukan *tuning hyperparameter* untuk dibandingkan parameter terbaik dan membandingkan dengan model analisis sentimen yang lain untuk evaluasi model.

2. METODE PENELITIAN

Metode penelitian yang digunakan adalah metode eksperimen dengan tahapan penelitian seperti pada gambar 1.



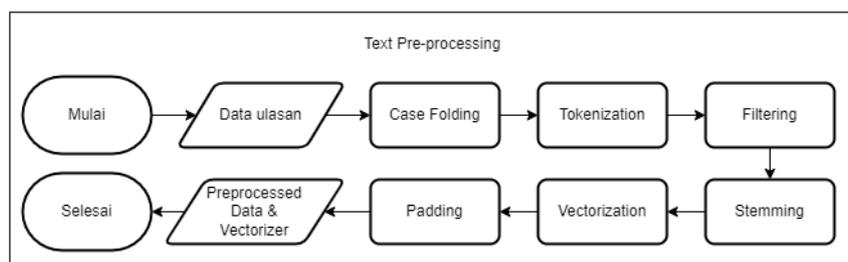
Gambar 1. Desain penelitian

1.1 Data Gathering

Metode pengumpulan data menggunakan metode *scrapping*. Sumber data didapat dari ulasan aplikasi PLN Mobile di google play store. Atribut atau fitur yang dibutuhkan hanya dua, yakni: ulasan dan sentimen. Atribut ulasan berisi tentang ulasan pelanggan terhadap aplikasi PLN Mobile dan atribut sentimen berisi tentang penilaian pengguna terhadap aplikasi. Atribut sentimen hanya terdapat dua label, yakni positif dan negatif. Untuk label netral dihapus karena menurut [1] pada bukunya, label netral tidak terdapat sentimen atau opini, dan sifatnya objektifitas.

1.2 Data Pre-Processing

Penelitian ini menggunakan data teks, oleh karena itu proses *data pre-processing* dilakukan dengan teknik *text pre-processing*. Umumnya *text preparation* dan *pre-processing* dilakukan pada penelitian *Natural Language Processing* (NLP). Tujuan dilakukan *text pre-processing* pada penelitian ini untuk menstandarisasi data, mengurangi jumlah dimensi dalam *dataset*, dan mengurangi kompleksitas akan tetapi dengan informasi yang sama. Ada enam tahap *text pre-processing* pada penelitian ini seperti pada gambar 2.



Gambar 2. Text pre-processing

a. Case folding

Proses ini untuk mentransformasikan semua huruf *dataset* ulasan PLN Mobile menjadi huruf kecil, dan menghapus semua karakter selain huruf atau dianggap delimiter. Tujuan *case folding* untuk menyamakan istilah dan konteks.

b. Tokenization

Setelah ulasan ditransformasikan ke *case folding*. Kalimat dijadikan token atau memecah dokumen/paragraf menjadi unit lebih kecil agar lebih mudah dipahami oleh mesin. Karena bagi mesin sulit ketika tidak dilakukan tokenization.

c. *Filtering*

Setelah proses *tokenization*, data ulasan PLN Mobile dihilangkan kata-kata yang tidak mengandung makna atau arti, atau dalam bidang NLP istilahnya *stop word list*. Kata-kata yang dibuang seperti “di”, “ke”, “dari”, “yang”, dan lain-lain. Tujuannya adalah untuk mereduksi data sehingga kompleksitas data berkurang.

d. *Stemming*

Setelah melakukan proses *filtering*, data ulasan diproses *stemming*. *Stemming* sendiri adalah proses mencari kata dasar dengan menghilangkan *prefix* dan *suffix*. Data ulasan dicari semua kata dasarnya untuk meningkatkan akurasi temu kembali informasi (*retrieval information*).

e. *Vectorization*

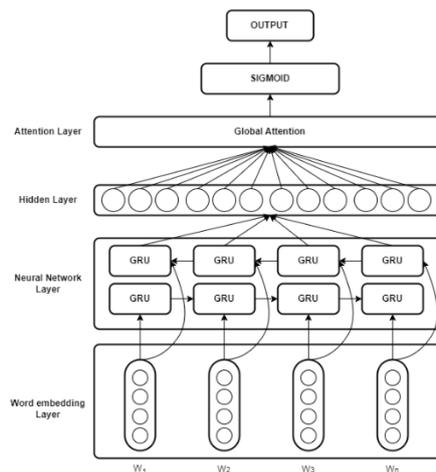
Pada penelitian ini data ulasan PLN Mobile semuanya berbentuk teks, sedangkan komputer tidak mengenal karakter selain angka, komputer hanya menerima *input* angka. Setelah data dilakukan proses *stemming*, token-token pada dokumen/paragraf ditransformasikan menjadi sebuah bilangan (vektor), agar model *deep learning* mengenali dataset.

f. *Padding*

Pada data ulasan PLN Mobile, bila dijadikan matrik masing-masing dokumen, maka akan terlihat bentuk (panjang) matrik tidak memiliki ukuran yang sama. Sedangkan ketika bekerja dengan *neural network* harus memiliki *input* dengan bentuk dan ukuran yang sama. Pada penelitian ini data ulasan dilakukan proses *padding* agar memiliki *input* teks dengan ukuran yang sama.

1.3 *Proposed Method*

Pada bagian ini, peneliti memaparkan usulan metode secara rinci. Pada penelitian ini penulis mengusulkan *attention mechanism* yang digabungkan dengan Bi-GRU. Arsitektur proposed method ditunjukkan pada gambar 3.

