

Artikel

Analisis Empiris tentang Keadaan Model Klasifikasi Seni dalam TI Prediksi Tingkat Keparahan Insiden Kerangka

Salman Ahmed, Muskaan Singh, Brendan Doherty, Effirul Ramelan, Kathryn Harkin, Magda Bucholc dan Damien Coyle

Edisi Khusus

Mesin/Pembelajaran Mendalam: Aplikasi, Teknologi, dan Algoritme –

Diedit oleh

Prof. Dr. Emilio Soria-Olivas



<https://doi.org/10.3390/app13063843>



Artikel

Analisis Empiris Model Klasifikasi Terkini di Kerangka Prediksi Keparahan Insiden TI

Salman Ahmed 1,*[†], Muskaan Singh 1,[‡], Brendan Doherty 2,[†], Effirul Ramalan 3,[†], Kathryn Harkin 2,[†], Magda Buchole 1,[†] dan Damien Coyle 1,4,*

¹ Pusat Penelitian Sistem Cerdas, Universitas Ulster, Northland Rd, Londonderry BT48 7JL, Inggris

² Data dan Sistem Cerdas, Allstate NI, Belfast BT1 3PH, Inggris

³ Sekolah Ilmu Komputer, Universitas Galway, University Road, H91 TK33 Galway, Irlandia

[†]Institut untuk Manusia Augmented, Universitas Bath, Bath BA2 7AY, Inggris*

Korespondensi: ahmed-s-17@ulster.ac.uk (SA); dh.coyle@ulster.ac.uk (DC)

[‡] Para penulis ini berkontribusi sama pada karya ini.

Abstrak: Perusahaan skala besar di berbagai sektor memelihara infrastruktur TI yang substansial untuk mendukung operasi mereka dan memberikan layanan berkualitas bagi pelanggan dan karyawan mereka. Operasi TI ini dikelola oleh tim yang berurusan langsung dengan laporan insiden (yaitu, yang dihasilkan secara otomatis melalui sistem otomotif atau operator manusia). (1) Latar Belakang: Identifikasi dini insiden besar dapat memberikan keuntungan yang signifikan untuk mengurangi gangguan pada operasi bisnis normal, terutama untuk mencegah gangguan bencana, seperti shutdown sistem total. (2) Metode: Studi ini melakukan analisis empiris dari sebelas (11) model canggih untuk memprediksi tingkat keparahan insiden ini menggunakan kasus penggunaan yang dipimpin industri yang terdiri dari 500.000 catatan yang dikumpulkan selama satu tahun. (3) Hasil: Kumpulan data dihasilkan dari tiga pemangku kepentingan (yaitu, agensi, pelanggan, dan karyawan). Secara terpisah, representasi encoder dua arah dari transformator (BERT), pendekatan pra-pelatihan BERT yang dioptimalkan dengan kuat (RoBERTa), representasi yang ditingkatkan melalui integrasi pengetahuan (ERNIE 2.0), dan metode peningkatan gradien ekstrem (XGBoost) melakukan yang terbaik untuk catatan agensi (93% AUC), sedangkan jaringan saraf konvolusional (CNN) adalah model terbaik untuk sisanya (catatan karyawan pada 95% AUC dan catatan pelanggan pada 74% AUC, masing-masing). Cakrawala prediksi rata-rata adalah sekitar 150 menit, yang signifikan untuk penyebarluasan waktu nyata. (4) Kesimpulan: Studi ini memberikan analisis komprehensif yang mendukung penyebarluasan kecerdasan buatan untuk operasi TI (AIOps), khususnya untuk manajemen insiden dalam organisasi skala besar.



Kutipan: Ahmed, S.; Singh, M.; Doherty, B.; Ramalan, E.; Harkin, K.; Buchole, M.; Coyle, D. Analisis Empiris Model Klasifikasi Terkini dalam Kerangka Prediksi Tingkat Keparahan Insiden TI. *Appl. Sci.* **2023**, *13*, 3843. <https://doi.org/10.3390/app13063843>

Editor Akademik: Emilio Soria-Olivas

Diterima: 10 Januari

2023

Direvisi: 11 Februari

2023

Diterima: 15 Februari

2023

Diterbitkan: 17 Maret 2023



Hak cipta: © 2023 oleh penulis.

Penerima Lisensi MDPI, Basel,

Swiss. Artikel ini adalah artikel akses

terbuka yang didistribusikan

berdasarkan syarat dan ketentuan

lisensi Creative Commons Attribution

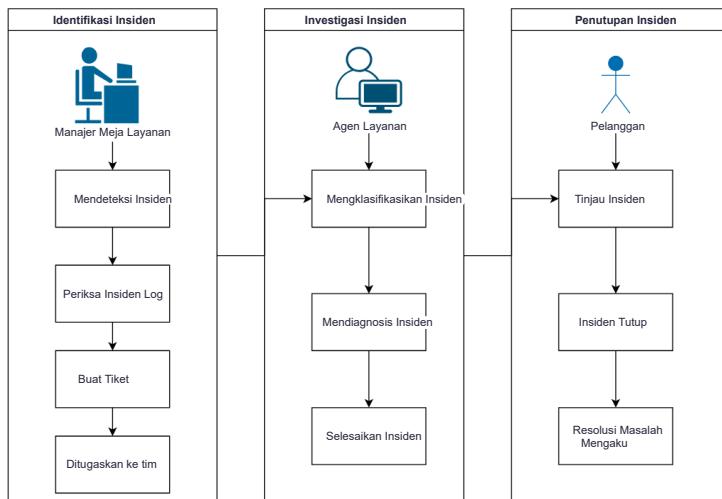
(CC BY)

4.0).

1. Pendahuluan Dalam ISO 20000, pemadaman insiden TI didefinisikan sebagai "Setiap insiden yang bukan bagian dari operasi standar layanan mengganggu atau mengurangi kualitas layanan" [1]. Mengurangi risiko dan ketidakpastian dari semua jenis dan ukuran (baik internal maupun eksternal) sangat penting dalam menentukan keberhasilan bisnis besar apa pun. Misalnya, insiden major yang terkait dengan pemadaman jaringan telah berdampak signifikan pada semua aspek bisnis, termasuk produktivitas karyawan dan kepuasan pelanggan. Laporan (<https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime>, diakses pada 30 Agustus 2022) menunjukkan bahwa di AS, efek insiden besar telah menelan biaya sekitar \$36,326/jam, dan biaya hilir rata-rata untuk bisnis telah menelan biaya tambahan \$105,302/jam. Sayangnya, sekitar 12 miliar laporan insiden dihasilkan setiap hari, khususnya untuk masalah infrastruktur TI, dan ini telah mengganggu bisnis besar [2]. Secara keseluruhan, insiden ini dapat dengan mudah menyebabkan biaya operasional yang tinggi dan, bahkan, berdampak pada reputasi organisasi dalam menangani masalah besar, yang dapat memiliki

efek knock-on pada keberhasilan organisasi secara keseluruhan [3]. Masalah-masalah ini menunjukkan semakin pentingnya deteksi insiden besar dini [4]. Deteksi dini potensi insiden besar dapat meningkatkan sementara waktu mengidentifikasi insiden besar, yang dapat mengurangi waktu penyelesaian dan mengurangi dampak buruk pada operasi bisnis TI [3].

Contoh sistem manajemen insiden TI yang khas disajikan pada Gambar 1.



Gambar 1. Ikhtisar sistem manajemen insiden TI.

Dalam proses manual yang khas, meja layanan TI sepenuhnya bertanggung jawab untuk mengelola insiden besar, mulai dari identifikasi hingga penyelesaian dan penutupan [5]. Karena aspek manualnya, kesalahan manusia secara langsung berdampak pada efisiensi operasi (misalnya, tiket yang salah ditetapkan yang mengakibatkan lebih banyak waktu untuk memproses dan kemungkinan pemborosan sumber daya) [6]. Selain itu, jika terlalu banyak insiden muncul dan antrean prioritas diaktifkan, penetapan tingkat keparahan menjadi kritis, karena insiden yang salah diprioritaskan semakin meningkatkan kompleksitas masalah. Refs. [7] mengidentifikasi bahwa perusahaan dengan sistem manajemen insiden yang dikelola dengan baik telah berhasil meminimalkan kehilangan produktivitas mereka dan mempertahankan kualitas layanan mereka. Oleh karena itu, prioritas insiden sangat penting untuk menentukan dampak dari pemadaman tersebut. Demikian pula, kemampuan untuk memprediksi insiden besar dari laporan insiden awal dapat memberikan keuntungan yang signifikan dalam mengurangi gangguan dan mencegah terjadinya pemadaman aktual. Namun, ini bukan proses yang mudah, dan ada beberapa masalah yang harus ditangani:

1. **Data Tidak Terstruktur Multi-modal:** Laporan insiden sering kali berisi informasi teks dan numerik.
2. **Mencampur Data:** Laporan insiden dengan konten serupa dapat dikaitkan dengan pemadaman yang signifikan atau hanya masalah kecil.

3. Data Tidak Seimbang: Banyak laporan insiden sering terkait dengan insiden kecil, bukan insiden besar.

1.1. Ruang Lingkup dan Definisi Masalah Proses manajemen insiden dimulai segera setelah log tiket insiden dihasilkan. Sebagian besar waktu, pemadaman ini diselesaikan di tingkat organisasi (tergantung ketersediaan sistem) atau melalui komponen sistem (semen sistem tertentu mengeluarkan peringatan). Berdasarkan tingkat keparahan masalah, tiket kemudian diteruskan ke ahli materi pelajaran yang relevan. Namun, proses manajemen layanan TI (ITSM) masih manual, yang mengakibatkan ketidakakuratan dan kerugian finansial (misalnya, backlog karena kesalahan penetapan tingkat keparahan, penggunaan memori disk yang tinggi, konsumsi waktu untuk analisis akar masalah). AIOPS mengatasi masalah dengan mengklasifikasi pemadaman dan merespons situasi dengan tepat. AI digunakan untuk melakukan ini. Kesiitan dan ancaman dunia nyata AIOPS tercermin dalam statistik dari insiden pemadaman TI. Informasi ini dirahasiasikan. Ini beroperasi secara optimal dengan arsitektur tiga tingkat yang menggabungkan sistem, data, dan alat manajemen.

1.2. Kontribusi Dalam makalah ini, kami telah menyediakan kerangka kerja untuk prediksi risiko insiden proaktif. Kerangka kerja ini menjelaskan nilai pembelajaran bagi perusahaan, mencegah dan mengurangi risiko, dan mengoptimalkan sumber daya dengan peningkatan layanan dan kepuasan pelanggan.

- Kami mengusulkan kerangka kerja baru untuk mengotomatiskan prediksi insiden besar untuk mengurangi risiko yang terkait dengan eskalasi insiden. Kami bertujuan untuk mengubah manajemen insiden besar reaktif menjadi proaktif, dalam infrastruktur TI.
- Kami menyajikan berbagai kemungkinan solusi untuk menangani ketidakseimbangan dalam catatan laporan insiden besar (MIR) dan laporan insiden non-besar (NMR) dalam kumpulan data.
- Kami kemudian melakukan analisis komparatif model klasifikasi canggih yang ditandatangani untuk memprediksi insiden TI besar. Tujuan utama organisasi harus mengatur manajemen risiko TI [8]. Mitigasi risiko harus mencakup respons segera, perbaikan yang tepat, dan pemantauan risiko berkelanjutan [9]. Mengingat meluasnya penggunaan AI di berbagai sektor bisnis, hipotesis kerja kami adalah bahwa manajemen insiden TI harus diotomatisasi. Kami menyediakan sistem manajemen layanan otomatis berdasarkan model AI mutakhir, seperti machine learning (ML) dan deep learning (DL). Organisasi besar dapat menggunakan pendekatan ini untuk memungkinkan deteksi dan prioritas peristiwa dini. Sisa makalah ini disusun sebagai berikut: Bagian 2 menyediakan tinjauan literatur, Bagian 3 menjelaskan data, diikuti oleh Bagian 4, yang mengusulkan kerangka kerja baru untuk prediksi risiko insiden dengan deskripsi metode klasifikasi yang canggih. Bagian 5 menjelaskan pengaturan eksperimental, dan Bagian 6 memberikan hasil dan analisis. Akhirnya, Bagian 7 membahas penelitian. Kesimpulan disediakan di Bagian 8.

