Published online on the page: https://journal.upp.ac.id/index.php/rjti/index

R J T I
Riau Jurnal Teknik Informatika
| E - ISSN (*Online*) 2829-2529 |

Klasifikasi Multilabel Artikel Ilmiah Bidang Komputer Menggunakan *SciBERT* dan *TextCNN*

Regina Nathania Agussalim[1,*], Agustinus Jacobus[2], Sherwin R. U. A. Sompie[3],

[1,2,3]Universitas Sam Ratulangi, Sulawesi Utara, Indonesia

Informasi Artikel

Sejarah Artikel:
Submit: 22 Juni 2026
Revisi: 06 Juli 2026
Diterima: 10 Juli 2026
Diterbitkan: 13 Juli 2026

Kata Kunci

Artikel Ilmiah, Klasifikasi Multilabel, *SciBERT, TextCNN, Threshold Tuning*

Korespondensi

E-mail: reginagussalim08@gmail.com

A B S T R A C T

The increasing number of scientific articles makes the process of searching, filtering, and organizing relevant literature more challenging. This problem becomes more complex because one scientific article can be associated with more than one research topic. This study aims to implement and evaluate a SciBERT-TextCNN model for multilabel classification of scientific articles in the Computer Science field based on title and abstract metadata. The dataset used was arXiv article metadata obtained from Kaggle and limited to Computer Science categories. The research stages included data preprocessing, Multi-Label Random Under-Sampling, tokenization using SciBERT tokenizer, model training, threshold tuning, and evaluation. The model was tested using five scenarios: SciBERT frozen and fine-tuning the last 1, 2, 3, and 4 layers. The best performance was obtained by fine-tuning the last 2 SciBERT layers, achieving precision of 0.708, recall of 0.726, F1-score of 0.715, macro F1-score of 0.649, and Hamming Loss of 0.02333. These results show that SciBERT-TextCNN can be used to support automatic multilabel classification of scientific articles.

Abstrak

Pertumbuhan jumlah artikel ilmiah menyebabkan proses pencarian, penyaringan, dan pengelompokan literatur yang relevan menjadi semakin menantang. Permasalahan ini semakin kompleks karena satu artikel ilmiah dapat berkaitan dengan lebih dari satu topik penelitian. Penelitian ini bertujuan untuk mengimplementasikan dan mengevaluasi model SciBERT-TextCNN dalam klasifikasi multilabel artikel ilmiah bidang Computer Science berdasarkan metadata judul dan abstrak. Dataset yang digunakan merupakan metadata artikel arXiv yang diperoleh melalui Kaggle dan dibatasi pada kategori Computer Science. Tahapan penelitian meliputi pra-pemrosesan data, Multi-Label Random Under-Sampling, tokenisasi menggunakan tokenizer SciBERT, pelatihan model, threshold tuning, dan evaluasi. Model diuji menggunakan lima skenario, yaitu SciBERT frozen serta fine-tuning 1, 2, 3, dan 4 lapisan terakhir. Hasil terbaik diperoleh pada skenario fine-tuning 2 lapisan terakhir SciBERT dengan precision 0,708, recall 0,726, F1-score 0,715, macro F1-score 0,649, dan Hamming Loss 0,02333. Hasil ini menunjukkan bahwa SciBERT-TextCNN dapat digunakan untuk mendukung klasifikasi multilabel artikel ilmiah secara otomatis.

1. Pendahuluan

Pertumbuhan jumlah artikel ilmiah yang sangat besar menyebabkan proses pencarian, penyaringan, dan pengelompokan literatur menjadi semakin menantang. National Science Board melaporkan bahwa publikasi sains dan rekayasa global mencapai 3,3 juta artikel pada tahun 2022,

meningkat 59% dibandingkan tahun 2012 [1]. Kondisi ini dapat menimbulkan information overload, yaitu keadaan ketika jumlah informasi yang tersedia sangat besar sehingga pengguna mengalami kesulitan dalam menemukan informasi yang benar-benar relevan [2]. Permasalahan tersebut semakin kompleks karena satu artikel ilmiah sering kali tidak hanya berkaitan dengan satu topik, tetapi dapat mencakup beberapa bidang kajian secara bersamaan. Oleh karena itu, diperlukan pendekatan klasifikasi otomatis yang mampu memberikan lebih dari satu label pada satu artikel ilmiah. Penelitian ini menawarkan solusi berupa model SciBERT-TextCNN untuk multilabel classification artikel ilmiah berdasarkan metadata judul dan abstrak. SciBERT digunakan untuk membentuk representasi kontekstual dari teks ilmiah, sedangkan TextCNN digunakan untuk mengekstraksi pola lokal dari representasi tersebut.

Klasifikasi artikel ilmiah dapat dilakukan menggunakan pendekatan content-based maupun metadata-based. Pendekatan content-based memanfaatkan isi penuh dokumen, sedangkan pendekatan metadata-based memanfaatkan informasi seperti judul, abstrak, kata kunci, penulis, dan kategori [3]. Meskipun isi penuh artikel dapat menyediakan informasi yang lebih lengkap, tidak semua artikel tersedia secara terbuka dalam bentuk full text. Oleh karena itu, metadata seperti judul dan abstrak menjadi alternatif yang lebih praktis karena umumnya tersedia dan tetap merepresentasikan inti artikel. Dalam konteks artikel ilmiah, pendekatan multilabel classification lebih sesuai dibandingkan single-label classification karena satu artikel dapat memiliki keterkaitan dengan beberapa kategori secara bersamaan [4]. Namun, tugas multilabel classification juga memiliki tantangan, seperti ketidakseimbangan distribusi label, korelasi antarlabel, dan sparsitas label [5].

Beberapa penelitian sebelumnya telah membahas klasifikasi multilabel pada dokumen ilmiah maupun teks domain khusus. Li dan Ou [6] melakukan klasifikasi multilabel makalah penelitian berdasarkan judul dan abstrak menggunakan algoritma Multi-Label K-Nearest Neighbour dan representasi TF-IDF. Penelitian tersebut menunjukkan bahwa metadata judul dan abstrak dapat digunakan untuk mengklasifikasikan makalah penelitian, tetapi pendekatan yang digunakan masih bergantung pada fitur berbasis frekuensi kata sehingga belum memanfaatkan representasi kontekstual berbasis Transformer. Fallah et al. [7] mengkaji adaptasi model Transformer untuk multilabel text classification dan menunjukkan bahwa penyesuaian nilai threshold dapat meningkatkan performa model pada dataset yang tidak seimbang. Penelitian tersebut relevan karena menegaskan pentingnya threshold tuning dalam klasifikasi multilabel.