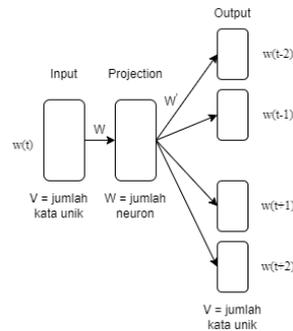


Gambar 3. Arsitektur “Glo-att-bigru+word2vec”

a. *Word Embedding*

Tradisional *text representation*, seperti *Bag of Words*, terdapat dua masalah, yaitu: kehilangan *learning representation* dan ukuran *dimensionality* yang besar. Untuk mengatasi masalah ini, penelitian ini menggunakan *word embedding*, untuk merepresentasikan setiap kata yang mirip secara semantik. *Word embedding* yang

digunakan adalah *word2vec* yang diusulkan oleh Mikolove [6]. Arsitektur yang digunakan dari *word2vec* adalah skip-gram. Model dilatih dengan memaksimalkan *average log probability* di semua kata.



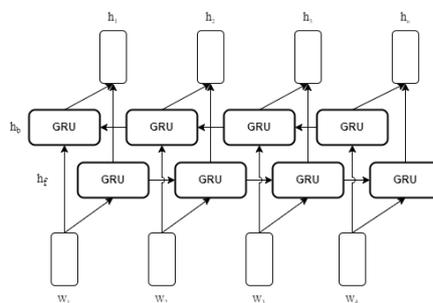
Gambar 4. Arsitektur skip-gram

Pada gambar 4 arsitektur skip-gram memberikan bobot pada W dan W' secara acak. W dan W' merupakan matrik dengan ukuran $W = V * N$ dan $W' = N * V$. Pada tahap *feedforward*, input di *dot product* dengan W dan menghasilkan nilai pada *hidden layer* (*projection*). Kemudian *hidden layer* di *dot product* dengan W' dan menghasilkan nilai *output*, setelah itu akan dihitung nilai errornya menggunakan *cross entropy*. Pada tahap *backpropagation*, menggunakan teknik *gradient descent* yaitu dengan melakukan *upgrade* W dan W' . Proses iterasi akan terus berjalan sampai nilai error minimum. Penelitian ini menggunakan ukuran dimensi setiap *word vector* sebesar 300.

b. Bi-GRU

GRU adalah salah satu mekanisme dari RNN yang mirip dengan LSTM [7]. GRU pertama kali diusulkan Gers, dkk. pada tahun 2014 yang merupakan model sederhana dari LSTM. Ada dua *gate* pada GRU yaitu *forget gate* dan *input gate* yang kemudian diteruskan ke *update gate*. Dari *update gate* informasi diteruskan secara selektif ke *hidden layer* untuk mengurangi masalah *gradient* saat mengingat informasi [8]. Sedangkan Bi-GRU sendiri GRU yang bekerja dari dua arah [9].

Secara konsep, analisis sentimen adalah proses *sequence learning* dengan jenis *task many-to-one*. *Word representation* yang dihasilkan oleh *word2vec* tidak terdapat *sequence information*. Karena itu, Bi-GRU digunakan untuk *sequence modelling* yang dapat mengekstrak *context information* dari *word representation* yang dihasilkan *word2vec*.



Gambar 5. Arsitektur bidirectional GRU

Pada gambar 5 Bi-GRU mempelajari *context information* dari dua arah yaitu: *forward* dan *backward*, setelah itu menggabungkan keduanya. Bi-GRU terdiri dari *forward GRU* (direpresentasikan sebagai \overrightarrow{GRU}) untuk mempelajari *feature sequence* dari Lc_1 ke Lc_{100} dan *backward GRU* (direpresentasikan sebagai \overleftarrow{GRU}) untuk mempelajari

feature sequence dari Lc_{100} ke Lc_1 . Berikut formula Bi-GRU secara detail:

$$\vec{h}_f = \overrightarrow{GRU}(Lc_n), n \in [1, 100] \dots\dots\dots (1)$$

$$\overleftarrow{h}_b = \overleftarrow{GRU}(Lc_n), n \in [100, 1] \dots\dots\dots (2)$$

c. *Global Attention*

Attention mechanism adalah mekanisme untuk memfokuskan setiap hasil *output* terhadap target kata. Sehingga kata yang dihasilkan memiliki hubungan yang lebih baik sesuai konteks [10]. Saat ini, *attention mechanism* telah terbukti efektif untuk mendapatkan hasil yang baik dengan memilih informasi yang penting. *Attention mechanism* pertama kali diusulkan di bidang *computer vision*, tujuannya untuk meniru *attention mechanism* pada manusia dengan memberikan bobot yang berbeda untuk bagian gambar [11].

Pada dokumen sentimen, setiap kata memiliki kontribusi yang berbeda terhadap konteks sentimen, dengan memberikan bobot yang berbeda pada kata, dapat memberikan fokus (*attend*) yang berbeda pada *context information* yang dihasilkan Bi-GRU. *Global attention* digunakan untuk memberikan bobot yang berbeda pada kata untuk meningkatkan pemahaman terhadap sentimen. *Global attention* dapat memfokuskan fitur dari *keyword* dan mengurangi fitur *non-keywords* menggunakan *fully-connected layer* dan *softmax function*. Berikut formula dari *global attention*:

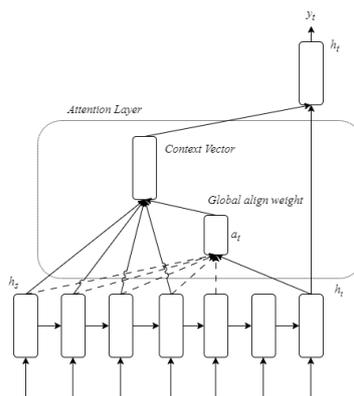
$$M = \tanh \left(\left[\begin{array}{c} W_h H \\ W_v v_a \otimes e_n \end{array} \right] \right) \dots\dots\dots (3)$$

$$a = \text{softmax}(W^T M) \dots\dots\dots (4)$$

$$r = H a^T \dots\dots\dots (5)$$

Dimana $H \in \mathbb{R}^{d \times N}$ adalah matriks yang terdiri dari *hidden vector* $[h_1, \dots, h_N]$ yang dihasilkan Bi-GRU, d adalah ukuran dari *hidden layer*. v_a adalah representasi *embedding* dan W adalah yang harus dipelajari selama *training*.

Pada formula 3 digunakan untuk mencari *alignment score* menggunakan *hyperbolic tangent* dan menghasilkan fitur yang lebih kompleks. Formula 4 digunakan untuk mencari *attention weight* dengan menambahkan *softmax* dan menghasilkan distribusi sentimen. Formula 5 digunakan untuk mencari *context vector*.



Gambar 6. Arsitektur *global attention*

Pada gambar 6 setiap *time step t*, *global attention* menghitung *alignment weight* a_t berdasarkan *target state* saat ini h_t . *context vector* c_t dihitung sebagai bobot rata-rata berdasarkan *alignment weight* dan semua *state*.

1.4 Model Test and Experiment

Eksperimen dilakukan untuk menguji model yang diusulkan. Penelitian ini menggunakan empat skenario pengujian yang berbeda untuk mencari akurasi model yang terbaik. Skenarionya adalah dengan melakukan *tuning hyperparameter*. Berikut skenario-skenario pengujian:

- a. Melakukan *tuning* pada *dropout*
Skenario pertama dengan melakukan *tuning* pada *dropout*. Yaitu dengan mengurangi *unit neural network* pada ukuran tertentu.
- b. Melakukan *tuning* pada *optimizer*
Skenario kedua dengan melakukan *tuning* pada *optimizer*. Yaitu dengan mencari *optimizer* mana yang dapat meminimalkan *error function (loss function)* hingga *cost* yang terkecil.
- c. Melakukan *tuning* pada *batch size*
Skenario ketiga dengan melakukan *tuning* pada *batch size*. Yaitu dengan menentukan jumlah sampel data *training* dalam satu kali iterasi.
- d. Melakukan *tuning* pada *units*
Skenario keempat dengan melakukan *tuning* pada *units*. Yaitu dengan menentukan jumlah *unit* atau *neuron* pada *hidden layer*.
- e. Melakukan *tuning* pada *learning rate*
Skenario kelima dengan melakukan *tuning* pada *learning rate*. Yaitu dengan menentukan *learning rate* pada *optimizer*.

1.5 Result Evaluation and Validation

Dari hasil eksperimen dan pengujian model, tahap selanjutnya dilakukan evaluasi model untuk membahas hasil pengujian dan observasi model. Evaluasi model disini *confusion matrix* yang digunakan untuk mengukur seberapa baik arsitektur “glo-att-bigru+word2vec” dalam mengatasi masalah menggunakan akurasi dan F1 score.

3. HASIL DAN PEMBAHASAN

3.1 Data Gathering

Dataset diambil / *scrapping* dari ulasan PLN mobile pada google playstore. *Dataset* dikumpulkan dari periode Agustus 2016 sampai dengan Februari 2023. *Dataset imbalance* yang digunakan untuk penelitian berjumlah 236905 berbahasa Indonesia, dengan jumlah label positif 214981 dan jumlah label negatif 21924. Sedangkan *dataset balance* yang digunakan untuk penelitian berjumlah 429962 dengan jumlah label positif dan negatif 214981. Teknik pembagian data yang digunakan untuk evaluasi model menggunakan *split ratio* dengan perbandingan 90:10.

3.2 Data Preprocessing

Sebelum dataset digunakan untuk uji coba metode yang diusulkan, *dataset* dilakukan *pre-processing* terlebih dahulu. Berikut untuk lebih detail *data preprocessing*-nya:

- a. *Case folding*

Pada proses *case folding* merubah semua huruf menjadi kecil dan menghapus semua karakter selain huruf. Hasil dari *case folding* seperti pada gambar 7.



Gambar 7. Hasil dari *case folding*

b. *Tokenization*

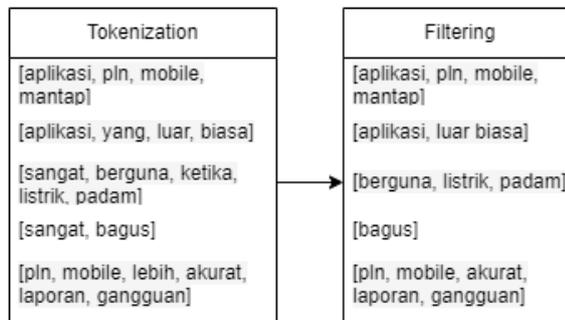
Pada proses *tokenization*, kalimat dipecah menjadi unit terkecil. Unit pada proses ini yaitu kalimat dipecah menjadi kata per kata. Hasil dari *tokenization* seperti pada gambar 8.