2. Latar Belakang Terbatas penerapan teknik ML yang khusus untuk klasifikasi insiden TI. Ringkasan studi terbaru tercantum dalam Tabel 1. Dari pengamatan kami, di antara karya-karya terkemuka yang mengadopsi pengklasifikasi ML konvensional untuk insiden TI, support vector machine (SVM) telah menjadi yang paling populer (22 kajian), diikuti oleh Bayes naïf (NB) dengan 13 studi, decision tree (DT) dengan 10 studi, k-tetangga terdekat (KNN) dengan 5 studi, regresi logistik (LR) dengan 4 studi, dan terakhir, hanya 2 studi yang menerapkan hutan acak (RF). Organisasi skala besar memiliki kesederhanaan model ML konvensional ini karena sumber dayanya yang terbatas dalam hal struktur organisasi dan fasilitas komputasi. Misalnya, SVM lebih disukai hanya karena algoritme tidak terlalu menuntut komputasi dan, dengan demikian, secara signifikan mengurangi biaya solusi. Selain itu, ada minat yang kurang bersemangat untuk mengadopsi model jaringan DL untuk klasifikasi IT incident, yang sangat kontras dengan popularitas model ini di sebagian besar industri lainnya. Salah satu masalah potensial adalah kompleksitas implementasi, khususnya

lary kemampuan interpretatif algoritme dan sumber daya komputasi yang dituntut saat melatih model DL. Ada sangat sedikit implementasi DL yang tersedia dalam literatur, dan hanya [10,11] menggunakan jaringan memori jangka pendek panjang (LSTM) [12] dan jaringan saraf konvolusional (CNN) [13] yang dicatat. Seperti yang diharapkan, implementasi metode yang lebih canggih (misalnya, transformator pembelajaran transfer mendalam seperti representasi en-coder dua arah dari transformator (BERT) [14], BERT yang dioptimalkan secara kuat (RoBERTa)[15], dan representasi yang ditingkatkan melalui integrasi pengetahuan (ERNIE 2.0) [16]) bahkan lebih jarang. Kami menemukan satu studi oleh [17] yang mengadopsi BERT dalam model prediksi risiko insiden TI mereka. Terlepas dari algoritme yang mendasarnya, sebagian besar model DL memerlukan serangkaian parameter yang tepat yang telah disesuaikan dengan masalah. Parameter ini sering kali statis dan sering dikodekan keras ke dalam model. Dengan demikian, memperbarui dan memelihara solusi ini rumit dan membutuhkan tim yang berdedikasi [18]. Model DL yang lebih canggih identik dengan kotak hitam, yang relatif rumit untuk ditafsirkan. Karena tingkat abstraksi yang tinggi ini, banyak perusahaan mungkin enggan mempercayai output yang dihasilkan dari model DL, yang dapat mengakibatkan kegagalan inisiatif otomatisasi.

Memahami semantik insiden TI sangat penting untuk membangun konteks keseluruhan korpus. Insiden dalam bentuk mentahnya mengandung batasan kualitas dan kegunaan yang harus dihilangkan menggunakan pemrosesan bahasa alami (NLP) dan pendekatan prapemrosesan [19]. Secara kritis, ketika mempertimbangkan distribusi yang tidak seimbang yang sering dikaitkan dengan kumpulan data real-time, para peneliti telah mengabaikan fitur penting saat mengembangkan korpus mereka. Kami memerlukan saluran prapemrosesan NLP yang ada dan mengamati beberapa tantangan menarik, termasuk penerapan strategi lemmatisasi dan stem selama fase prapemrosesan. Analisis sebelumnya telah mengungkapkan bahwa teknik pemodelan bahasa telah menghasilkan hasil yang lebih baik untuk lemmatisasi daripada untuk stem untuk pengambilan dokumen ketika presisi adalah metrik kinerja. Studi lain oleh [17] membahas masalah mengidentifikasi fitur tekstual yang tepat untuk korpus besar menggunakan pipa pra-proses yang disesuaikan. Namun, alur tidak umum dan memerlukan perubahan substansial, berdasarkan atribut spesifik dari kumpulan data, untuk berfungsi. Terlepas dari kompleksitas pemahaman konteks, teknik NLP telah dikembangkan untuk ekstraksi fitur (seperti yang tercantum pada Tabel 1) untuk menangkap atribut maksimum dari korpus tertentu.

Dalam hal teknik rekayasa fitur tertentu, istilah frekuensi dokumen terbalik frekuensi (TF-IDF) [20] telah menjadi yang paling disukai (12 implementasi untuk studi prediksi tingkat keparahan insiden ditemukan). TF-IDF lebih disukai karena vektorisasi TF-IDF menentukan skor TF-IDF untuk setiap kata dalam korpus dan menetapkan informasi tersebut ke vektor. Akibatnya, setiap dokumen dalam korpus memiliki vektornya sendiri, dan vektor memiliki skor TF-IDF yang ditetapkan untuk setiap kata yang muncul di mana saja dalam koleksi dokumen kapasitudo saja. Vektor mendekati apakah dua teks sebanding atau tidak dengan membandingkan vektor TF-IDF mereka menggunakan metrik kesamaan kosinus. TF-IDF menyediakan cara sederhana untuk menghitung hubungan antara fitur dan kepentingannya dalam teks. Ini adalah memori dan efisiensi secara operasional, yang berkontribusi pada popularitasnya. Pendekatan populer lainnya adalah count vectorizer, metode ekstraksi fitur tradisional berdasarkan pendekatan kantong kata [21]. Dibandingkan dengan TF-IDF, count vectorizer lebih intensif secara komputasi dan seringkali tidak dapat mengidentifikasi kata kunci penting [22]. Terlepas dari popularitas kedua metode tersebut, ada batasan yang jelas, terutama saat menangani kumpulan data berskala besar, tidak terstruktur, dan tidak seimbang yang biasa ditemukan dalam pelaporan insiden TI.

Vektorizer ML konvensional memiliki kosakata yang terbatas, yang tidak cukup untuk kumpulan data besar. Seperti yang ditunjukkan dalam [21], vektorizer konvensional tidak efektif dalam menangani data ITSM dunia nyata karena tidak dapat ditingkatkan. Kosakata MLapproach tradisional tetap dan tidak dapat beradaptasi dengan titik real-time yang akan datang. Sebagian besar vectorizer konvensional juga mahal karena tinjauan kardinalitas yang harus dilakukan secara ekstensif untuk setiap kata. Kelemahan ini telah diatasi dengan mengadopsi metode ekstraksi fitur yang lebih canggih yang ditawarkan oleh model canggih, seperti BERT, RoBERTa, dan ERNIE 2.0. Model Transformer ini dapat menghindari pengulangan, menyediakan penyematan posisi-

dings dan hubungan pembelajaran antar kata-kata. Misalnya, ref. [17] menyajikan hasil yang menunjukkan ekstraktor fitur berbasis transformator menggunakan vektorizer konvensional. Survei yang disajikan di sini mengungkapkan bahwa dalam hal kinerja saat menggunakan akurasi sebagai metrik, model ML konvensional berkinerja jauh lebih baik daripada model DL yang lebih canggih. Misalnya, [10], dalam analisis mereka menggunakan kumpulan data tiket dukungan TI terbuka, menunjukkan bahwa NB mengungguli model LSTM yang lebih canggih (akurasi 74% hingga 69%, masing-masing). Selain itu, ref. [12] menunjukkan model gradient boosting (GB) yang sedikit mengungguli model CNN sebesar 3% dengan skor akurasi 95%–92% untuk mengoptimalkan dataset Information Technology Infrastructure Library (I). Sebaliknya, kami menemukan ini tidak mencerminkan potensi sebenarnya dari model DL canggih ini. Ada masalah penting dalam implementasinya, seperti pelatihan yang tidak memadai karena infrastruktur yang terbatas (sebagian besar metode DL canggih memerlukan pelatihan ekstensif untuk memfasilitasi pembelajaran [23]), kurangnya pemahaman tentang arsitektur DL (sulit untuk mengidentifikasi bias model), kurangnya pengoptimalan hiperparameter (yang memakan waktu dan membutuhkan pemanfaatan memori yang luas), dan kurangnya validasi untuk evaluasi kinerja.

Dalam survei literatur ini, kami menyadari bahwa analisis komparatif yang diperbarui dari algoritma ML dan pendekatan DL canggih, termasuk arsitektur transformator untuk prediksi insiden, diperlukan, karena kosakata yang ditangani oleh pengklasifikasi ML terbatas, membuat ML tidak mencukupi ketika belajar dari kumpulan data yang lebih besar yang kompleks (campuran dan tidak terstruktur) dan tidak seimbang. Oleh karena itu, kami menyajikan analisis komprehensif dalam makalah ini, yang berfokus pada model transformator pertama kali. Kami berhipotesis bahwa transformermodeL dapat secara signifikan meningkatkan metrik kinerja karena mekanisme perhatiannya dan kemampuannya untuk menangani ukuran kosakata yang lebih besar.

Tabel 1. Implementasi prediksi insiden TI otomatis yang ada menggunakan model AI.

Data	Pra-Pemrosesan	Rekayasa Fitur	Teknik	Ref
Data sumber terbuka operator helpdesk Endava	anonimisasi, huruf kecil, lematiasi, modus penghapusan	Tidak	CNN, RF, GB, rata-rata, dan ansambel tumpukan	[24] [1]
Departemen CHM	menghapus kata berhenti, tanda baca, beralih ke huruf kecil, beralih ke huruf kunci, berisi linguistik fitur		TF-IDF dan KNN, pohon keputusan, NB, LR, SVM, QUICKSUCCESS	
Rantai restoran cepat saji	Kolom DateTime bernama transformasi tiket tertutup dari format string ke format DateTime.	Ekstraksi fitur dilakukan berdasarkan data horian. Fitur yang tidak benarwana dihapus. Fitur ini dipilih berdasarkan teori probabilitas.	NB, LR, dan model pohon keputusan peningkatan gradien	[26] [1]
Departemen TI dari perusahaan multinasional besar	tokenisasi, penghapusan kata berhenti dan digit, penggabungan bagian kata, dan pemfilteran bagian dari ucapan dengan hanya memilih kelas tata bahasa berbuka	Jumlah kata per kategori solusi terhadap data teks.	TF-IDF dan Bayes naïf multinomial SVM, KNN, pohon keputusan, dan regresi logistik	[27] [1]
Sistem pemantauan IBM Tivoli	Kekuatan		Model HMDB dan ICTR	[28]
Universitas Yordania Jerman.	menghapus tag HTML dan karakter khusus	Vektorisasi fitur TF-IDF	SVM, NB, berbasis aturan, dan pohon keputusan	[29]
Data insiden Infrastruktur TI	Hapus kata berhenti, karakter khusus, tanggal dan waktu, nomor telepon, dan alamat email	χ^2 -squared digunakan untuk memilih fitur penting menggunakan vektor TFIDF. 1000 fitur penting teratas yang kemudian dipilih.	NB, SVM, dan Ada SVM	[30] [1]
Himpunan data IBM CMDB	kata kunci dan anotasi sebagai fitur klasifikasi	Memilih catatan item Konfigurasi teratas dari CMDB untuk kerumitan pra-vent. TF-IDF digunakan untuk memementingkan	SVM	[31] [1]
Sistem pelacakan masalah insiden data Universitas Teknis Istanbul	pemurnian tiket dari HTML dan tag ekspresi numerik dilakukan	Kantong kata menggunakan TF-IDF	pohon keputusan, SVM, KNN, dan NB	[32] [1]
Data rahasia Perusahaan TI	tokenisasi, penghapusan kata henti, dan penghentian	bagian dari penandaan ucapan dilakukan untuk menyaring kosakata SVM		[19]
Data tiket insiden dunia nyata	tokenisasi, penghapusan dan penghentian kata henti	TFIDF	SVM	[33]
Data tiket mejah layanan infrastruktur TI dunia nyata menghapus kata berhenti, karakter khusus, tanggal dan waktu, nomor telepon, dan alamat email		TF-IDF	LR, K-NN, MNB dan SVM	[34] [1]
Data tiket organisasi	lemmatiasi, penandaan POS	TFIDF dengan filtrasi koefisien Jaccard	k-berarti pengelompokan, jarak Jaccard, dan jarak kosinus	[35]
Repositori ML UCI	Tidak	Embedding	jaringan konvolusional grafik relasional	[36]
Data studi kasus	Tidak	TFIDF	SVM menggunakan kernel RBF dan XGBoost	[37]
Himpunan data real-time IBM	HTML na Pengha penggantian header/footer/entities	pengambilan sampel peran semantik	BERT	[17] [1]
Himpunan data tiket masalah telekomunikasi	Hapus tanda baca dan kata berhenti	TFIDF	hutan acak, DL, peningkatan gradien, XGBoost, dan pengklasifikasi pohon acak yang luar biasa	[38] [1]
Himpunan data Layanan Sekarang di mejah bantuan	Hapus tanda baca dan kata berhenti	Doc2vec menggunakan sistem tiket ServiceNow	regresi logistik	[18]
Himpunan data tiket perjanjian tingkat layanan	hapus tanda baca dan kata henti, URL..	Hitung Vektor	pohon keputusan, SVM, KNN dan NB	[39] [1]