Zhang et al. [8] mengembangkan model klasifikasi multilabel untuk literatur kanker pada tingkat publikasi dengan menggunakan gabungan judul, abstrak, dan kata kunci. Penelitian tersebut menggunakan pendekatan berbasis BERT dan beberapa arsitektur deep learning, serta menerapkan strategi down-sampling untuk mengurangi ketidakseimbangan data. Hasilnya menunjukkan bahwa kombinasi representasi berbasis BERT dan arsitektur deep learning dapat menghasilkan performa yang baik pada klasifikasi multilabel literatur ilmiah. Ahmed Sajid et al. [9] juga mengusulkan teknik klasifikasi dokumen multilabel berbasis metadata dengan memanfaatkan judul dan kata kunci. Penelitian tersebut menekankan bahwa metadata dapat menjadi alternatif penting ketika teks lengkap dokumen tidak tersedia, tetapi pendekatannya belum secara khusus mengombinasikan representasi bahasa ilmiah berbasis SciBERT dengan ekstraksi fitur lokal berbasis TextCNN.

Penelitian lain oleh Likhareva et al. [10] mengusulkan pendekatan SciBERT-CNN untuk klasifikasi teks multilabel pada artikel ilmiah dengan strategi masukan multi-segmen dan topic modeling. Penelitian tersebut menunjukkan bahwa penggabungan SciBERT dan CNN mampu meningkatkan performa klasifikasi teks ilmiah dibandingkan model dasar. Rostam dan Kertész [11] membandingkan beberapa model bahasa besar, seperti BERT, SciBERT, BioBERT, dan BlueBERT, untuk klasifikasi teks ilmiah. Hasil penelitian tersebut menunjukkan bahwa model yang dilatih pada domain ilmiah, terutama SciBERT, memiliki performa yang kompetitif dalam memahami teks ilmiah. Selain itu, Ramakrishnan dan Dhinesh Babu [12] menerapkan BERT dan Clipped Asymmetric Loss untuk

klasifikasi emosi multilabel pada data yang tidak seimbang, sehingga menunjukkan pentingnya strategi penanganan ketidakseimbangan label pada tugas multilabel.

Berdasarkan penelitian-penelitian tersebut, dapat dilihat bahwa klasifikasi multilabel teks ilmiah telah banyak dikaji menggunakan metadata, BERT, SciBERT, CNN, maupun teknik penyesuaian threshold. Namun, masih terdapat celah penelitian pada penerapan kombinasi SciBERT dan TextCNN untuk klasifikasi multilabel artikel ilmiah berbasis metadata judul dan abstrak pada kategori Computer Science arXiv. Selain itu, belum banyak penelitian yang secara khusus menganalisis pengaruh skenario pembekuan parameter SciBERT dan fine-tuning beberapa lapisan terakhir SciBERT terhadap performa model SciBERT-TextCNN. Dengan demikian, penelitian ini berfokus pada pengembangan dan evaluasi model SciBERT-TextCNN melalui beberapa skenario, yaitu SciBERT frozen serta fine-tuning 1, 2, 3, dan 4 lapisan terakhir. Perbandingan dan ringkasan dari beberapa penelitian terdahulu tersebut dapat dilihat pada Tabel 1.

Tabel 1. Ringkasan Penelitian Terdahulu

Penelitian	Metode	Temuan Utama
Li dan Ou [6]	ML-KNN + TF-IDF pada judul dan abstrak	Metadata efektif untuk klasifikasi, namun berbasis frekuensi kata
Fallah et al. [7]	Adaptasi Transformer untuk multilabel text classification	Threshold tuning meningkatkan performa pada data tidak seimbang
Zhang et al. [8]	BERT + deep learning + down-sampling, judul-abstrak-kata kunci	Kombinasi BERT dan deep learning efektif untuk literatur ilmiah
Ahmed Sajid et al. [9]	Klasifikasi multilabel berbasis metadata judul dan kata kunci	Metadata efektif ketika full text tidak tersedia
Likhareva et al. [10]	SciBERT-CNN, masukan multi-segmen dan topic modeling	SciBERT+CNN meningkatkan performa dibanding model dasar
Rostam & Kertész [11]	Perbandingan BERT, SciBERT, BioBERT, BlueBERT	SciBERT kompetitif untuk memahami teks ilmiah

Berdasarkan uraian tersebut, rumusan masalah dalam penelitian ini adalah: (1) bagaimana merancang dan mengimplementasikan model SciBERT-TextCNN untuk klasifikasi multilabel artikel ilmiah bidang Computer Science berdasarkan metadata judul dan abstrak; (2) bagaimana pengaruh skenario pembekuan (frozen) dan fine-tuning beberapa lapisan terakhir SciBERT terhadap performa klasifikasi yang dihasilkan; dan (3) bagaimana pengaruh penerapan threshold tuning per kelas terhadap keseimbangan precision dan recall pada tugas klasifikasi multilabel yang memiliki distribusi label tidak seimbang.

Tujuan dari penelitian ini adalah mengimplementasikan model SciBERT-TextCNN untuk klasifikasi multilabel artikel ilmiah berdasarkan judul dan abstrak, mengevaluasi performa model menggunakan metrik precision, recall, F1-score, dan Hamming Loss, serta menganalisis pengaruh skenario pembekuan dan fine-tuning lapisan SciBERT terhadap performa klasifikasi. Hasil penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan metode klasifikasi artikel ilmiah secara otomatis, khususnya pada tugas multilabel classification berbasis teks ilmiah, serta dapat menjadi dasar bagi pengembangan sistem pengelolaan dan pencarian referensi ilmiah yang lebih efisien.