Gambar 8. Hasil dari *tokenization*

c. *Filtering*

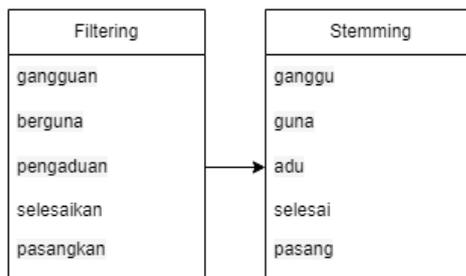
Pada proses *filtering* menghapus kata yang tidak perlu berdasarkan *stopword list*. Hasil dari *filtering* seperti pada gambar 9.



Gambar 9. Hasil dari *filtering*

d. *Stemming*

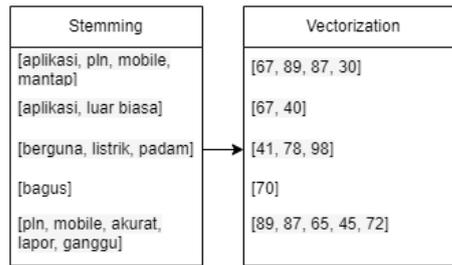
Pada proses *stemming* merubah kata menjadi kata dasar dengan menghilangkan *prefix* dan *suffix*. Hasil dari *stemming* seperti pada gambar 10.



Gambar 10. Hasil dari *stemming*

e. *Vectorization*

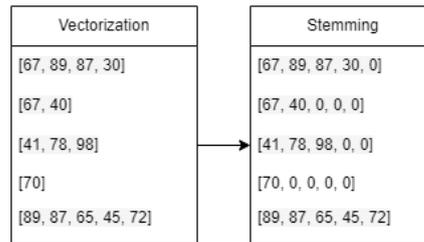
Pada proses *vectorization*, kata-kata atau token ditransformasikan menjadi angka agar bisa diproses model. Hasil dari *vectorization* seperti pada gambar 11.



Gambar 11. Hasil dari *vectorization*

f. *Padding*

Pada proses *padding* menyamakan bentuk dan ukuran teks agar memiliki input dan teks dengan ukurang yang sama. Hasil dari *padding* seperti pada gambar 12.



Gambar 12. Hasil dari *vectorization*

g. *Handling Data Imbalance*

Pada proses ini menggunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) yaitu dengan menggenerate data sampel buatan pada label yang minoritas. Karena *dataset* pada pada penelitian ini yang minoritas adalah label negatif, maka data yang digenerate adalah data berlabel negatif.

Setelah dilakukan tahapan *proprocessing*, data siap digunakan untuk proses pengujian model. Hasil setelah dilakukan *preprocessing* seperti pada tabel 1.

Tabel 1. Hasil *preprocessing*

Review	Sentiment
[67, 89, 87, 30, 0]	1
[67, 40, 0, 0, 0]	1
[41, 78, 98, 0, 0]	1
[70, 0, 0, 0, 0]	1
[89, 87, 65, 45, 72]	1

3.3 *Proposed Method*

a. *Word embedding*

Word embedding sendiri adalah teknik untuk merepresentasikan kata dalam bentuk vektor. *Input* dari *word embedding* disini adalah *dataset* yang telah dilakukan tahap *preprocessing* dan *output*-nya adalah vektor numerik yang digunakan untuk *task* klasifikasi sentimen.

b. BiGRU

BiGRU digunakan untuk *sequence learning*. *Input* dari BiGRU adalah *word embedding* dan *output*-nya adalah hasil prediksi.

c. *Global attention*

Global attention digunakan untuk memfokuskan kata yang kontribusinya paling besar. *Input* dari *global attention* adalah Bi-GRU dan *output*-nya adalah *weight* (bobot) dari *attention*.

3.4 Model Test and Experiment

A. Indikator Evaluasi

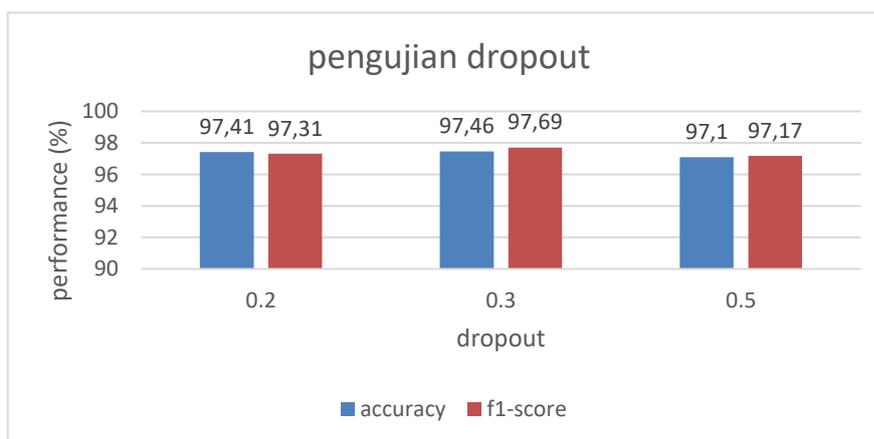
Eksperimen ini menggunakan alat pengukuran *confusion matrix*. *Confusion matrix* digunakan untuk mengukur F1-Score dan akurasi. F1-Score dipilih untuk mengukur model pada pengujian *dataset* yang *imbalance*, sedangkan akurasi dipilih untuk mengukur model pada pengujian *dataset* yang *balance*. Jumlah data *imbalance* yang digunakan untuk pengujian sebesar 213214 untuk data *training* dan 23691 untuk data *testing*. Sedangkan jumlah data *balance* yang digunakan untuk pengujian sebesar 386965 untuk data *training* dan 42997 untuk data *testing*.