3. Deskripsi Dataset Untuk penelitian ini, kami menggunakan kumpulan data dari perusahaan multinasional besar yang terdiri dari 500.000 total catatan. Himpunan data terdiri dari insiden TI real-time yang dikumpulkan dari Januari 2020 hingga Maret 2021. Insiden ini dicatat oleh tiga pemangku kepentingan utama organisasi (agensi, karyawan, dan pelanggan). Insiden yang dilaporkan diklasifikasikan sebagai laporan insiden besar (MIR) atau laporan insiden non-besar (NMIR). Contoh laporan insiden disediakan di bawah ini:

- MIR: "Mengapa saya tidak dapat mencetak dari aplikasi apa pun? Semua staf tidak dapat mencetak. Pesan Noerror."

Non-MIR: "Bagaimana cara memperbarui Java? Versi web Outlook berjalan lambat."

MIR adalah laporan insiden yang telah diekskalasi ke status utama atau memiliki hubungan langsung ke insiden besar yang diidentifikasi oleh tim TI. Sebaliknya, NMIR adalah insiden reguler dengan dampak yang lebih kecil pada operasi. Kelas setiap insiden diberi label retrospektif setelah diproses. Himpunan data disusun sebagai:

$$\text{Agency_records} = \text{Badan} \sum (\text{MIR}, \text{NMIR}),$$

$$\text{Employee_records} = \sum \text{Karyawan} (\text{MIR}, \text{NMIR}),$$

$$\text{Customer_records} = \sum \text{Pelanggan} (\text{MIR}, \text{NMIR})$$

Distribusi kumpulan data digambarkan pada Gambar 2. Agency_records berisi 493.503

insiden, di mana 15.257 adalah MIR dan 478.246 adalah catatan NMIR.

TheEmployee_records berisi 245.696 insiden dengan 3779 nilai MIR dan 241.917 NMIR. TheCustomer_records memiliki 217.540 insiden dengan 173 sebagai MIR dan 217.367 sebagai NMIR. Seperti yang diamati, kumpulan data secara signifikan tidak seimbang dan sangat condong mendukung insiden NMIR.



Gambar 2. Distribusi kumpulan data.

Khusus untuk penelitian ini, kami hanya menggunakan Deskripsi, Deskripsi singkat (detail insiden), dan Status (label MIR atau NMIR) dalam kolom kumpulan data yang tersedia. Selain itu, kami memanfaatkan kolom MIR yang dibuka di (MOA) dan Insiden yang dibuka di (IOA) untuk memperkirakan cakrawala prediksi (yaitu, keuntungan waktu yang dapat diwujudkan dengan menyelidiki penyebab insiden yang berpotensi terkait dengan pemadaman besar). Contoh himpunan data dijelaskan dalam Tabel 2. Untuk mengatasi masalah kelas yang tidak seimbang dalam kumpulan data kami, kami mengkursi kumpulan data baru menggunakan pendekatan augmentasi data khusus [40]. Kami menambahkan catatan MIR aktual dari

kumpulan data pemangku kepentingan lainnya ke setiap kumpulan data pemangku kepentingan. Ini memperkaya setiap kumpulan data sambil mencegah informasi asing tambahan dalam kosakata. Augmentasi data dimaksudkan untuk meningkatkan terjadinya insiden klasifikasi MIR yang sangat kurang dalam kumpulan data asli. Komposisi dataset adalah, sebagai berikut:

$$\text{Agency_resampled} = \sum \text{Agency_records} + \sum \text{Karyawan (MIR)} + \sum \text{Pelanggan (MIR)},$$

$$\text{Employee_resampled} = \sum \text{Employee_records} + \sum \text{Agen (MIR)} + \sum \text{Pelanggan (MIR)},$$

$$\text{Customer_resampled} = \sum \text{Customer_records} + \sum \text{Karyawan (MIR)} + \sum \text{Agensi (MIR)}$$

Kami juga menggabungkan semua catatan untuk meningkatkan jumlah catatan MIR, sebagai berikut:

$\text{Combine_All} = \sum (\text{Agency_records} + \text{Employee_records} + \text{Customer_records})$

Kami juga melakukan up-sampling menggunakan teknik oversampling minoritas sintetis (SMOTE) [41]. Kami menggunakan SMOTE untuk oversampling karena menghasilkan titik data sintetis yang berbeda dari titik aktual alih-alih menduplicasi catatan tanpa informasi tambahan dan meningkatkan ukuran vektor. Dengan menggunakan SMOTE, kami menghasilkan tiga subset sampel:

$$\text{Agency_smote} = \sum \text{SMOTE(Agenzi (MIR, NMIR))},$$

$$\text{Employee_smote} = \sum \text{SMOTE(Karyawan(MIR, NMIR))},$$

$$\text{Customer_smote} = \sum \text{SMOTE(Pelanggan(MIR, NMIR))}$$

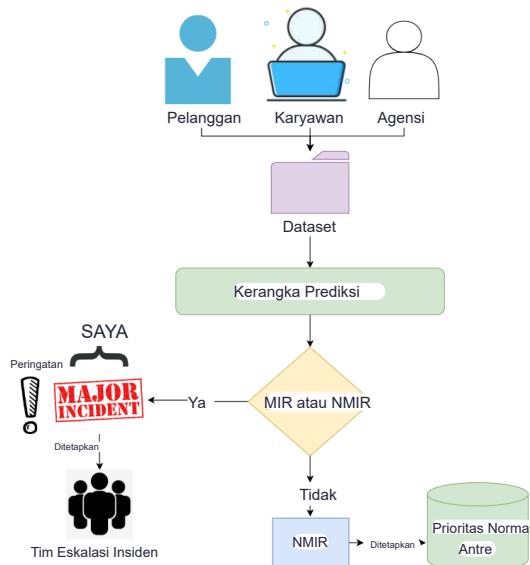
Tabel 2. Kamus data.

ID Kolom	Deskripsi	Nilai
Nomor insiden	Kode internal unik dari insiden tersebut	INC123xxxx
Grup tugas	Kelompok tempat insiden itu telah ditugaskan.	
Dibuka di	Tanggal/Stempel Waktu saat membuat rekaman insiden.	17 Maret 2020
Tutup pada	Tanggal/Stempel waktu saat catatan insiden ditutup.	18 Maret 2020
Tingkat keparahan insiden	Tingkat dampak untuk setiap insiden.	(1—Tinggi; 2—Sedang; 3—Rendah; 4—Tidak ada)
CMDB	Nama database manajemen konfigurasi yang terkait dengan insiden tersebut	
Golongan	Kategori yang terkait dengan insiden tersebut	
Deskripsi singkat	Informasi singkat tentang insiden tersebut.	
Deskripsi	Informasi terperinci tentang insiden tersebut.	
Keadaan	Pemetaan manual dari masalah ke insiden.	(0—MIR; 1—Non-MIR).
MIR dibuka pada	Tanggal/stempel waktu saat membuat catatan MIR.	17 Maret 2020
Insiden dibuka pada	Stempel tanggal/waktu kapan catatan insiden dibuka.	18 Maret 2020

Survei literatur kami menunjukkan bahwa sebagian besar penelitian yang dilaporkan tidak menggunakan resam-pling, yang mungkin menghasilkan pembelajaran yang tidak optimal [19,29,37]. Kami berhipotesis bahwa, dengan meningkatkan kelas minoritas (yaitu, dalam situasi yang sangat tidak seimbang yang diamati dalam kumpulan data kami), kami dapat meningkatkan ekstraksi fitur kelas minoritas (dalam hal ini, laporan MIRclassifed), yang akan mengarah pada akurasi yang lebih tinggi. Dengan melakukan SMOTE, kami berhasil meningkatkan kejadian catatan kelas minoritas (MIR). Untuk Agency_smote, kami menghasilkan 462.989 catatan MIR, yang ditambahkan ke 478.246 catatan NMIR yang ada. Untuk Employee_smote, kami menghasilkan 238.138 catatan MIR, yang ditambahkan ke 241.917 catatan NMIR yang ada. Untuk Customer_smote, kami menghasilkan 217.194 catatan MIR, yang ditambahkan ke 217.367 catatan NMIR yang ada.

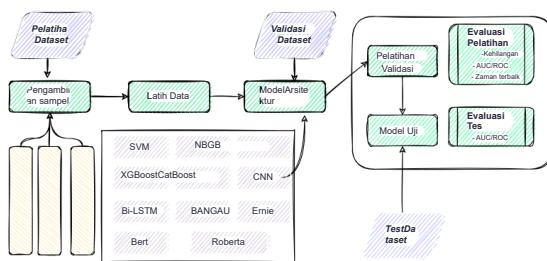
4. Kerangka Kerja yang Diusulkan
 Saat ini, ada kesenjangan dalam literatur mengenai evaluasi kinerja algoritma AI dalam domain ini. Dalam penelitian ini, kami mengusulkan kerangka kerja komprehensif yang

membandingkan berbagai pendekatan ML dan DL, khususnya untuk prediksi insiden TI, menggunakan data tingkat industri milik kami. Kami mengembangkan alur komputasi yang dimulai dengan prapemrosesan, diikuti dengan ekstraksi fitur, pelatihan, dan evaluasi (deskripsi terperinci tentang alur disajikan di Bagian 5.1) dan ditunjukkan pada Gambar 3.



Gambar 3. Kerangka kerja risiko insiden TI yang diusulkan.

Pipa komparatif kami ditunjukkan pada Gambar 4. Fase pertama dari pipeline adalah persiapan data. Seperti yang dibahas sebelumnya di Bagian 3, kami telah melakukan augmentasi data dan pengambilan sampel ulang sintetis pada kumpulan data asli kami. Kumpulan data ini dibagi (70% untuk pelatihan dan 30% untuk pengujian). Dengan menggunakan data pelatihan, kami telah melakukan pelatihan yang diawali menggunakan semua 11 pengklasifikasi. Khusus untuk pengklasifikasi berbasis DL, kami mengalokasikan 20% dari trainingdataset kami untuk validasi pelatihan menggunakan fungsi fit model Keras default. Fase pelatihan dilakukan selama 10 zaman dengan `binary_crossentropy` sebagai fungsi kerugian dan AUCas sebagai metrik kinerja. Kami memilih AUC sebagai metrik untuk mencegah pemasangan yang berlebihan karena karakteristik dataset kami yang tidak seimbang. Akhirnya, model diuji dengan data pengujian (30% dari kumpulan data). Demikian pula, skor AUC ditentukan untuk mengevaluasi kinerja semua 11 pengklasifikasi.