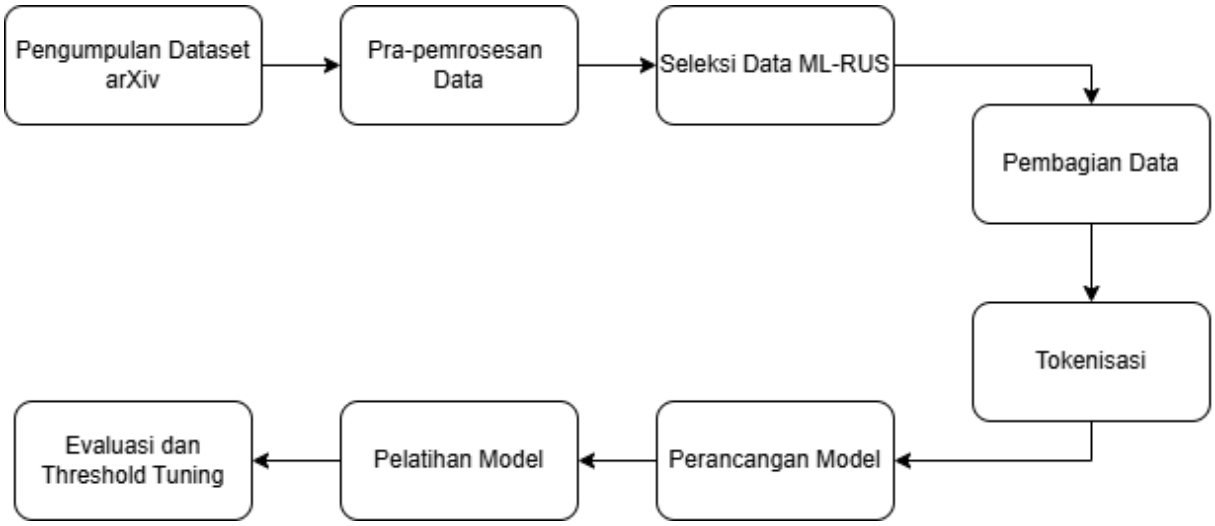
2. Metode Penelitian

2.1. Tahapan Penelitian

Secara umum, tahapan penelitian meliputi pengumpulan dataset, pra-pemrosesan data, seleksi data menggunakan Multi-Label Random Under-Sampling (ML-RUS), pembagian data, tokenisasi,

perancangan model SciBERT-TextCNN, pelatihan, evaluasi, threshold tuning, dan pengujian model pada data baru. Dataset yang digunakan adalah metadata artikel ilmiah arXiv, dengan atribut utama berupa title, abstract, dan categories. Karena penelitian berfokus pada bidang Computer Science, kategori yang digunakan dibatasi pada label arXiv dengan awalan 'cs.'. Secara keseluruhan, alur tahapan penelitian ini diilustrasikan pada Gambar 1.

Setelah data diperoleh, dilakukan pra-pemrosesan untuk membersihkan teks, menghapus data duplikat dan data tidak valid, serta menggabungkan judul dan abstrak sebagai masukan model. Selanjutnya, ML-RUS digunakan untuk mengurangi dominasi label mayoritas pada dataset multilabel. Data hasil seleksi kemudian dibagi menjadi data latih, data validasi, dan data uji. Data latih digunakan untuk pelatihan model, data validasi untuk pemantauan pelatihan dan penentuan threshold optimal, sedangkan data uji digunakan untuk evaluasi akhir.



Gambar 1. Diagram Alur Tahapan Penelitian

Teks masukan kemudian ditokenisasi menggunakan tokenizer SciBERT menjadi input_ids dan attention_mask. Hasil tokenisasi diproses oleh SciBERT untuk menghasilkan last hidden state, kemudian diteruskan ke TextCNN untuk mengekstraksi pola lokal melalui beberapa ukuran kernel. Pelatihan dilakukan dalam lima skenario, yaitu SciBERT frozen serta fine-tuning 1, 2, 3, dan 4 lapisan terakhir. Setiap skenario dievaluasi menggunakan precision, recall, F1-score, macro F1-score, dan Hamming Loss. Selain evaluasi dengan threshold standar 0,5, dilakukan threshold tuning per kelas untuk memperoleh nilai ambang yang lebih sesuai bagi setiap label.

2.2. Dataset dan Pra-Pemrosesan Data

Dataset yang digunakan dalam penelitian ini merupakan metadata artikel ilmiah arXiv yang diperoleh melalui Kaggle. Atribut yang digunakan dibatasi pada *title*, *abstract*, dan *categories*, karena *title* dan *abstract* digunakan sebagai masukan teks, sedangkan *categories* digunakan sebagai label target. Penelitian ini berfokus pada artikel bidang *Computer Science*, sehingga hanya kategori dengan awalan `cs.` yang dipertahankan. Jika satu artikel memiliki lebih dari satu kategori *Computer Science*, seluruh kategori tersebut tetap digunakan sesuai dengan karakteristik *multilabel classification*.

Pra-pemrosesan data dilakukan untuk menyiapkan teks dan label sebelum pelatihan model. Tahapan ini meliputi seleksi fitur, seleksi kategori *Computer Science*, penghapusan data duplikat, penghapusan data tidak valid, pembersihan teks, serta penggabungan judul dan abstrak. Data duplikat dihapus berdasarkan kombinasi *title* dan *abstract*, sedangkan data tidak valid seperti artikel *withdrawn* atau artikel dengan teks terlalu pendek juga dihapus dari *dataset*.

Pra-pemrosesan data dilakukan untuk menyiapkan teks dan label sebelum pelatihan model. Tahapan ini meliputi seleksi fitur, seleksi kategori *Computer Science*, penghapusan data duplikat,

penghapusan data tidak valid, pembersihan teks, serta penggabungan judul dan abstrak. Data duplikat dihapus berdasarkan kombinasi *title* dan *abstract*, sedangkan data tidak valid seperti artikel *withdrawn* atau artikel dengan teks terlalu pendek juga dihapus dari *dataset* Untuk memperlihatkan dampak tahap pra-pemrosesan secara transparan, jumlah data sebelum seleksi kategori Computer Science tercatat sebanyak 2.902.254 baris. Setelah seleksi kategori, penghapusan duplikat, dan penghapusan data tidak valid (withdrawn/teks terlalu pendek), jumlah data berkurang menjadi 850.178 baris, sebelum akhirnya diseimbangkan melalui ML-RUS menjadi 170.001 baris sebagaimana ditunjukkan pada Tabel 2.