B. Tuning Hyperparameter

Untuk melatih model yang diusulkan agar lebih baik, melakukan *tuning hyperparameter* (mencari *hyperparameter* terbaik) adalah hal yang penting. *Tuning hyperparameter* pada model yang diusulkan mencakup *dropout*, *optimizer*, *batch size*, *units* dan *learning rate*. *Hyperparameter* terbaik akan diterapkan pada pengujian model selanjutnya, yaitu pengujian dengan menggunakan dan tanpa menggunakan *global attention* dan pengujian perbandingan dengan arsitektur yang lain.

a. Dropout

Dropout adalah teknik pada *neural network* untuk *regularization* dan mengurangi *overfitting* dengan cara menghilangkan sementara beberapa *neuron* pada *hidden layer*. Dengan menentukan *range dropout* yang tepat dapat memberikan hasil model yang terbaik. Pada pengujian ini *range dropout* yang akan diuji adalah 0.2, 0.3, dan 0.5 seperti pada gambar 13.



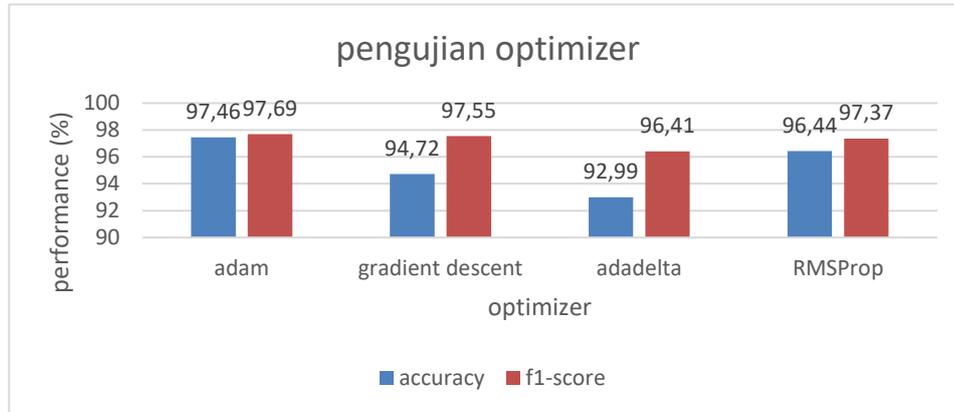
Gambar 13. Pengujian akurasi dan f1-score pada *dropout*

Pada gambar 13 menunjukkan *range dropout* terbaik ada pada *range* 0.3. Hal ini dikarenakan jika *range dropout* terlalu kecil dapat mengurangi model dalam menggeneralisasi data sehingga kemampuan model dalam memprediksi data yang baru menjadi berkurang. Sedangkan jika *dropout* terlalu besar maka model dapat menjadi kurang efektif dalam mempelajari pola-pola yang kompleks pada data sehingga model kehilangan terlalu banyak informasi selama pelatihan, dan menyebabkan kinerja model menurun.

b. Optimizer

Optimizer adalah algoritma yang digunakan untuk mengoptimalkan atau meminimalkan nilai fungsi kerugian (*loss function*) pada saat melakukan *training* model *neural network*. *Optimizer* ini bekerja dengan mengubah bobot (*weights*) dan bias (*biases*) pada setiap iterasi *training*, sehingga memperbaiki prediksi model dan

membuatnya lebih akurat. Pemilihan *optimizer* yang tepat harus dilakukan untuk meningkatkan akurasi model, menghindari *overfitting*, mengoptimalkan *learning rate*. Pada pengujian ini *optimizer* yang akan diuji adalah *gradient descent* (SGD), *adam*, *adadelata*, dan *RMSprop* seperti pada gambar 14.

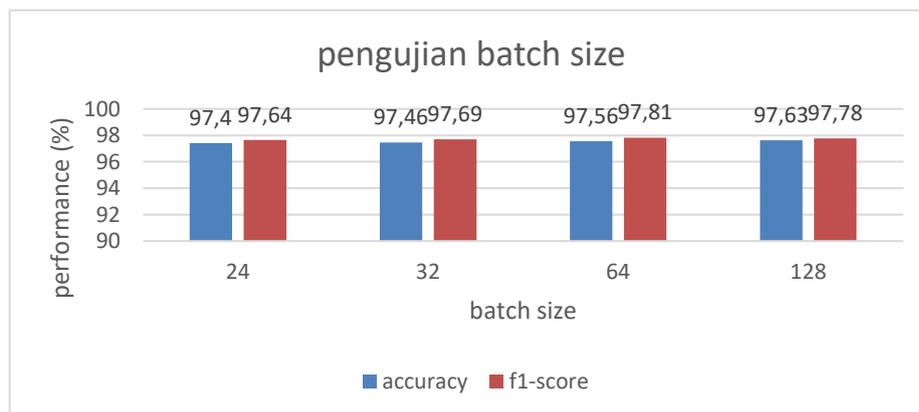


Gambar 14. Pengujian akurasi dan f1-score pada *optimizer*

Pada gambar 14 menunjukkan *optimizer* terbaik adalah *adam*. Hal ini dikarenakan *adam* (*adaptive moment estimation*) menggabungkan konsep dari SGD dan AdaGrad yaitu dengan memperbarui bobot dan bias dengan mengambil gradien rata-rata dan rata-rata kuadrat dari gradien, dan kemudian menyesuaikan *learning rate* untuk setiap parameter. Sedangkan AdaDelta dan RMSProp umumnya dipakai pada data non-stasioner seperti harga saham, suhu, dan pengujung situs web.

c. Batch Size

Batch size adalah jumlah sampel data yang diberikan kepada model pada setiap iterasi training. Pada *neural network*, menentukan *batch size* yang tepat penting untuk dilakukan karena mempengaruhi waktu *training* dan akurasi model. Pada pengujian ini, jumlah *batch size* yang akan diuji sebanyak 24, 32, 64 dan 128 seperti pada gambar 15.