Gambar 4. Representasi grafis dari tahapan pipa untuk pelatihan dan pengujian untuk model ML, DL, dan transformator.

Elemen penting dari kerangka prediksi adalah pemanfaatan berbagai pengklasifikasi canggih. Dalam pekerjaan ini, kami mengkategorikan pengklasifikasi ini sebagai (1) metode ML konvensional, termasuk NB, SVM, gradient boosting (GB), extreme gradient boosting (XGBoost), dan categorical boosting (CatBoost); (2) pendekatan berbasis DL, termasuk gated recurrentunit (GRU), CNN, dan Bi-LSTM; dan terakhir, (3) transformator, termasuk BERT, BERT yang dioptimalkan dengan kuat (RoBERTa), dan representasi yang disingkatkan melalui integrasi pengetahuan (ERNIE 2.0). Deskripsi singkat tentang setiap pengklasifikasi disediakan di bawah ini.

- NB [42]: Pengklasifikasi Bayes naïf didasarkan pada teorema Bayes dan merupakan salah satu algoritma ML yang populer untuk klasifikasi. NB adalah model probabilistik sederhana yang mengasumsikan bahwa setiap fitur atau variabel dari kelas yang sama memberikan kontribusi independen dan setara terhadap hasil. Kumpulan data kami dibagi menjadi matriks fitur dan vektor target. Matriks fitur (X) berisi semua vektor (baris) kumpulan data, di mana setiap vektor terdiri dari nilai fitur dependen. Kami berasumsi bahwa jumlah fiturnya adalah d , $X = (x_1, x_2, x_3, \dots, x_d)$ sedangkan vektor target (y) berisi nilai variabel kelas untuk setiap baris, sesuai dengan matriks fiturnya. Matriks fitur (X) adalah deskripsi insiden yang dilaporkan dalam sistem database. Nilai target (y) adalah MIR atau NMIR. Dalam hal ini, $P(y|X)$ adalah probabilitas kelas y (forMIR atau NMIR), menurut deskripsi insiden (X). Fungsi probabilitas maksimum menyediakan label klasifikasi, baik MIR atau NMIR.
- GB [43]: Pengklasifikasi penambahan gradien adalah pendekatan yang dikembangkan untuk melatih hy-pothesis lemah secara berulang untuk sampai pada hipotesis yang lebih baik. Peningkatan gradien menggabungkan model sebelumnya dengan model yang dihasilkan berulang untuk meminimalkan kesalahan prediksi. Ini meminimalkan fungsi kerugian secara berulang, dimulai dengan gradien negatif, yaitu hipotesis lemah. Dalam pekerjaan ini, kami melakukan klasifikasi biner (MIR atau NMIR) menggunakan kolom deskripsi insiden. Algoritma dimulai dengan satu simpul daun yang memprediksi nilai awal untuk setiap deskripsi insiden. Selanjutnya, algoritme menggunakan log (peluang) dari nilai target, menghasilkan nilai kelangsungan hidup rata-rata yang ditampilkan ke simpul daun awal kami. Jika probabilitas bertahan hidup melebihi 0,5, pertama-tama kami mengklasifikasikan setiap sampel dalam kumpulan data pelatihan sebagai MIR. (Catatan: di mana 0,5 adalah nilai ambang batas umum yang terkait dengan keputusan klasifikasi biner berdasarkan probabilitas).
- XGBoost [44]: Peningkatan gradien ekstrem adalah teknik GB yang dioptimalkan yang memberikan hasil pohon yang efisien, fleksibel, dan portabel. Ini menawarkan peningkatan pohon paralel yang memberikan solusi dengan cepat dan akurat. XGBoost efisien untuk implementasi sumber terbuka dan lingkungan yang didistribusikan secara signifikan, seperti Hadoop, SunGrid Engine (SGE), dan Message Passing Interface (MPI). Ini memprediksi sisa atau kesalahan model sebelumnya untuk mendapatkan prediksi akhirnya. Algoritma penurunan gradien meminimalkan kerugian saat menambahkan model baru. Kami melatih model dengan 500 pohon dan

- kedalaman (max_depth) 1 (untuk simpul akar). Dibandingkan dengan teknik peningkatan lain yang tersedia, misalnya, GB, XGBoost cepat, hemat memori, dan sangat akurat [45].
- Catboost [46]: Kategori dan peningkatan mengadopsi pengambilan sampel varians minimal, sejenis, peningkatan gradien stokastik dengan pengambilan sampel berbobot. Dalam hal ini, weightedsampling terjadi pada tingkat pohon dan bukan tingkat split. Tumbuh sebagai pohon yang selimbing; Skor split meminimalkan kerugian dan memaksimalkan skor akurasi. Mengubah parameter kebijakan (dengan fungsi penalti di node level) juga dimungkinkan. Berbeda dengan pendekatan tradisional, penelitian ini membahas masalah klasifikasi MIR dan NMIR. Kami melatih model dengan 400 pohon dan kedalaman (max_depth) 1 (hanya simpul akar). Dibandingkan dengan XGBoost, Catboost dua kali lebih cepat dengan akurasi yang lebih baik [47].
 - SVM [48]: Mesin vektor pendukung menggunakan pembelajaran terkait untuk menganalisis data untuk klasifikasi. Berdasarkan deskripsi insiden yang diketahui, SVM membangun model yang menetapkan label MIR atau NMIR untuk insiden baru. Algoritme mempertimbangkan semua deskripsi insiden dan memetakannya ke ruang yang memaksimalkan jarak antara keduai kelas. Dalam konteks penelitian ini, kami memiliki kumpulan data pelatihan deskripsi insiden berlabel "0" untuk MIR dan "1" untuk NMIR, $(x_1, y_1) \dots (x_n, y_n)$ di mana y_i adalah 0 atau 1, masing-masing menunjukkan titik x_i , di mana setiap x_i adalah vektor n -dimensi. Kami tertarik pada hyperplane margin maksimum yang membagi kelompok titik x_i , di mana $y_i = 0$ dari kelompok titik di mana $y_i = 1$, yang didefinisikan sehingga jarak antara hyperplane dan titik terdekat x_i dari salah satu kelompok dimaksimalkan. Setiap hyperplane dapat dituliskan sebagai himpunan titik x , memuaskan $w^T x - b = 0$, di mana w adalah vektor normal (tidak harus dinormalisasi) ke hyperplane. Ini mirip dengan bentuk normal Hesse, kecuali bahwa w belum tentu merupakan vektor satuan. Parameter w menentukan offset hyperplane dari asal di sepanjang vektor normal w .
 - Bi-LSTM [49]: Memori jangka pendek jangka panjang dua arah adalah versi yang ditingkatkan dari model LSTM (varian dari arsitektur DL rekursif) yang dapat memproses data dalam arah maju dan mundur. Bi-LSTM membantu memetakan model yang memungkinkan dependensi berurutan dalam kata dan frasa. Untuk kerangka kerja kami, Bi-LSTM membantu dalam melatih urutan MIR dan NMIR dengan melestarikan informasi ini menggunakan dua sel RNN independen yang menyimpan keadaan berulang untuk waktu yang lebih lama. Bi-LSTM berisi input insiden TI (X_t) yang digabungkan dengan status tersembunyi ($h_t = 1$), yang selanjutnya diteruskan ke tiga gerbang (Forget, Input, dan Output). Gerbang Input memerlukan gerbang tertanam yang dikenal sebagai Pembiaran. Gerbang Pembiaran menghafal urutan kalimat masa lalu dan sekarang. Nilai dari gerbang pembiaran mengalihkan status sel ($C_t = 1$), menghasilkan status atau unit tersembunyi (h_t). Arsitektur Bi-LSTM kami terdiri dari lima lapisan: satu lapisan penyematan dengan ukuran 300, satu lapisan dua arah dengan 280 neuron LSTM, dua lapisan putus, dan satu lapisan klasifikasi (lapisan padat). Jumlah total parameter yang dapat dilatih untuk model ini adalah 23.346.401.
 - CNN [50]: Jaringan saraf konvolusional identik dengan klasifikasi gambar, tetapi baru-baru ini, telah berkontribusi secara signifikan pada NLP. CNN digunakan untuk mengekstrak fungsi fitur tingkat tinggi dari n-gram. Misalnya, ref. [51] mengembangkan lapisan matriks penyematan kata yang menghafal bobot selama fase pelatihan jaringan. Kami menggunakan label input kerangka kerja kami (MIR atau NMIR) sebagai token untuk dipetakan ke word-embeddingmatrix. Dalam pemetaan ini, setiap filter konvolisional dipetakan ke setiap jendela lapisan penyematan. Di CNN, reduksi matriks mengurangi dimensi matriks menjadi panjang konstan. Kami mengakses kunci pengurangan matriks pada setiap jendela yang memungkinkan. Setiap matriks yang direduksi adalah input untuk lapisan yang terhubung sepenuhnya. Di lapisan berikutnya, kami menggunakan fungsi aktivasi untuk membuat input dimensi tunggal dari fitur per output tensor; Lapisan pengumpulan maks global dengan 256 ukuran batch dan 23.027.144 parameter yang dapat dilatih digunakan. Di lapisan terakhir, kami menerapkan fungsi aktivasi Softmax untuk menerjemahkan nilai probabilitas nyata ke dalam label MIR atau NMIR.
 - GRU [52]: Unit berulang berpagar adalah varian RNN DL yang menyimpan informasi untuk waktu yang lebih lama dan meningkatkan kecepatan komputasi. GRU terdiri dari dua gerbang saraf

(Perbarui dan Atur Ulang gerbang) untuk memperbarui status sel sebelumnya dan membuat status yang relevan. Untuk tujuan ini, kami menyediakan nilai MIR atau NMIR sebagai input X_t , yang digabungkan dengan status tersembunyi H_{t-1} dan dipindahkan ke gerbang Pembaruan. Pada fase akhir, kami menggunakan fungsi aktivasi Sigmoid untuk mempertahankan output dalam kisaran (0,1), menghasilkan nilai MIR atau NMIR. Arsitekturnya terdiri dari 5 lapisan, berisi satu lapisan penyematan dengan ukuran 300 vektor, satu lapisan GRU dengan 140 neuron, dan satu lapisan padat dengan fungsi aktivasi sigmoid. Model ini memiliki total 23.037.981 parameter yang dapat dilatih.