Pembersihan teks mencakup penghapusan URL, alamat email, sitasi LaTeX, entitas HTML, baris baru, spasi berlebih, normalisasi Unicode, serta konversi ekspresi LaTeX ke bentuk teks biasa agar informasi ilmiah tetap dipertahankan. Setelah teks dibersihkan, *title* dan *abstract* digabungkan menjadi satu kolom teks sebagai masukan model. Selanjutnya, seleksi data menggunakan ML-RUS dilakukan untuk mengurangi dominasi label mayoritas dengan tetap mempertahankan data yang mengandung label minoritas. Pendekatan ini mengacu pada strategi random resampling pada klasifikasi multilabel yang digunakan untuk menangani ketidakseimbangan distribusi label [13].

Data hasil seleksi kemudian dibagi menjadi data latih, data validasi, dan data uji dengan proporsi sekitar 80:10:10. Pembagian data dilakukan menggunakan pendekatan *iterative train-test* split agar distribusi label multilabel tetap terjaga pada setiap subset. Data latih digunakan untuk pelatihan model, data validasi digunakan untuk pemantauan pelatihan dan penentuan *threshold* optimal, sedangkan data uji digunakan untuk evaluasi akhir. Hasil pembagian data ditunjukkan pada Tabel 2.

Tabel 2. Hasil Pembagian Data

Subset	Jumlah Data	Presentase
Data Latih	136.361	80,21%
Data Validasi	16.827	9,90%
Data Uji	16.813	9,89%
Total	170.001	100%

2.3. Perancangan Model SciBERT-TextCNN

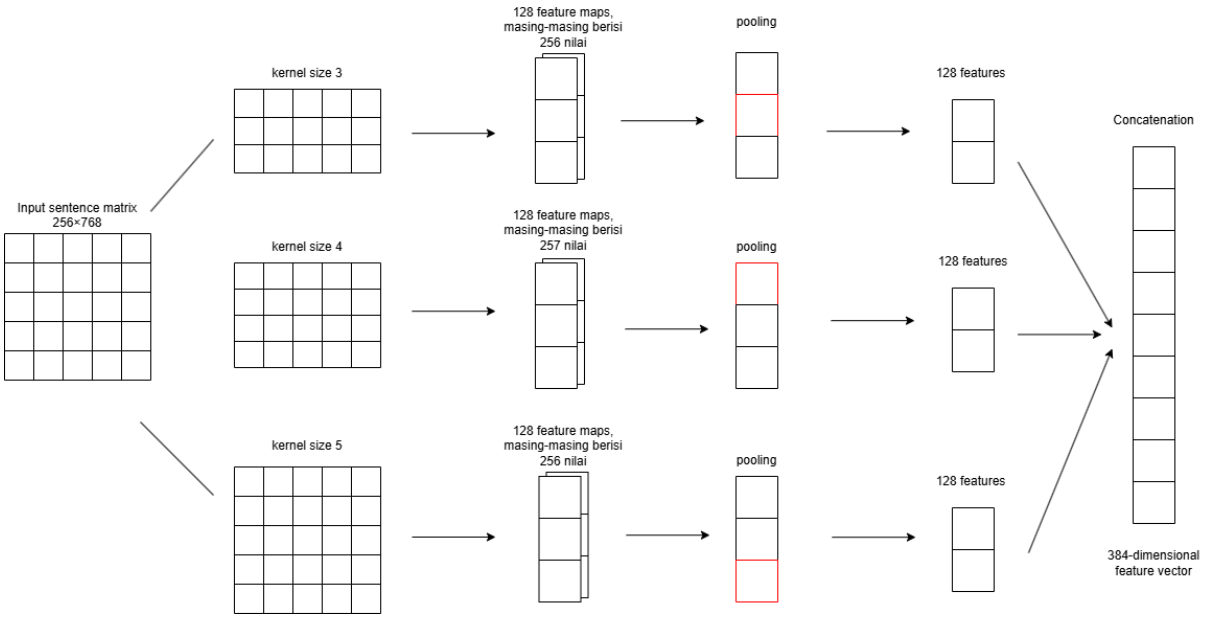
Model yang digunakan dalam penelitian ini merupakan kombinasi SciBERT dan TextCNN. SciBERT digunakan sebagai model representasi teks ilmiah, sedangkan TextCNN digunakan untuk mengekstraksi pola lokal dari representasi kontekstual yang dihasilkan oleh SciBERT. Penggunaan SciBERT didasarkan pada karakteristiknya sebagai model bahasa yang dilatih menggunakan korpus ilmiah, sehingga lebih sesuai untuk memproses teks artikel ilmiah yang mengandung istilah teknis [14].

Sebelum diproses oleh model, teks hasil penggabungan judul dan abstrak ditokenisasi menggunakan tokenizer. Setiap teks dibatasi hingga panjang maksimum 256 token. Apabila teks melebihi batas tersebut, dilakukan truncation. Sebaliknya, apabila teks memiliki panjang kurang dari 256 token, dilakukan padding agar seluruh data memiliki ukuran masukan yang sama. Hasil tokenisasi berupa input_ids dan attention_mask. input_ids merepresentasikan token dalam bentuk indeks numerik, sedangkan attention_mask digunakan untuk membedakan token asli dan token hasil padding.

Masukan berupa input_ids dan attention_mask kemudian diproses oleh SciBERT. Keluaran yang digunakan dari SciBERT adalah last hidden state, yaitu representasi kontekstual dari seluruh token pada lapisan encoder terakhir. Dengan panjang maksimum 256 token dan ukuran hidden state 768, setiap teks direpresentasikan dalam bentuk matriks berukuran 256 × 768. Representasi ini tetap mempertahankan informasi urutan token sehingga dapat digunakan oleh TextCNN untuk mengekstraksi pola lokal dari teks.

TextCNN pada penelitian ini menggunakan tiga ukuran kernel, yaitu 3, 4, dan 5. Penggunaan beberapa ukuran kernel mengacu pada konsep TextCNN yang memanfaatkan filter konvolusi dengan ukuran berbeda untuk menangkap pola lokal atau pola n-gram pada teks [15]. Setiap ukuran kernel memiliki 128 filter. Kernel berukuran 3 digunakan untuk menangkap pola dari tiga token berurutan, kernel berukuran 4 menangkap pola dari empat token berurutan, dan kernel berukuran 5 menangkap pola dari lima token berurutan. Setelah proses konvolusi, fungsi aktivasi ReLU diterapkan untuk menambahkan sifat non-linear pada model.