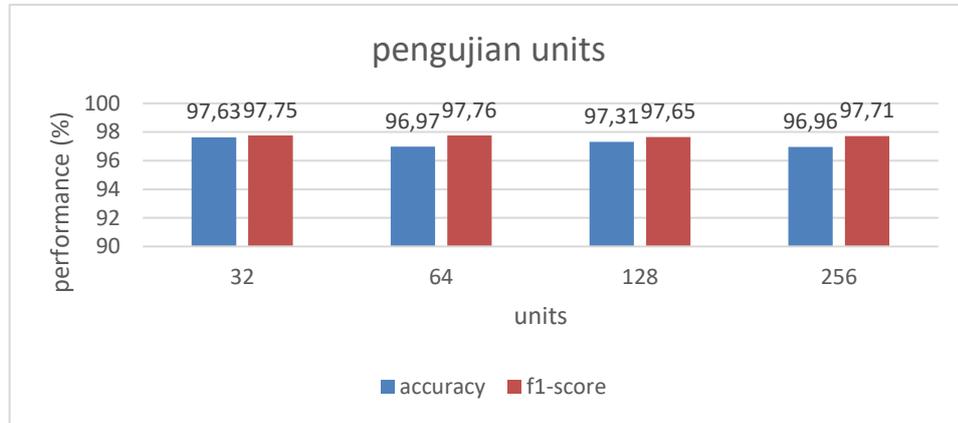


Gambar 15. Pengujian akurasi dan f1-score pada *batch size*

Pada gambar 15 pengujian akurasi terbaik ada pada *batch size* 128 sedangkan f1-score terbaik ada pada *batch size* 64. Hal ini dikarenakan *batch size* yang besar dapat membantu model lebih stabil dalam mempelajari pola-pola umum dalam data pelatihan, sehingga model lebih dapat digeneralisasi pada data yang tidak pernah dilihat sebelumnya.

d. *Units*

Units pada *neural network* merujuk pada jumlah *neuron* atau *node* yang terdapat pada *hidden layer* dari jaringan saraf tiruan. Dengan menentukan *units* yang tepat membuat model menggeneralisasi data lebih baik, akurasi yang tepat dan mengurangi kompleksitas pada *neural network*. Pada pengujian ini, jumlah *units* yang akan diuji sebanyak 32, 64, 128, 256 seperti pada gambar 16.

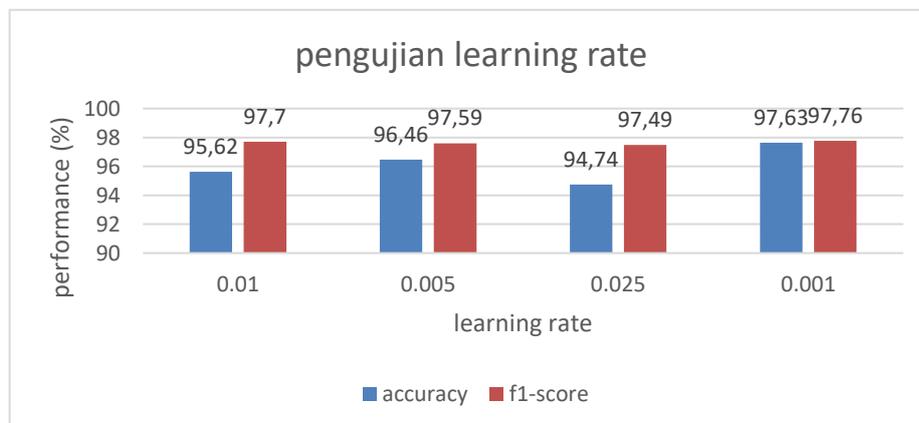


Gambar 16. Pengujian akurasi dan f1-score pada *units*

Pada gambar 16 pengujian akurasi terbaik ada pada jumlah *units* 32 sedangkan f1-score terbaik ada pada jumlah *units* 64. Hal ini dikarenakan *units* 32 dan 64 membuat model tidak terlalu kompleks sehingga mudah dipahami sedangkan *units* 128 dan 256 membuat model sulit memahami untuk membuat prediksi karena terlalu kompleks.

e. *Learning Rate*

Learning rate adalah parameter yang digunakan dalam *neural network* untuk menentukan seberapa besar langkah yang diambil oleh algoritma saat memperbarui bobot atau parameter model pada setiap iterasi. Secara sederhana, *learning rate* dapat diartikan sebagai seberapa cepat atau lambat model belajar dari data yang diberikan. Menentukan *learning rate* yang tepat perlu dilakukan karena dapat meningkatkan konvergensi lebih cepat sehingga performa lebih baik, dan meningkatkan akurasi. Pada pengujian ini, beberapa *learning rate* yang akan diuji adalah 0.1, 0.005, 0.025, 0.001 seperti pada gambar 17.



Gambar 17. Pengujian akurasi dan f1-score pada *learning rate*

Pada gambar 17 menunjukkan *learning rate* terbaik adalah 0.001. Hal ini dikarenakan *learning rate* 0.001 dapat membantu mencegah kehilangan informasi penting, karena model melakukan perubahan bobot yang lebih kecil pada setiap iterasi, sehingga dapat mempertahankan informasi yang lebih *smooth*.