- BERT [14]: Representasi encoder dua arah dari transformator terdiri dari beberapa transformator encoder dalam kumpulan model yang telah dilatih sebelumnya. BERT mengikuti orientasi bidi-directional dari pembelajaran informasi dari urutan kata dari kiri ke kanan dan kanan ke kiri. Setiap encoder merangkum dua sub-lapisan: lapisan perhatian diri dan lapisan umpan ke depan. Kami telah menggunakan arsitektur BERT terlatih yang berisi 12 lapisan encoder, 12 kepala perhatian, 768 ukuran tersembunyi, dan 110M parameter yang dapat dilatih. Itu telah dilatih sebelumnya pada 800 juta data tidak berlabel yang diekstraksi dari BooksCorpus dan 2500 juta kata dari Wikipedia, dan kemudian, ditransfer ke masalah prediksi insiden. Kami melakukan langkah prapemrosesan tambahan untuk himpunan data kami menggunakan theBERTtokenizer. Ini menkenyalisasi dan memformat ulang urutan token dengan menambahkan CLS (token klasifikasi yang menunjukkan awal urutan) dan token SEP (ditambahkan ke akhir urutan). Panjang token deskripsi insiden kami kurang dari 512 token; oleh karena itu, kami menggunakan padding (PAD) untuk mengisi slot token yang tidak terpakai (detail lebih lanjut di bawah). Model BERT kami menghasilkan vektor penyematan 768 di masing-masing token dan memiliki 340 juta parameter yang dapat dilatih.
- ERNIE 2.0 [16]: Enhanced Representation through knowledge Integration (ERNIE) adalah kerangka kerja yang telah dilatih sebelumnya yang melakukan pelatihan urutan baru dengan tugas-tugas yang dilatih secara historis. Untuk klasifikasi teks, ERNIE 2.0 telah mengungguli BERT dengan hasil yang sangat akurat [16]. ERNIE 2.0 menangkap informasi kontekstual dengan serangkaian lapisan pengkodean teks bersama, disesuaikan dengan jaringan saraf berulang atau transformator dalam dengan lapisan perhatian diri yang ditumpuk. Pembelajaran multi-tugasnya mengkodekan informasi leksikal, sintaksis, dan semantik di seluruh tugas. Ketika tugas baru tiba, kerangka kerja ini dapat melatih representasi terdistribusi secara bertahap tanpa melupakan parameter yang dilatih sebelumnya. Dalam kerangka kerja kami, kami menggunakan 12 lapisan, 12 kepala perhatian diri, dan 768 dimensi di lapisan tersembunyi, menghasilkan 94 juta parameter yang dapat dilatih.
- RoBERTa [15]: BERT yang dioptimalkan dengan kuat menekankan data yang digunakan untuk pra-pelatihan dan jumlah kelulusan untuk pelatihan. Arsitektur RoBERTa diusulkan untuk mengatasi kelemahan model BERT asli dengan meningkatkan ukuran batch dari 256 menjadi 8K, memberikan kecepatan yang lebih baik untuk metrik kinerja [15]. Kami menggunakan transformator pra-terlatih dengan ukuran built-in (vocab) 160GB untuk RoBERTa untuk menghemat sumber daya komputasi kami. Ini mengurangi kebingungan model bahwa bertopen dengan menyediakan ketentuan untuk melatih dengan batch yang lebih besar dan urutan yang lebih panjang. Ini juga menyediakan pola penyamaran dinamis atas data pelatihan selama pra-proses data. Dalam kerangka kerja kami, kami mengimplementasikan model RoBERTa besar dengan 12 lapisan encoder, 12 kepala perhatian, dan 768 dimensi di lapisan tersembunyi, menghasilkan parameter 110Mtrainable. Akhirnya, jika suatu insiden diklasifikasikan sebagai NMIR, itu ditetapkan ke antrian prioritas normal. Jika suatu insiden diklasifikasikan sebagai MIR, itu diarahkan ke antrian prioritas tinggi dan segera ditugaskan ke tim akselerasi pemrosesan insiden untuk menyelesaikan masalah tersebut sesegera mungkin.

5. Pengaturan Eksperimental

Kami melakukan analisis empiris menggunakan state-of-art untuk kerangka kerja prediksi insiden TI yang diusulkan. Kami mengevaluasi hasil yang kami hasilkan dengan model siap pakai yang dijelaskan di Bagian 4. Kami melakukan prapemrosesan, seperti yang dijelaskan dalam Bagian 5.1, dan

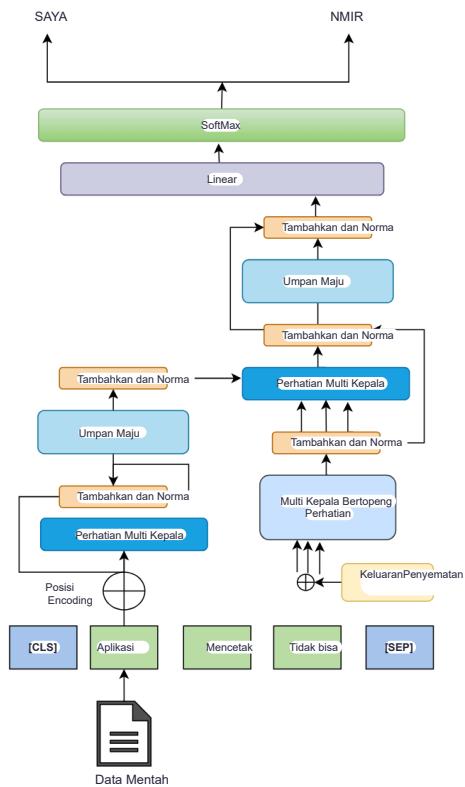
analisis tokenizer, seperti yang dijelaskan dalam Bagian 5.2. Kami melaporkan pelatihan kami di Bagian 5.3 dan validasi di Bagian 5.4.

5.1. Pra-Pemrosesan Persiapan data sangat penting dalam tugas klasifikasi teks apa pun. Untuk penelitian ini, deskripsi insiden dalam kumpulan data kami sangat tidak terstruktur tanpa pemformatan khusus (yaitu, sepenuhnya bergantung pada pandangan pengguna). Oleh karena itu, kami mengembangkan pipelinenormalisasi untuk menstandarkan teks input dan menghilangkan kebisingan yang tidak perlu. Kami mulai dengan penghapusan noiseentity untuk tag HTML, kata berhenti, tanda baca, spasi kosong, dan URL. Selanjutnya, kami menormalkan data menggunakan toolkit NLTK standar [53]. Selama normalisasi, kami melakukan tokenisasi, lemmatisasi, stemming, dan segmentasi kalimat. Penghapusan tanda baca, kami menggunakan ekspresi regulir, di mana kami menghapus semua nilai selain kata-kata alfabet (yaitu, perintah `[A-Za-z]`), lalu mengubah semua kata menjadi representasi huruf kecil. Kami menetapkan kata-kata huruf kecil ini sebagai token menggunakan WordPunctTokenizer. Fungsi ini menandai teks ke dalam urutan karakter alfabet dan non-alfabet mengikuti regexps. Ruang putih yang tidak perlu, yang merupakan produk sampingan dari konversi huruf besar ke kecil, dihapus. Terakhir, pembersihan manual urutan ini menggunakan fungsi join dan strip dilakukan. Himpunan data yang dihasilkan dibagi sebagai berikut: 70% untuk pelatihan dan 30% untuk pengujian. Untuk pelatihan, kami membagi lebih lanjut himpunan data pelatihan dengan memoles 20% dari kumpulan data pelatihan (dari 70% dari bagian aslinya) untuk validasi (yang diperlukan untuk model berbasis DL).

5.2. Tokenizer Menerima pembedaan antara insiden MIR dan NMIR adalah kunci dalam kerangka kerja kami. Oleh karena itu, untuk memungkinkan generalisasi yang baik antara teks insiden dan kelasnya, kami harus mengidentifikasi kosakata yang ada dalam kumpulan data. Untuk melakukannya, kami mengoptimalkan ukuran setiap token (yaitu, kumpulan sub-kata yang mewakili kosakata) sebagai `35 wordsmax_length`. Tokenisasi memungkinkan kami untuk memahami perbedaan antara kelas, menganalisis, dan membandingkan kosakata antar kumpulan data dan mengidentifikasi korelasi apa pun dalam setiap kelas dengan akurasi. Untuk model berbasis DL, kami menggunakan tokenizer Keras, dan untuk transformator, kami menggunakan tokenizer BERT, tokenizer RoBERTa, dan tokenizer ERNIE 2.0, yang kompatibel dengan model yang dipilih. Data yang dihasilkan setelah tokenisasi memiliki panjang yang bervariasi. Untuk mengurangi masalah ini, kami menambahkan data dengan menambahkan nol untuk mencapai panjang spesifik (35) untuk memastikan semua vektor fitur memiliki dimensi yang sama.

5.3. Pelatihan Untuk model ML, kami menggunakan TF-IDF untuk ekstraksi fitur. TF-IDF beroperasi dengan mengekstraksi kata-kata penting dalam korpus, di mana TF adalah istilah frekuensi kata-kata yang muncul dalam korpus ITSM (semua catatan insiden dalam set pelatihan), dan IDF adalah frekuensi kumpulan data yang berisi kata-kata tersebut. Semakin sedikit kata yang muncul dalam korpus, semakin tinggi nilai TF-IDF. Kami menggunakan pustaka sklearn untuk analisis kami untuk menghitung nilai TF-IDF. Setelah ekstraksi fitur melalui vektor TF-IDF, kami menggunakan pengaturan default untuk semua pengklasifikasi ML konvensional, seperti yang dijelaskan dalam sklearnlibrary [20] untuk pelatihan. Label untuk setiap vektor fitur yang tersedia dari data mentah (MIR dan NMIR) menggunakan fungsi prediksi sklearn untuk evaluasi. Untuk model DL, data teksual diubah menjadi angka dan dipetakan ke penyematian untuk mencapai kardinalitas yang lebih baik untuk setiap token. Lapisan pertama dari model DL kami adalah lapisan penyematian, yang terdiri dari tiga parameter: dimensi input, panjang input, dan dimensi output). Lapisan berikutnya, yang disebut lapisan model, berisi neuron tersembunyi yang membantu memodelkan data kompleks. Ini berisi dua parameter (jumlah neuron dan urutan pengembalian). Terakhir, kami menggunakan lapisan padat dengan parameter seperti bentuk input, ukuran batch, dan sigmoid sebagai fungsi aktivasi. Lapisan ini juga dikenal sebagai lapisan klasifikasi karena mengklasifikasikan output sebagai nilai MIR dan non-MIR. Mengkompilasi model memerlukan parameter seperti pengoptimasi dan kehilangan

fungsi. Kami menggunakan pengoptimalkan Adam dan entropi silang biner sebagai fungsi kerugian. Untuk transformator, kami menggunakan transformator yang telah dilatih sebelumnya, dari mana kami telah mengonversi laporan insiden dari kumpulan data pelatihan, validasi, dan pengujian menjadi urutan bilangan bulat dengan panjang masing-masing 35 token. Selanjutnya, kami mengubah urutan bilangan bulat menjadi tensor. Kami membuat pemuat data untuk set pelatihan dan validasi. Pemuat data ini lulus batch data pelatihan dan validasi sebagai input ke model selama fase pelatihan. Kami menerapkan konfigurasi default untuk BERT, ERNIE 2.0, dan RoBERTa. Gambar 5 menunjukkan arsitektur arsitektur transformator berkinerja terbaik kami.



Gambar 5. Representasi grafis dari arsitektur model transformator (Roberta) berkinerja terbaik kami.

5.4. ValidasiUntuk memvalidasi hasil kami, kami melakukan validasi silang, sebuah pendekatan yang menguji efisiensi kinerja model. Intinya, semakin tinggi skor metrik kinerja untuk validasi silang, semakin kecil kesalahan yang terkait dengan model. Kami memilih sklearn validasi silang k-fold bertingkat karena ini adalah bentuk halus dari teknik validasi silang lipatan tradisional. Secara acak membagi korpus sehingga rasio kelas target di setiap lipatan tetap pada tingkat yang sama seperti dalam kumpulan data keseluruhan, sehingga hasil analisisnya konsisten. Ini memastikan bahwa kelas minoritas memiliki sampel yang cukup selama pelatihan dan pengujian; Kami mengelompokkan data (Combine_All) menjadi 5 lipatan. Sebagai langkah tambahan, kami memastikan bahwa subset untuk melatih dan pengujian tidak tumpang tindih. Konfigurasi sistem ditunjukkan pada Tabel 3 dan 4.

Tabel 3. Parameter yang dipilih.