Hasil dari setiap cabang konvolusi kemudian diproses menggunakan global max pooling. Proses pooling digunakan untuk mengambil fitur paling dominan dari setiap feature map. Karena setiap ukuran kernel memiliki 128 filter, setiap cabang menghasilkan 128 fitur utama. Hasil dari tiga cabang konvolusi kemudian digabungkan melalui proses concatenation, sehingga menghasilkan vektor fitur berdimensi 384. Vektor fitur tersebut diteruskan ke lapisan dropout, dense layer dengan 128 unit, dropout kedua, dan lapisan keluaran.



Gambar 2. Arsitektur Model SciBERT-TextCNN

Lapisan keluaran menghasilkan 40 nilai logits sesuai jumlah label kategori Computer Science yang digunakan. Karena penelitian ini merupakan multilabel classification, setiap label diprediksi secara independen. Pada tahap evaluasi dan inference, nilai logits dikonversi menjadi probabilitas menggunakan fungsi sigmoid. Arsitektur model SciBERT-TextCNN ditunjukkan pada Gambar 2.

2.4. Skenario Pengujian dan Evaluasi

Pengujian dilakukan melalui lima skenario pelatihan untuk mengetahui pengaruh pembekuan dan pembukaan lapisan SciBERT terhadap performa klasifikasi. Skenario pertama menggunakan SciBERT dalam kondisi frozen, yaitu parameter utama SciBERT tidak diperbarui selama pelatihan. Pada skenario ini, SciBERT berperan sebagai pembentuk representasi kontekstual, sedangkan proses pembelajaran dilakukan pada lapisan TextCNN dan lapisan klasifikasi.

Tabel 3. Konfigurasi Hyperparameter Model SciBERT-TextCNN

Hyperparameter	Nilai
Panjang token maksimum	256
Jumlah filter TextCNN	128
Ukuran kernel	3, 4, 5
Dense layer	128
Dropout 1 / Dropout 2	0,3 / 0,5
Optimizer	AdamW

Learning rate SciBERT (layer fine-tuning)	1×10^{-5}
Learning rate TextCNN & klasifikasi	5×10^{-5}
Gamma positif	0,0
Gamma negatif	4,0
Clip	0,05
Batch size	128
Epoch maksimum	25
Early stopping patience	5
Perangkat komputasi	GPU P100 Kaggle (CUDA)

Empat skenario berikutnya menggunakan pendekatan fine-tuning pada lapisan terakhir SciBERT. Pada skenario fine-tuning, sebagian lapisan akhir SciBERT dibuka sehingga parameternya ikut diperbarui selama proses pelatihan. Skenario ini digunakan untuk menganalisis apakah pembaruan parameter SciBERT dapat meningkatkan kemampuan model dalam menyesuaikan representasi teks terhadap karakteristik dataset artikel ilmiah yang digunakan.

Seluruh konfigurasi hyperparameter yang digunakan selama proses pelatihan model dirangkum pada Tabel 3. Pada proses pelatihan, model menggunakan Asymmetric Loss sebagai fungsi loss. Fungsi ini digunakan karena tugas multilabel classification memiliki kecenderungan jumlah label negatif yang lebih banyak dibandingkan label positif [16]. Dengan menggunakan parameter gamma positive, gamma negative, dan clip, Asymmetric Loss membantu model agar tidak terlalu didominasi oleh label negatif selama proses pembelajaran.

Karena keluaran model berupa probabilitas untuk setiap label, diperlukan nilai threshold untuk mengubah probabilitas menjadi label biner. Evaluasi awal dilakukan menggunakan threshold standar 0,5. Setelah itu, dilakukan threshold tuning per kelas menggunakan data validasi. Nilai threshold optimal dipilih berdasarkan nilai yang menghasilkan F1-score terbaik pada masing-masing label. Nilai threshold tersebut kemudian diterapkan pada data uji untuk memperoleh hasil evaluasi akhir. Pendekatan ini digunakan karena setiap label dapat memiliki distribusi probabilitas yang berbeda, sehingga satu nilai threshold yang sama untuk seluruh label belum tentu memberikan hasil terbaik.

3. Hasil dan Pembahasan

Eksperimen dilakukan untuk mengetahui performa model SciBERT-TextCNN dalam melakukan multilabel classification artikel ilmiah bidang Computer Science. Pengujian dilakukan pada lima skenario, yaitu SciBERT-TextCNN frozen, fine-tuning 1 lapisan terakhir, fine-tuning 2 lapisan terakhir, fine-tuning 3 lapisan terakhir, dan fine-tuning 4 lapisan terakhir. Setiap skenario dievaluasi menggunakan weighted precision, weighted recall, weighted F1-score, macro F1-score, dan Hamming Loss. Evaluasi dilakukan dalam dua kondisi, yaitu menggunakan threshold standar 0,5 dan menggunakan threshold optimal per kelas.

Berdasarkan Tabel 4, penggunaan threshold standar 0,5 menghasilkan nilai recall yang tinggi pada seluruh skenario, tetapi nilai precision relatif lebih rendah. Hal ini menunjukkan bahwa model cenderung menghasilkan lebih banyak prediksi positif. Dalam kasus multilabel classification, kondisi ini dapat membantu model menemukan banyak label yang benar, tetapi juga meningkatkan kemungkinan munculnya label yang sebenarnya tidak relevan. Akibatnya, nilai precision menjadi lebih rendah karena jumlah prediksi positif yang keliru meningkat.

Setelah dilakukan threshold tuning per kelas yang dapat dilihat pada Tabel 5, nilai precision meningkat pada seluruh skenario. Pada skenario SciBERT-TextCNN frozen, precision meningkat dari 0,567 menjadi 0,694, sedangkan pada skenario fine-tuning 2 lapisan terakhir, precision meningkat dari

0,594 menjadi 0,708. Peningkatan ini menunjukkan bahwa penggunaan threshold optimal membuat model lebih selektif dalam menentukan label positif. Meskipun nilai recall menurun dibandingkan penggunaan threshold standar 0,5, penurunan tersebut diikuti dengan peningkatan F1-score. Dengan demikian, threshold tuning mampu menghasilkan keseimbangan yang lebih baik antara precision dan recall.