3.5 Result and Validation

A. Komparatif Eksperimen

Komparatif eksperimen digunakan untuk membandingkan dan mengevaluasi arsitektur yang diusulkan menggunakan data ulasan PLN Mobile yang *balance* dan *imbalance* dengan tujuan mengukur performanya melalui mekanisme tertentu. Mekanisme pada penelitian ini ada dua, yang pertama membandingkan menggunakan dan tanpa menggunakan *global attention* dan yang kedua membandingkan dengan metode analisis sentimen yang lain. Parameter yang digunakan untuk komparatif eksperimen seperti pada tabel 2.

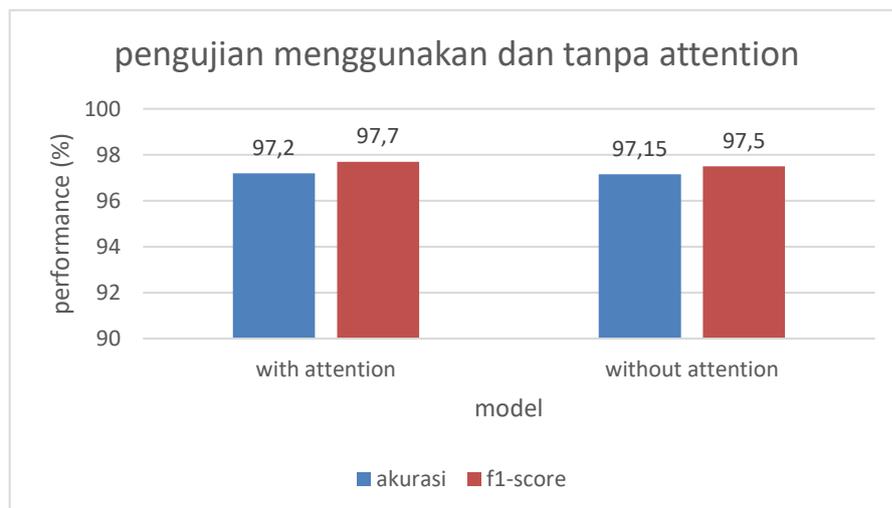
Tabel 2. *Hyperparameter list*

Hyperparameter	Value (data balance)	Value (data imbalance)
dropout	0.3	0.3
optimizer	adam	adam
Batch size	128	64
units	32	64
learning rate	0.001	0.001
Loss function	binary crossentropy	binary crossentropy

Daftar *hyperparameter* pada tabel 2 didapat dari pengujian *hyperparameter* pada bab 3.4 yang merupakan *hyperparameter* terbaik.

a. Perbandingan menggunakan dan tanpa *global attention*

Perbandingan menggunakan dan tanpa *global attention* bertujuan untuk menguji seberapa baik arsitektur ketika ditambahkan *global attention* seperti pada gambar 18.



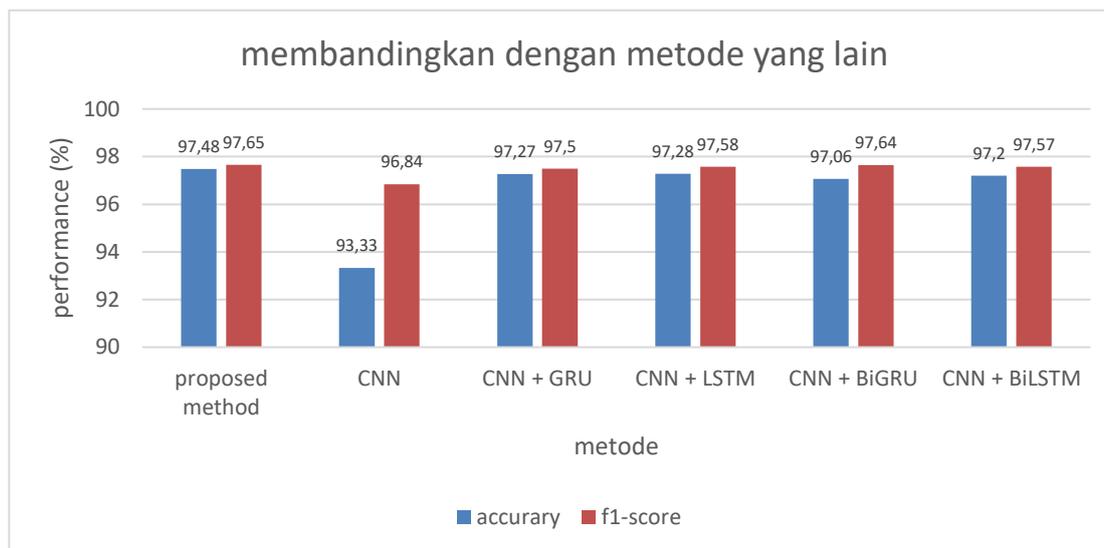
Gambar 18. Pengujian akurasi dan f1-score pada *global attention*

Pada gambar 18 menunjukkan ketika *global attention* diterapkan dalam arsitektur maka performa meningkat. Hal ini dikarenakan *global attention*

memfokuskan kata-kata yang paling berkontribusi pada *sequence learning* sehingga mampu menangkap informasi penting dengan begitu performa menjadi meningkat.

b. Perbandingan menggunakan metode analisis sentimen yang lain

Untuk lebih membuktikan efektifitas metode analisis sentimen yang diusulkan pada penelitian ini. Metode yang diusulkan akan dibandingkan dengan metode analisis sentimen yang lain yaitu: CNN, CNN+GRU, CNN+LSTM, CNN+BiGRU, dan CNN+BiLSTM. Pengujian menggunakan data yang sama yaitu ulasan PLN Mobile yang *balance* dan *imbalance*. Hasil pengujian seperti pada gambar 17.