Filter	Vocab_Size	Max_Features	Embedding_dim	Batch_Size	Pengklasifikasi	Kernel	Pengaktifan	kehilangan	Pengoptimal	Learning_Rate
	(500, 1, 10, (5K, 10K, 20-an, 30-an, 50-an))	(64, 128, 256, 512)	(8, 16, 64,128, 256)	(200, 400,600, 800)	(Relu, Sigmoid, Tanh)	(RBF, (1, 2, 3, 4, Tidak Ada))	(kategorikal, ssentropy, Entropi, Entropi)	(kehilangan, KategoriSilang, KategoriSilang)	(Adam, AdamW)	(0,01, 1 × 10 ⁻³ , 1 × 10 ⁻⁵)
CNN	10,000	1000	128	256	100	4	Relay	kategorikal, ssentropy	Adam	0.01
Bi-LSTM	10,000	1000	128	256	25	Tidak	Sigmoid	Entropi	Adam	0.01
BANGAU	10,000	1000	128	256	128	3	Sigmoid	Entropi	Adam	1 × 10 ⁻³
Bert	30,522	3072	512	8	768	Tidak	Kehilangan gelurossentropi	Kehilangan entropi starg	AdamW	1 × 10 ⁻⁵
RoBERTa	50,000	3072	514	8	768	Tidak	Es	Kehilangan entropi crossentropy	AdamW	1 × 10 ⁻⁵
ERNIE	18,000	3072	256	8	768	Tidak	Relay	Kehilangan crossentropy	AdamW	1 × 10 ⁻⁵

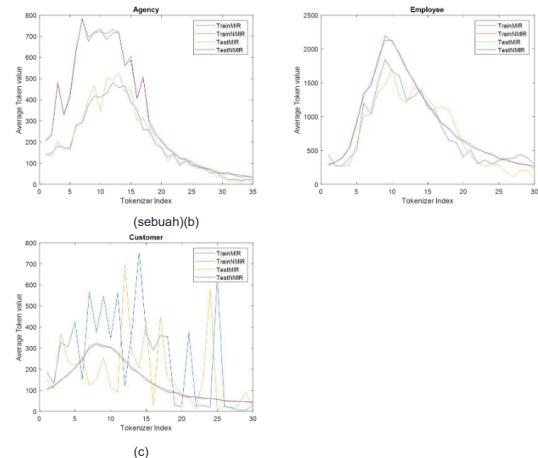
Tabel 4. Konfigurasi Sistem

Nama Konfigurasi	Versi
Aliran tensor	2.8.0
Ular sawah	3.7.7
Buku catatan Jupyter	6.2.0
GPU	Tesla P100-PCIE-12 GB
Memori GPU	HBM2 12 GB
Antarmuka Sistem	PCI dan 3.0 * 16
Konsumsi daya	250 W

6. Hasil dan AnalisisKami melakukan analisis empiris menggunakan state-of-art untuk kerangka kerja prediksi insiden TI yang diusulkan. Kami mengevaluasi hasil yang dihasilkan dari model siap pakai, seperti yang dijelaskan di Bagian 4. Kami memberikan analisis pada tokenizer di Bagian 6.1, melaporkan evaluasi kami secara kuantitatif di Bagian 6.2, validasi silang di Bagian 6.3, dan resolusi waktu rata-rata di Bagian 6.4.

6.1. Analisis TokenizerTokenisasi memungkinkan kami untuk menentukan kosakata dalam kumpulan data kami. Mempertimbangkan ukuran dan pemahaman kumpulan data asli kami tentang karakteristik antar-kumpulan data, kami membentuk analisis individu dari setiap kumpulan data, seperti yang disajikan pada Gambar 6. Untuk Agency_records(Gambar 6a), perbedaan maksimum antar kelas terlihat di 20 token pertama, saat membandingkan nilai tokenizer rata-rata. Ini menunjukkan pemisahan yang baik antar kelas berarti konsistensi selama pelatihan dan pengujian. Kami mengamati kisaran nilai token rata-rata di Agency_records lebih rendah daripada Employee_records (Gambar 6b), yang juga berkorelasi dengan varians token yang lebih tinggi untuk Employee_records. Sehingga

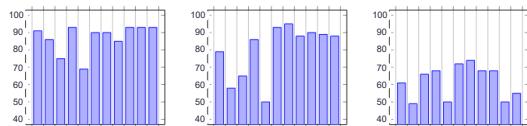
`Employee_records` memiliki kosakata yang lebih bervariasi dalam korpus mereka. Pemisahan maksimum tampaknya terjadi antara tokenizer ke-5 dan ke-15 untuk `Agency_records` dan `theEmployee_records`. Setelah diperiksa lebih dekat, kami mengamati perbedaan relatif antara classmean untuk `Employee_records` sekitar 23 token. Profilnya berbeda secara signifikan untuk `Customer_records` (Gambar 6c), yang memiliki kelas MIR yang jauh lebih kecil dengan hanya 112 sampel MIR dalam set pelatihan, dibandingkan dengan 65.000 sampel untuk NMIR. Akibatnya, nilai token rata-rata sangat bervariasi untuk kelas minoritas (dalam hal ini, MIR). Dengan demikian, nilai tokenizer rata-rata untuk kelas NMIR jauh lebih rendah, menunjukkan korpus yang lebih kecil.



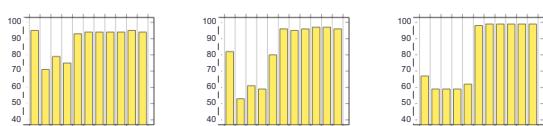
Gambar 6. Rata-rata tokenizer/kelas untuk data pelatihan dan pengujian di (a) `Agency_records`, (b) `Employee_records` dan (c) `Customer_records`.

6.2. Evaluasi Kuantitatif/Dibandingkan dengan metrik kinerja yang lebih umum (misalnya, akurasi), AUC atau ROC adalah metrik yang lebih cocok dengan kumpulan data yang tidak seimbang karena membantu mengidentifikasi over-fitting [54]. Nilai AUC atau ROC yang lebih tinggi mewakili pemisahan yang lebih baik antara kelas MIR dan NMIR. Gambar 7 menggambarkan skor kualitatif dengan ROC/AUC untuk `Agency_records`, `Employee_records`, dan `Customer_records`. Skor yang disajikan pada Gambar 7 dirata-ratakan di seluruh `Agency_records`, `Employee_records`, dan `Customer_records`. Model berbasis transformator (BERT, RoBERTa, dan ERNIE 2.0) dan XGBoost berkinerja terbaik dengan skor ROCscore 93% untuk kumpulan data `Agency_records`. Untuk `Employees_records` dan `Customers_records`, CNN berkinerja terbaik pada 95% dan 74% ROC. Secara keseluruhan, Catboost mencapai skor ROC minimum 69%, 50%, dan 50%. Hal ini diharapkan karena Catboost berjuang dalam kategori yang tidak diketahui dan perlu membangun pohon keputusan yang mendalam untuk fitur kardinalitas tinggi [55]. Seperti yang ditunjukkan di Bagian 3, kumpulan data kami sangat tidak seimbang. Oleh karena itu, kami memohon ulang himpunan data kami dan menjalankan ulang alur prediksi kami. Hasilnya digambarkan pada Gambar 8. ERNIE 2.0 mencapai ROC 95% untuk `Agency_resampled`. BERT dan ERNIE 2.0 mencapai skor tertinggi, 97% untuk `Employee_resampled` dan ERNIE 2.0 pada 98% ROC untuk Cus-

tomer_resampled. NB mencapai 71%, 53%, 59%, dan 54% ROC untuk Agency_resampled, Employee_resampled, Customer_resampled dan Combine_All, masing-masing, yang merupakan yang terendah di antara semua pengklasifikasi. Masalah berulang dengan model NB adalah bahwa dengan tidak ada kemunculan label kelas dan nilai atribut tertentu, estimasi probabilitas berbasis frekuensi akan menjadi nol. Kumpulan data besar diperlukan untuk prediksi probabilitas setiap kelas yang andal.



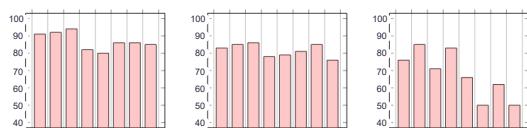
Gambar 7. Hasil ROC / AUC kuantitatif di (a) Agency_records, (b) Employee_records, dan (c) Customer_records untuk metode canggih.



Gambar 8. Hasil ROC / AUC kuantitatif di (a) Agency_resampled, (b) Employee_resampled(c) Customer_resampled, dan (d) Combine_All untuk metode canggih.

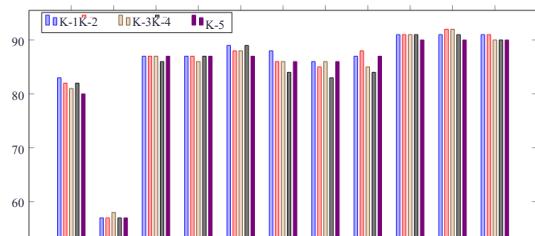
Kami juga bereksperimen dengan kurasi data sintetis (SMOTE) kami. Untuk Agency_smote and Employee_smote, XGBoost berkinerja terbaik dengan skor ROC 94% dan 86%, secara respektif. GB berkinerja terbaik untuk Customer_smote dengan nilai ROC 85%. Gambar 9

menunjukkan bahwa model DL (Bi-LSTM, GRU, CNN) berkinerja relatif buruk pada semua subset sintetis. Ini sebagian besar berkontribusi pada profil data sintetis yang dihasilkan oleh teknik SMOTE. SMOTE menciptakan batas keputusan yang lebih luas dan kurang spesifik yang meningkatkan kemampuan generalisasi pengklasifikasi ML, yang lebih cocok untuk pengkodean suatu panas. Sebaliknya, model DL menggunakan tokenizer, menyebabkan ruang fitur yang luas di NLP dan mengakibatkan model DL jatuh dengan cepat ke dalam dimensi jarang yang tinggi.



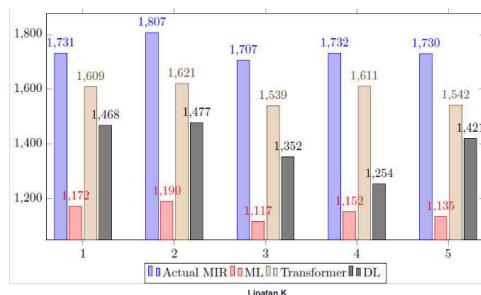
Gambar 9. Hasil ROC/AUC kualitatif untuk data sintetis (SMOTE) di seluruh (a) agensi, (b) karyawan, dan (c) pelanggan untuk metode canggih.

6.3. Validasi SilangUntuk memvalidasi temuan kami, kami menggunakan kumpulan data Combine_All. Awalnya, kami mengelompokkan kumpulan data kami menjadi 5 lipatan dengan memastikan bahwa itu tidak berisi catatan apa pun dari set pelatihan. Kami kemudian menerapkan kerangka kerja komparatif kami untuk melatih setiap algoritma secara berulang pada $k - 1$ kali lipat sambil menggunakan lipatan penahan yang tersisa sebagai set pengujian. Semua model transformator berkinerja terbaik, dengan skor ROC 90% atau lebih di setiap lipatan. Sebaliknya, NB berkinerja terburuk, dengan skor ROC kurang dari 60% (Gambar 10). Hasil k-fold kami menunjukkan bahwa temuan kami pada Gambar 8 konsisten dengan nilai ROC komparatif dengan validasi silang.



Gambar 10. Hasil ROC/AUC di seluruh validasi silang k-fold.

Pada Gambar 11, kami memberikan jumlah ansambel rata-rata (rata-rata label yang diprediksi untuk semua model) dari ketiga kelas yang sesuai dengan ML konvensional, berbasis DL, dan transformator. Selain itu, kami juga melaporkan matriks kebingungan untuk memverifikasi kelas benar-positif (TP), benar-negatif (TN), negatif palsu (FN), dan positif palsu (FP) untuk setiap lipatan. Misalkan, jumlah ansambel rata-rata adalah nilai MIR aktual versus nilai yang diprediksi untuk setiap lipatan (dari 1 hingga 5), yaitu benar-positif (TP). Matriks kebingungan untuk ML, DL, dan Transformers tercantum dalam Tabel 5 dan 6. Misalkan, dalam k-fold, ada 1731 total MIR, sedangkan rata-rata ML (yaitu, rata-rata label yang diprediksi dari semua model ML) memprediksi 1172 TP MIR, serta 1468 TP MIR untuk berbasis DL dan 1609 TPMIR untuk transformator. Demikian pula, untuk TN, NB memprediksi 64.722 TN untuk ML. Untuk DL, Bi-LSTM memprediksi 64.600 TN, dan untuk transformator, BERT melaporkan nilai 64.605 TN tertinggi. Untuk ML konvensional, nilai prediksi rata-rata untuk semua lipatan adalah 1153, dengan SVM memiliki pangsa tertinggi dengan 1499, yang lebih rendah dari label yang diprediksi 1731. Model DL menunjukkan hasil ansambel yang lebih baik dengan 1394 label yang diprediksi untuk semua lipatan, dengan CNN memiliki jumlah tertinggi dari 1673 label yang diprediksi. Transformers mengungguli teknik ML konvensional dengan 1584 label ansambel yang diprediksi rata-rata. RoBERTa tampil sebagai model transformator yang paling baik dengan 1693. Secara keseluruhan, model transformator berkinerja terbaik untuk kumpulan data *theCombine_All* (Lihat Gambar 8).