Tabel 4. Hasil Eksperimen Model SciBERT-TextCNN pada Kondisi Threshold 0,5

Skenario	Precision	Recall	F1-Score	Macro F1-Score	Hamming Loss
SciBERT-TextCNN Frozen	0,567	0,829	0,666	0,588	0,03452
SciBERT-TextCNN Fine-tuning 1 lapisan	0,586	0,830	0,680	0,610	0,03230
SciBERT-TextCNN Fine-tuning 2 lapisan	0,594	0,827	0,686	0,617	0,03138
SciBERT-TextCNN Fine-tuning 3 lapisan	0,595	0,829	0,686	0,616	0,03144
SciBERT-TextCNN Fine-tuning 4 lapisan	0,598	0,824	0,687	0,614	0,03111

Pada kondisi setelah threshold tuning, seluruh skenario fine-tuning menghasilkan performa yang lebih baik dibandingkan skenario frozen. Skenario SciBERT-TextCNN frozen memperoleh F1-score sebesar 0,701 dan macro F1-score sebesar 0,631. Setelah sebagian lapisan SciBERT dibuka untuk dilatih kembali, nilai F1-score dan macro F1-score meningkat. Skenario fine-tuning 1 lapisan terakhir memperoleh F1-score sebesar 0,712 dan macro F1-score sebesar 0,646, sedangkan skenario fine-tuning 2 lapisan terakhir memperoleh F1-score tertinggi sebesar 0,715 dan macro F1-score tertinggi sebesar 0,649. Perbandingan visual dari metrik precision, recall, F1-score, dan macro F1-score antar skenario pada kondisi threshold optimal ini juga diilustrasikan secara lebih jelas pada Gambar 3.

Tabel 5. Hasil Eksperimen Model SciBERT-TextCNN pada Kondisi Threshold Optimal

Skenario	Precision	Recall	F1-Score	Macro F1-Score	Hamming Loss
SciBERT-TextCNN Frozen	0,694	0,710	0,701	0,631	0,02401
SciBERT-TextCNN Fine-tuning 1 lapisan	0,707	0,721	0,712	0,646	0,02330
SciBERT-TextCNN Fine-tuning 2 lapisan	0,708	0,726	0,715	0,649	0,02333
SciBERT-TextCNN Fine-tuning 3 lapisan	0,708	0,722	0,713	0,644	0,02302
SciBERT-TextCNN Fine-tuning 4 lapisan	0,705	0,726	0,713	0,644	0,02339

Hasil tersebut menunjukkan bahwa pembukaan dua lapisan terakhir SciBERT memberikan keseimbangan terbaik antara kemampuan adaptasi representasi kontekstual dan kestabilan proses pelatihan. Pada skenario ini, model dapat menyesuaikan representasi teks ilmiah terhadap karakteristik dataset yang digunakan tanpa membuka terlalu banyak parameter SciBERT. Dengan demikian, fine-tuning terbatas pada lapisan akhir dapat meningkatkan performa klasifikasi dibandingkan penggunaan SciBERT secara frozen.

Skenario fine-tuning 3 lapisan terakhir memperoleh F1-score sebesar 0,713 dan macro F1-score sebesar 0,644. Meskipun nilainya sedikit lebih rendah dibandingkan fine-tuning 2 lapisan terakhir, skenario ini menghasilkan Hamming Loss paling rendah, yaitu 0,02302. Hal ini menunjukkan bahwa fine-tuning 3 lapisan terakhir mampu menekan kesalahan prediksi label pada keseluruhan pasangan

data dan label. Namun, jika dilihat dari metrik utama berupa F1-score dan macro F1-score, skenario ini belum melampaui fine-tuning 2 lapisan terakhir.

Tabel 6. Perbandingan Parameter dan Waktu Pelatihan Model SciBERT-TextCNN

Skenario	Trainable Parameter	Epoch	Waktu Pelatihan
SciBERT-TextCNN Frozen	1.825.064	25/25	8 jam 9 menit
SciBERT-TextCNN Fine-tuning 1 lapisan	8.912.936	25/25	9 jam 33 menit
SciBERT-TextCNN Fine-tuning 2 lapisan	16.000.808	25/25	10 jam 36 menit
SciBERT-TextCNN Fine-tuning 3 lapisan	23.088.680	23/25	10 jam 55 menit
SciBERT-TextCNN Fine-tuning 4 lapisan	30.176.552	23/25	11 jam 31 menit

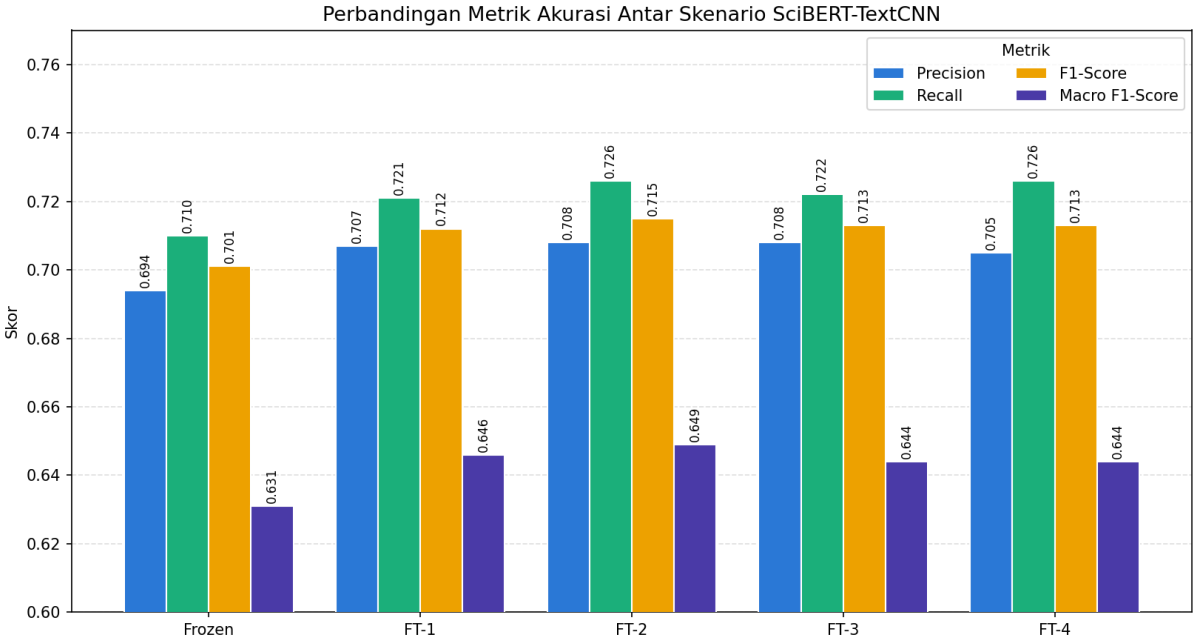
Sementara itu, skenario fine-tuning 4 lapisan terakhir memperoleh F1-score sebesar 0,713 dan macro F1-score sebesar 0,644. Nilai tersebut tidak lebih tinggi dibandingkan skenario fine-tuning 2 lapisan terakhir. Hal ini menunjukkan bahwa membuka lebih banyak lapisan SciBERT tidak selalu menghasilkan peningkatan performa yang lebih baik. Penambahan jumlah lapisan yang dilatih dapat meningkatkan kapasitas model, tetapi juga menambah kompleksitas pelatihan dan risiko model menjadi terlalu menyesuaikan diri terhadap data latih.