Gambar 3. Pengujian akurasi dan f1-score pada metode analisis sentimen yang lain

Pada gambar 17 menunjukkan pengujian model yang diusulkan mengungguli metode yang lain dengan hasil akurasi 97.48%, *f1-score* 97.65%. Hal ini dikarenakan *global attention* mampu menangkap informasi penting dari GRU, ditambah GRU menggunakan dua arah (*bidirectional*) sehingga dapat mempelajari semantik dari belakang ke depan (*forward*) dan dari belakang ke depan (*backward*). Ketika model CNN dikombinasikan dengan model LSTM, BiLSTM, GRU, BiGRU memang dapat mengekstraksi semantik seperti *global attention* akan tetapi pada kasus *sequence learning* CNN kurang efektif. Hal ini disebabkan ketika bekerja pada *sequence learning* CNN mengabaikan *sequence* pada data karena *convolution layer* hanya melihat blok-blok kecil dari data *input* dan tidak memiliki kemampuan untuk menangkap konteks dari *sequence* data.

4. KESIMPULAN DAN SARAN

Penelitian ini bertujuan untuk mengatasi masalah pada algoritma *deep learning* seperti LSTM dan GRU yang tidak menangkap informasi penting pada *sequence learning*. Metode yang diusulkan untuk mengatasi masalah tersebut adalah menggunakan *bidirectional* GRU dengan mengkombinasikan *global attention* dan *word2vec* sebagai *word embedding*. *Global attention* bekerja dengan memfokuskan kata-kata yang paling berkontribusi sehingga membentuk seperti *keyword*. Model yang diusulkan kemudian dilakukan *hyperparameter setting* untuk menentukan parameter terbaik dan membandingkan dengan metode analisis sentimen yang lain.

Model tersebut dilatih menggunakan *dataset* ulasan PLN Mobile. Untuk memvalidasi model, penulis membandingkan dengan lima metode yaitu CNN, CNN+LSTM, CNN+GRU, CNN+BiLSTM, dan CNN+BiGRU dengan data yang sama hasilnya dapat disimpulkan bahwasanya metode yang diusulkan mengungguli dengan akurasi 97.48%, *f1-score* 97.65%.

Adapun saran untuk penelitian selanjutnya adalah sebagai berikut: 1). Menggunakan *attention mechanism* yang lain seperti *self-attention*, *multi-head attention* dan *hierarchical attention* untuk meningkatkan performa. 2). Menerapkan pengujian *cross validation* pada saat *training* untuk mengurangi *overfitting* dan *underfitting*. 3). Menggunakan *word embedding* yang lain seperti *GloVe*, *FastText*. 4). Menerapkan model yang diusulkan untuk aplikasi yang nyata.

DAFTAR PUSTAKA

- [1] B. Liu, *Sentiment Analysis and Opinion Mining*. Cham: Springer International Publishing, 2012.
- [2] B. N. Rodrigues Chagas, J. A. Nogueira Viana, O. Reinhold, F. Lobato, A. F. L. Jacob, and R. Alt, "Current Applications of Machine Learning Techniques in CRM: A Literature Review and Practical Implications," in *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*, Dec. 2018, pp. 452–458, doi: 10.1109/WI.2018.00-53.
- [3] M. Rhanoui, M. Mikram, S. Yousfi, and S. Barzali, "A CNN-BiLSTM Model for Document-Level Sentiment Analysis," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 3, pp. 832–847, Jul. 2019, doi: 10.3390/make1030048.
- [4] R. Ni and H. Cao, "Sentiment Analysis based on GloVe and LSTM-GRU," *Chinese Control Conf. CCC*, vol. 2020-July, pp. 7492–7497, 2020, doi: 10.23919/CCC50068.2020.9188578.
- [5] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM," *IEEE Access*, vol. 7, no. c, pp. 51522–51532, 2019, doi: 10.1109/ACCESS.2019.2909919.
- [6] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, Jan. 2013, [Online]. Available: <http://arxiv.org/abs/1301.3781>.
- [7] A. Bhuvaneswari, J. T. Jones Thomas, and P. Kesavan, "Embedded Bi-directional GRU and LSTM Learning Models to Predict Disaster on Twitter Data," *Procedia Comput. Sci.*, vol. 165, pp. 511–516, 2019, doi: 10.1016/j.procs.2020.01.020.
- [8] L. Zeng, W. Ren, and L. Shan, "Attention-based bidirectional gated recurrent unit neural networks for well logs prediction and lithology identification," *Neurocomputing*, vol. 414, pp. 153–171, 2020, doi: 10.1016/j.neucom.2020.07.026.
- [9] H. M. Lynn, S. B. Pan, and P. Kim, "A Deep Bidirectional GRU Network Model for Biometric Electrocardiogram Classification Based on Recurrent Neural Networks," *IEEE Access*, vol. 7, pp. 145395–145405, 2019, doi: 10.1109/ACCESS.2019.2939947.
- [10] D. Bahdanau, K. Cho, Y. Bengio, P. By, and R. Aharoni, "Machine Translation is Everywhere," 2014.
- [11] W. Li, F. Qi, M. Tang, and Z. Yu, "Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification," *Neurocomputing*, vol. 387, pp. 63–77, 2020, doi: 10.1016/j.neucom.2020.01.006.