Gambar 11. Jumlah MIR di seluruh algoritma aktual, ML (Bayes naif, gradient boost, XGBoost, CatBoost, SVM), transformator (BERT, RoBERTa, ERNIE 2.0), dan algoritma DL (GRU, Bi-LSTM, CNN).

Tabel 5. Validasi silang k-fold dengan data lengkap (Combine_All seperti yang dijelaskan di Bagian 3) disegmentasikan menjadi lima lipatan. Kami membagi data menjadi pemisahan 80:20 untuk pelatihan dan pengujian dengan memastikan set pengujian tidak berisi catatan apa pun dari set pelatihan. Kami menerapkan model ML Bayes naif, gradient boosting, XGBoost, CatBoost, dan SVM.

lipatan k	Bayes yang naif			Peningkatan Gradien			Peningkatan XG		
	HCMC	TNFN	FPTP	TNFN	FPTP	TNFN	FP		
1	276	64,735	1455	3	1154	64,638	577100	1322	64,623
2	271	64,659	1536	3	1166	64,584	641	78	1363
3	275	64,758	1432	3	1068	64,682	639	79	1272
4	267	64,732	1465	4	1112	64,644	620	92	1284
5	261	64,729	1469	8	1075	64,461	655	96	1286
Berarti	270	64,722	1471	4	1115	64,601	626	89	1305
CatBoosting									
lipatan k	SVM			TNFN			FP		
	HCMC	TNFN	FPTP	TNFN					
1	1302	64,621	429117	1370	64,635	361103			
2	1373	64,565	434	97	1407	64,570	400	92	
3	1256	64,646	451115		1301	64,666	406	95	
4	1303	64,621	429115		1355	64,624	377112		
5	1287	64,611	443126		1304	64,634	426103		
Berarti	1304	64,612	437114		1347	64,625	394101		

Tabel 6. Validasi silang k-fold dengan semua data (Combine_All seperti yang dijelaskan di Bagian 3) tersegmentasi menjadi lima lipatan. Kami membagi data menjadi pemisahan 80:20 untuk pelatihan dan pengujian dengan memastikan set pengujian tidak berisi catatan apa pun dari set pelatihan. Kami menerapkan transformator (BERT, RoBERTa dan ERNIE 2.0) dan algoritma DL (GRU, Bi-LSTM dan CNN).

lipatan k	BANGAU			Bi-LSTM			CNN		
	HCMC	TNFN	FPTP	TNFN	FPTP	TNFN	FP		
1	1334	64,513	397225	1251	64,588	480150	1304	64,553	427185
2	1328	64,537	497125	1301	64,514	506148	1403	64,392	404270
3	1260	64,599	447162	1238	64,652	469109	1205	64,633	502128
4	1205	64,643	527	93	1169	64,666	563	70	1189
5	1280	64,575	450162		1278	64,584	452153		1289
Berarti	1281	64,573	460153		1247	64,600	494126		1278
Bert									
lipatan k	Roberta			ERNIE					
	HCMC	TN	FN	FP	HCMC	TN	FN	FP	
1	1437	64,590	294148		1429	64,616	302122		1430
2	1438	64,597	293141		1539	64,508	268154		1486
3	1423	64,623	284138		1525	64,495	282167		1395
4	1454	64,613	278123		1428	64,592	304144		1401
5	1411	64,602	319135		1390	64,580	340157		1394
Berarti	1432	64,605	239137		1462	64,558	299148		1421

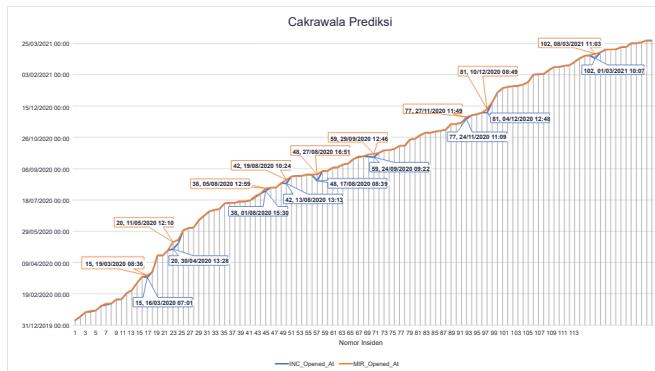
Mirip dengan jumlah ansambel rata-rata, RoBERTa dengan benar memprediksi 1462 TP dan 64.558 TN, sedangkan NB berkinerja terburuk dengan mengidentifikasi hanya 270 TP dan 64.722 TN.

6.4. Resolusi Waktu Rata-rata Untuk sementara itu, kami menggunakan catatan pemadaman MIR karena berisi catatan insiden yang dieksklusif. Untuk menghitung sementara untuk menyelesaikan, dua kolom DateTime digunakan: Incident_Opened_at dan MIR_Opened_at. Kami mendefinisikan sementara waktu untuk diselesaikan

seperti waktu yang dicatat, dimulai ketika insiden telah dilaporkan dan kemudian diselesaikan oleh tim pemadam. Sementara waktu untuk diselesaikan berkisar dari menit hingga jam. Evaluasi sementara untuk menyelesaikan dilakukan dengan menggunakan cakrawala prediksi, sebagai berikut:

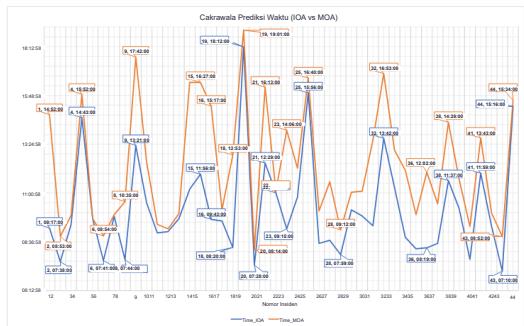
$$\text{Prediction_Horizon} = \sum T2n - \sum T1n,$$

di mana $\sum T2n$ menunjukkan MIR dibuka pada (T_2, \dots, T_{2n}) dan $\sum T1n$ menunjukkan insiden yang dibuka pada insiden (IOA) (T_1, \dots, T_{1n}). Cakrawala prediksi kami (sementara untuk menyelesaikan) untuk catatan MIR dari Januari 2020 hingga Maret 2021 digambarkan pada Gambar 12. Garis biru menunjukkan catatan IOA sedangkan oranye menunjukkan catatan MOA. Titik IOA (berwarna oranye) menunjukkan DateTime record tertinggi tempat log insiden dibuat. MOA menunjukkan titik di mana kami menyatakan insiden sebagai MIR. Misalnya, tiket #20 telah dilaporkan ke IOA pada 30 April 2020 dan diberi label sebagai MIR pada 11 Mei 2020 oleh tim pemadam, sehingga butuh waktu sekitar 11 hari untuk menyelesaikan insiden ini. Tiket lain, #48, telah dilaporkan sebagai IOA pada 17 Agustus 2020 dan diberi label sebagai MIR pada 27 Agustus 2020, memakan waktu sekitar sepuluh hari. Pada Gambar 12, kami mencatat bahwa garis oranye tumpang tindih dengan garis biru, menunjukkan penutupan insiden pada contoh yang sama. Kami memprediksi waktu cakrawala prediksi rata-rata MIR sebagai 4 hari, 3 jam, dan 26 menit, yang dapat memberikan keuntungan signifikan dalam mengatasi insiden dan dapat mencegah insiden besar atau pemadaman.



Gambar 12. Perbandingan cakrawala prediksi (MIR dibuka pada vs. insiden dibuka pada. Garis biru menunjukkan catatan IOA sedangkan oranye menunjukkan catatan MOA).

Untuk menyeleksi evaluasi kami lebih lanjut, kami menyempurnakan cakrawala prediksi dengan hanya memiliki catatan insiden yang dieksklusi dengan durasi waktu 6 jam atau kurang. Alasannya adalah untuk meneliti hasil dalam konteks jam (menghapus outlier yang membutuhkan waktu lebih dari berhari-hari untuk diselesaikan) untuk mengukur lebih baik sementara yang harus diselesaikan untuk sebagian besar insiden, jika diprediksi dengan benar. Insiden yang tersisa berjumlah 44 (lihat Gambar 13). Garis oranye menunjukkan nilai MOA, menunjukkan waktu insiden yang meningkat, sedangkan garis biru menunjukkan catatan IOA. Insiden # 1 membutuhkan waktu maksimal 5 jam dan 35 menit untuk meringkat; Sebaliknya, insiden #7 membutuhkan waktu 5 menit untuk menandai insiden signifikan. Hasil yang dianalisis berdasarkan pencarian yang disempurnakan menunjukkan kepada kami waktu rata-rata untuk menyelesaikan insiden dalam 1 hari adalah 2 jam dan 30 menit.



Gambar 13. Perbandingan cakrawala prediksi (MIR dibuka pada vs. insiden dibuka pada).

7. DiskusiStudi ini secara komprehensif mengevaluasi pengklasifikasi AI menggunakan dataset eksklusif dunia nyata. Seperti yang diamati, kami menghadapi distribusi yang sangat miring dengan hampir satu juta catatan untuk NMIR dan hanya 19.000 catatan untuk MIR (yang hanya 2% dari keseluruhan kumpulan data). Kumpulan data sangat tidak terstruktur, yang semakin memperumit tugas praproses data. Himpunan data kami terdiri dari laporan/tiket insiden mentah. Oleh karena itu, sebagian besar catatan berisi kesalahan ketul, kesalahan ejaan, kesalahan tata bahasa, dan kata-kata gaul. Kami juga memperhatikan bahwa 6% dari log insiden berisi catatan duplikat, yang tidak memberikan informasi tambahan dan menghabiskan sumber daya memori tambahan selama pemrosesan. Kami menyarankan untuk menghapus catatan ini karena membutuhkan masalah jarang dan meningkatkan ukuran vektor. Dalam pengaturan dunia nyata, kumpulan data insiden TI memerlukan pembersihan ekstensif menggunakan alur praprosesos yang dipantau secara ketat untuk mengekstrak fitur linguistik penting, yang kami serangkan sebagai bagian dari kerangka prediksi yang dilengkapi. Selain itu, kami menunjukkan bahwa augmentasi data memberikan hasil yang relatif lebih baik daripada data sintetis. Kami mengembangkan pendekatan khusus untuk augmentasi data, di mana data aktual dari berbagai satuan (mewakili label minoritas) ditambahkan untuk memperbaiki kumpulan data milik pemangku kepentingan industri. Pengayaan ini lebih bermanfaat daripada menghasilkan catatan sintetis untuk menghindari duplikasi yang tidak menambahkan informasi tambahan ke kosakata. Analisis kami menunjukkan bahwa transformator berkinerja sedikit lebih baik daripada model berbasis DL dan secara signifikan lebih baik daripada model ML konvensional. Misalnya, mengingat himpunan data *Combine_All*, ERNIE 2.0 mengungguli semua model ML konvensional, berkinerja lebih baik daripada NB, GB, SVM, XGBoost, dan CatBoost, masing-masing sebesar 39%, 10%, 3%, 19%, dan 4%. Dibandingkan dengan model berbasis DL, ERNIE 2.0 menghasilkan hasil yang sedikit lebih baik, dengan peningkatan 1% dibandingkan CNN dan peningkatan 2% dibandingkan Bi-LSTM. Umumnya, model berbasis DL berkinerja lebih baik karena unit memori penarikan mereka yang mengekstrak fitur abstrak tingkat tinggi; namun, mereka berkinerja sedikit lebih buruk, dibandingkan dengan Transformers karena kosakata mereka yang terbatas [10]. Temuan kami tidak sesuai dengan pengamatan ini. Kami menemukan bahwa akurasi bukanlah metrik kinerja yang tepat karena hasil awal kami untuk data yang tidak seimbang lebih condong ke arah kelas mayoritas. Kami menggunakan matriks kebingungan dan area di bawah kurva ROC untuk mengukur prediksi yang benar dan salah dengan lebih baik. Berdasarkan temuan kami, metrik ROC menunjukkan ketepatan dan kelengkapan pengklasifikasi jauh lebih baik daripada akurasi. Misalnya, *Customer_records* memiliki catatan kelas minoritas (MIR) terdiri sebesar 0,05%, yang sangat tidak seimbang. Skor ROC-nya bervariasi dari 50% hingga 74%, sedangkan skor akurasiannya adalah 99%, yang menyatakan dan