Tabel 7. Classification Report Lengkap per Label pada Skenario Terbaik (Fine-tuning 2 Lapisan, Threshold Optimal)

Kategori	Precision	Recall	F1-Score	Support
cs.AI	0,492	0,632	0,553	2889
cs.AR	0,738	0,676	0,706	259
cs.CC	0,571	0,602	0,586	259
cs.CE	0,374	0,458	0,412	260
cs.CG	0,799	0,688	0,740	260
cs.CL	0,879	0,857	0,868	1772
cs.CR	0,822	0,828	0,825	838
cs.CV	0,894	0,878	0,886	3254
cs.CY	0,468	0,587	0,521	492
cs.DB	0,709	0,638	0,672	260
cs.DC	0,612	0,697	0,651	613
cs.DL	0,876	0,735	0,799	260
cs.DM	0,455	0,593	0,515	295
cs.DS	0,675	0,596	0,633	547
cs.ET	0,575	0,592	0,583	260
cs.FL	0,818	0,658	0,729	260
cs.GL	0,167	0,227	0,192	22
cs.GR	0,520	0,542	0,531	260
cs.GT	0,685	0,799	0,738	259
cs.HC	0,633	0,675	0,653	507
cs.IR	0,558	0,619	0,587	443
cs.IT	0,827	0,820	0,823	959
cs.LG	0,728	0,781	0,754	4560
cs.LO	0,667	0,769	0,714	419
cs.MA	0,519	0,521	0,520	259
cs.MM	0,410	0,440	0,425	259
cs.MS	0,760	0,629	0,688	251
cs.NA	0,802	0,748	0,774	638
cs.NE	0,462	0,521	0,489	334
cs.NI	0,728	0,624	0,672	510
cs.OH	0,648	0,412	0,504	228
cs.OS	0,652	0,625	0,638	120
cs.PF	0,385	0,454	0,417	262
cs.PL	0,692	0,631	0,660	260
cs.RO	0,831	0,782	0,806	865
cs.SC	0,766	0,707	0,735	273
cs.SD	0,796	0,853	0,824	361
cs.SE	0,745	0,692	0,718	461

cs.SI	0,731	0,693	0,712	427
cs.SY	0,687	0,697	0,692	786
micro avg	0,696	0,726	0,710	26501
macro avg	0,654	0,649	0,649	26501
weighted avg	0,708	0,726	0,715	26501
samples avg	0,748	0,783	0,729	26501

Selain performa evaluasi, jumlah parameter yang dilatih dan waktu pelatihan juga perlu diperhatikan. Perbandingan jumlah trainable parameter dan waktu pelatihan pada setiap skenario ditunjukkan pada Tabel 6.



Gambar 3. Perbandingan Precision, Recall, F1-Score, dan Macro F1-Score Antar Skenario pada Kondisi Threshold Optimal

Berdasarkan Tabel 6, semakin banyak lapisan SciBERT yang dibuka untuk fine-tuning, semakin besar jumlah parameter yang dilatih. Skenario frozen hanya melatih 1.825.064 parameter, sedangkan skenario fine-tuning 4 lapisan terakhir melatih 30.176.552 parameter. Peningkatan jumlah parameter tersebut juga berdampak pada peningkatan waktu pelatihan. Skenario frozen membutuhkan waktu pelatihan paling singkat, yaitu 8 jam 9 menit, sedangkan skenario fine-tuning 4 lapisan terakhir membutuhkan waktu paling lama, yaitu 11 jam 31 menit.

Walaupun fine-tuning 4 lapisan terakhir melatih parameter paling banyak, performanya tidak lebih baik dibandingkan fine-tuning 2 lapisan terakhir. Skenario fine-tuning 2 lapisan terakhir menghasilkan F1-score dan macro F1-score tertinggi dengan jumlah parameter dan waktu pelatihan yang masih lebih rendah dibandingkan skenario fine-tuning 3 dan 4 lapisan terakhir. Oleh karena itu, skenario fine-tuning 2 lapisan terakhir dapat dianggap sebagai skenario terbaik karena memberikan keseimbangan paling baik antara performa dan efisiensi komputasi.

Berdasarkan rincian laporan klasifikasi pada Tabel 7, analisis per label pada skenario terbaik menunjukkan bahwa model mampu mengenali beberapa kategori dengan performa tinggi. Label cs.CV memperoleh F1-score sebesar 0,886, cs.CL sebesar 0,868, cs.CR sebesar 0,825, cs.SD sebesar 0,820, dan cs.DL sebesar 0,799. Performa tinggi pada label-label tersebut menunjukkan bahwa model dapat mengenali pola teks yang cukup kuat dan konsisten pada beberapa kategori Computer Science. Kategori seperti cs.CV dan cs.CL umumnya memiliki istilah teknis yang lebih khas, sehingga lebih mudah dibedakan dari kategori lainnya.

Namun, masih terdapat label dengan performa rendah, terutama label yang memiliki jumlah data terbatas. Salah satu label yang paling sulit dikenali adalah cs.GL dengan F1-score sebesar 0,192 dan support sebanyak 22 data. Rendahnya performa pada label ini menunjukkan bahwa jumlah data yang sangat kecil tetap menjadi tantangan dalam klasifikasi multilabel, meskipun telah dilakukan ML-RUS dan threshold tuning. Selain itu, beberapa label juga memiliki karakteristik teks yang saling beririsan, sehingga model dapat mengalami kesulitan dalam membedakan kategori yang memiliki kemiripan topik.

Secara keseluruhan, hasil eksperimen menunjukkan bahwa kombinasi SciBERT dan TextCNN mampu digunakan untuk klasifikasi multilabel artikel ilmiah berbasis judul dan abstrak. SciBERT berperan dalam membentuk representasi kontekstual dari teks ilmiah, sedangkan TextCNN mengekstraksi pola lokal dari representasi token yang dihasilkan oleh SciBERT. Penerapan fine-tuning pada lapisan akhir SciBERT terbukti meningkatkan performa dibandingkan skenario frozen. Selain itu, threshold tuning per kelas juga memberikan pengaruh penting dalam meningkatkan keseimbangan antara precision dan recall. Berdasarkan hasil pengujian, skenario fine-tuning 2 lapisan terakhir menjadi model terbaik dengan precision 0,708, recall 0,726, F1-score 0,715, macro F1-score 0,649, dan Hamming Loss 0,02333.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan dan mengevaluasi model SciBERT-TextCNN untuk klasifikasi multilabel 40 kategori artikel ilmiah bidang Computer Science berdasarkan metadata judul dan abstrak. Threshold tuning per kelas terbukti memberikan pengaruh penting terhadap peningkatan performa, terutama dalam menyeimbangkan precision dan recall dibandingkan threshold standar 0,5. Dari lima skenario yang diuji, fine-tuning 2 lapisan terakhir SciBERT menjadi skenario terbaik dengan precision 0,708, recall 0,726, F1-score 0,715, macro F1-score 0,649, dan Hamming Loss 0,02333, sehingga menunjukkan keseimbangan terbaik antara performa klasifikasi dan efisiensi komputasi. Skenario dengan lebih banyak lapisan fine-tuning tidak selalu menghasilkan performa lebih tinggi meskipun jumlah parameter dan waktu pelatihan meningkat. Meskipun demikian, masih terdapat keterbatasan pada label dengan jumlah data sangat sedikit, seperti cs.GL, yang memperoleh performa jauh lebih rendah dibandingkan label lainnya.

Daftar Pustaka

[1] National Science Board, "Publications Output: U.S. Trends and International Comparisons," 2023.

[2] D. Bawden and L. Robinson, "Information Overload: An Introduction," in *Oxford Research Encyclopedia of Politics*, Oxford University Press, 2020. doi: 10.1093/acrefore/9780190228637.013.1360.

[3] N. A. Sajid et al., "Single vs. Multi-Label: The Issues, Challenges and Insights of Contemporary Classification Schemes," Applied Sciences, vol. 13, no. 11, p. 6804, Jun. 2023, doi: 10.3390/app13116804.

[4] Autumn Toney and James Dunham, "Multi-label Classification of Scientific Research Documents Across Domains and Languages," in Proceedings of the Third Workshop on Scholarly Document Processing, Association for Computational Linguistics, 2022, pp. 105–114.

[5] A. N. Tarekegn, M. Giacobini, and K. Michalak, "A review of methods for imbalanced multi-label classification," Pattern Recognit., vol. 118, p. 107965, Oct. 2021, doi: 10.1016/j.patcog.2021.107965.

[6] S. Li and J. Ou, "Multi-Label Classification of Research Papers Using Multi-Label K-Nearest Neighbour Algorithm," J. Phys. Conf. Ser., vol. 1994, no. 1, p. 012031, Aug. 2021, doi: 10.1088/1742-6596/1994/1/012031.

[7] H. Fallah, P. Bellot, E. Bruno, and E. Murisasco, "Adapting Transformers for Multi-Label Text Classification," 2022. [Online]. Available: http://ceur-ws.org

[8] Y. Zhang, X. Li, Y. Liu, A. Li, X. Yang, and X. Tang, "A Multilabel Text Classifier of Cancer Literature at the Publication Level: Methods Study of Medical Text Classification," JMIR Med. Inform., vol. 11, p. e44892, Oct. 2023, doi: 10.2196/44892.

[9] N. Ahmed Sajid et al., "A Novel Metadata Based Multi-Label Document Classification Technique," Computer Systems Science and Engineering, vol. 46, no. 2, pp. 2195–2214, 2023, doi: 10.32604/csse.2023.033844.

[10] D. Likhareva, H. Sankaran, and S. Thiyagarajan, "Empowering Interdisciplinary Research with BERT-Based Models: An Approach Through SciBERT-CNN with Topic Modeling," Apr. 2024.

[11] Z. R. K. Rostam and G. Kertész, "Fine-Tuning Large Language Models for Scientific Text Classification: A Comparative Study," in 2024 IEEE 6th International Symposium on Logistics and Industrial Informatics (LINDI), IEEE, Nov. 2024, pp. 000233--00023. [Online]. Available: http://arxiv.org/abs/2412.00098

[12] S. Ramakrishnan and L. D. Dhinesh Babu, "Improving Multi-Label Emotion Classification on Imbalanced Social Media Data With BERT and Clipped Asymmetric Loss," IEEE Access, vol. 13, pp. 60589–60601, 2025, doi: 10.1109/ACCESS.2025.3557091.

[13] F. Charte, A. J. Rivera, M. J. del Jesus, and F. Herrera, "Addressing imbalance in multilabel classification: Measures and random resampling algorithms," Neurocomputing, vol. 163, pp. 3–16, Sep. 2015, doi: 10.1016/j.neucom.2014.08.091.

[14] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1903.10676

[15] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

[16] T. Ridnik et al., "Asymmetric Loss For Multi-Label Classification," 2021. [Online]. Available: https://github.com/Alibaba-MIIL/ASL.