sangat miring oleh kelas mayoritas (NMIR). Kami menghitung matriks kebingungan dengan k-foldvalidation dan rata-rata untuk setiap lipatan. Hasil kami menunjukkan bahwa RoBERTa memiliki skor rata-rata tertinggi dengan nilai 1462 TP dan 64.539 TN. Kami menghitung cakrawala prediksi untuk mengidentifikasi waktu rata-rata pemulihan sistem ITSM dari kebutuhan yang meningkat. Kami menargetkan insiden dibuka di (IOA) dan MIR openedit (MOA) untuk menghitung cakrawala prediksi. Kami mengevaluasi waktu eskalasi insiden rata-rata, yaitu 2 jam 30 menit waktu yang dihemat karena menerapkan kerangka insiden TI yang kami usulkan. Itu tidak hanya menghemat sumber daya tetapi juga membantu mencegah situasi kebutuhan. Mempertimbangkan kompleksitas prediksi, kemungkinan penerapan RL untuk meningkatkan transformator ini adalah kemungkinan yang signifikan. Pada saat itu, kami dapat mengidentifikasi studi apa pun yang terkait dengan implementasi RL untuk prediksi insiden TI. Tingkat konvergensi model RL untuk klasifikasi diperkirakan 40% lebih banyak, dibandingkan dengan model canggih lainnya [56], yang dapat menghambat adopsi untuk kumpulan data industri skalas besar. Selain itu, pekerjaan di masa depan dapat melibatkan evaluasi NLPGym [57], toolkit yang menjembatani NLP dan RL. NLPGym memberikan kebijakan untuk hadiah dan penalti terhadap tindakan, dan jaringan deep-Q (DQN) dapat digunakan untuk melatih model. Kami optimis bahwa kami dapat meningkatkan skor akurasi secara signifikan melalui RL.

8. KesimpulanMakalah ini memanfaatkan pembelajaran mesin canggih (NB, SVM, GB, XGBoost, CatBoost), pembelajaran mendalam (GRU, CNN, Bi-LSTM), dan arsitektur transformator (BERT, Roberta, dan ERNIE) untuk mengklasifikasi insiden yang disediakan sebagai mayor (MIR) dan non-besar (NMIR) untuk mengatasi tantangan prediksi insiden TI. Makalah ini adalah upaya pertama untuk menggunakan pembelajaran mendalam dan transformator untuk memecahkan masalah manajemen layanan TI, sepengetahuan kami. Kami bereksperimen dengan tiga sumber insiden yang berbeda: agensi, pelanggan, dan karyawan. Arsitektur transformator (BERT, Roberta, dan ERNIE) dan XGBoost mengungguli semua metode lainnya, dengan ROC 93% untuk catatan agensi, dan dengan CNN, masing-masing 95% dan 74% untuk karyawan dan pelanggan. Untuk mengatasi masalah ketidakseimbangan data dalam pekerjaan kami, kami melakukan pengambilan sampel ulang dan mengkursi data sintetis (SMOTE). Dengan pengambilan sampel, kami mencapai 95% dengan ERNIE untuk agensi, 95% dengan BERT dan ERNIE untuk karyawan, 99% dengan model transformator dan pembelajaran mendalam untuk pelanggan, dan 93% dengan BERT dan ERNIE untuk semua gabungan (karyawan, agensi, dan pelanggan). Untuk data sintetis, XGBoost memiliki 94% untuk agensi, 86% untuk karyawan, dan dengan GB, 85% untuk pelanggan. Dengan pengambilan sampel ulang dan kurasi data sintetis, kami melihat peningkatan relatif yang signifikan, sebesar 2,10% untuk agensi dan karyawan dan 33% untuk pelanggan. Kami juga mengidentifikasi keterbatasan model pembelajaran mendalam untuk SMOTE, yang kami rencanakan untuk ditangani di masa mendatang. Selain itu, kerangka kerja yang kami usulkan menghitung cakrawala prediksi rata-rata, yaitu 2 jam 30 menit dan memanfaatkan ini untuk meminimalkan sementara waktu untuk menyelesaikan insiden. Di masa mendatang, kami akan menerapkan kerangka kerja kami untuk mengelola dan mengurangi tingkat kegagalan perubahan yang terkait dengan waktu rata-rata penyelesaian insiden (MTTR). Untuk pekerjaan di masa depan, kami berencana untuk melakukan analisis lengkap ITSM dengan teknik validasi silang yang canggih karena literatur tidak memiliki domain ini.

Kontribusi Penulis: S.A.: Menulis—meninjau dan mengedit, Menulis—draft asli, Usulan Bingkai, MS: Metodologi, Penulisan—draft asli, Penulisan tinjauan dan penyuntingan, Pengawasan. B.D.: tinjauan dan administrasi proyek. E.R.: Proofreading, Penulisan—tinjauan dan penyuntingan, Pengawasan.K.H.: tinjauan dan administrasi proyek. M.B.: Mengoreksi, Menulis—meninjau dan mengedit, Pengawasan dan akuisisi Pendanaan. D.C.: Metodologi, Penulisan—draft asli, Penulisan—peninjauan, penyuntingan, Pengawasan, Administrasi proyek, dan akuisisi Pendanaan. Semua penulis telah membaca dan menyertai versi naskah yang diterbitkan. Pendanaan: Pekerjaan ini didukung oleh Penelitian dan Inovasi Inggris Turing AI Fellowship 2021–2025 yang didanai oleh Dewan Riset Teknik dan Ilmu Fisika (nomor hibah EP/V/025724/1). Pernyataan Dewan Peninjau Institusional: Tidak berlaku. Pernyataan Persetujuan Informasi: Tidak berlaku.

Pernyataan Ketersediaan Data: Data akan tersedia berdasarkan permintaan. Ucapan terima kasih: Kami berterima kasih atas akses ke sumber daya komputasi kinerja tinggi tingkat 2 yang disediakan oleh fasilitas Komputasi Kinerja Tinggi Irlandia Utara (NI-HPC), yang didanai oleh Inggris. Dewan Riset Teknik dan Ilmu Fisika (EPSRC), Hibah No. EP/T022175/ dan EP/W03204X/1. Damien Coyle didukung oleh UKRI Turing AI Fellowship 2021–2025 yang didanai oleh EPSRC (nomor hibah EP/V025724/1). Salman Ahmed didukung oleh George MoorePh.D. beasiswa. Konflik Kepentingan: Penulis menyatakan tidak ada konflik kepentingan.

Referensi

- Cortina, S.; Barafot, B.; Picard, M.; Renault, A. Menggunakan Model Penilaian Proses untuk Mempersiapkan Sertifikasi ISO/IEC 20000-1: ISO/IEC 15504-8 TIPA untuk? Dalam Prosiding Peningkatan Proses Sistem, Perangkat Lunak, dan Layanan—Konferensi Eropa ke-23, EuroSPI 2016, Graz, Austria, 14–16 September 2016; Komunikasi dalam Ilmu Komputer dan Informatika; Kreiner, C., O'Connor, RV, Poth, A., Messnarz, R., Eds.; Springer: Berlin/Heidelberg, Jerman, 2016; Volume 633, hlm. 83–93. [Referensi Silang]
- Lou, J.G.; Lin, Q.; Ding, R.; Fu, Q.; Zheng, D.; Xie, T. Analitik perangkat lunak untuk manajemen insiden layanan online: Laporan pengalaman. Dalam Prosiding Konferensi Internasional IEEE/ACM ke-28 2013 tentang Rekayasa Perangkat Lunak Otomatis (ASE), Silicon Valley, CA, AS, 11–15 November 2013; hlm. 475–485.
- Bartolini, C.; Sallé, M.; Trastour, D. Manajemen layanan TI yang didorong oleh tujuan bisnis Aplikasi untuk manajemen insiden. Dalam Prosiding Simposium Operasi dan Manajemen Jaringan IEEE/IFIP 2006 NOMS 2006, Vancouver, BC, Kanada, 3–7 April 2006; hlm. 45–55.
- Takeshita, K.; Yokota, M.; Nishimatsu, K. Sistem deteksi kegagalan jaringan dini dengan menganalisis data Twitter. Dalam Prosiding Simposium Internasional IFIP/IEEE 2015 tentang Manajemen Jaringan Terpadu (IM), Ottawa, ON, Kanada, 11–15 Mei 2015; hlm. 279–286.
- Zhou, W.; Tang, L.; Zeng, C.; Li, T.; Schwartz, L.; Grabarnik, G.Y. Rekomendasi resolusi untuk tiket acara dalam manajemen layanan. IEEE Trans. Netw. Melayani Manag. 2016, 13, 954–967.
- Zhou, W.; Li, T.; Schwartz, L.; Grabarnik, G.Y. Merekomenadasi resolusi tiket menggunakan adaptasi fitur. Dalam Prosiding Konferensi Internasional 2015/11 tentang Manajemen Jaringan dan Layanan (CNMS), Barcelona, Spanyol, 9–13 November 2015; hlm. 15–21.
- Dejic, Ž.; Randić, M.; Krčelić, G. Deteksi dini pemadaman elemen jaringan berdasarkan panggilan masalah pelanggan. Keputusan. Support Syst. 2015, 73, 57–73.
- Glenn, J.S.; Rose, K.L. Menetapkan Tata Kelola untuk Manajemen Projek dan Layanan. Dalam Prosiding ACM 2019 Konferensi Tahunan SIGUCCS, SIGUCCS 2019, New Orleans, LA, AS, 3–6 November 2019; Haring-Smith, B., McIntosh, KM, Lineberry, B., Eds.; ACM: New York, NY, AS, 2019; hlm. 145–147. [Referensi Silang]
- Pembakar Batu, G.; Goquen, A.; Feringa, A. Panduan Manajemen Risiko untuk Sistem Teknologi Informasi; SP 800-30; Institut Standar dan Teknologi Nasional: Gaithersburg, MD, AS, 2022.
- Beresnev, A.; Gusarova, N. Perbandingan Algoritma Klasifikasi Cerdas untuk Sistem Pembelajaran Tempat Kerja di Perusahaan Berorientasi Layanan Berteknologi Tinggi. Dalam Prosiding Konferensi Internasional tentang Transformasi Digital dan Masyarakat Global, St. Petersburg, Rusia, 17–19 Juni 2020; Springer: Berlin/Heidelberg, Jerman, 2020; hlm. 363–372.
- Tolciu, D.T.; Sacarea, C.; Matei, C. Analisa Pola dan Kesamaan dalam Tiket Layanan menggunakan Pemrosesan Bahasa Alami J. Komunitas Lembut. Syst. 2021, 17, 29–35.
- Nikulin, V.; Shibaikin, S.; Vishnyakov, A. Penerapan metode pembelajaran mesin untuk klasifikasi dan perutean otomatis di. J. Phys. Conf. Ser. 2021, 2081, 012041.
- Kodegodu, K.R.; Annam, J.R.; Vippalla, A.; Krishna, B.V.N.V.S.; Kumar, N.; Viswanathan, R.; Gaddala, L.K.; Chandrapalli, S.K.A. Jaringan Saraf Konektifional Dalam Baru untuk Diagnosis Penyakit Kulit. Sifat. Sinyal 2022, 39, 1873–1877. [Referensi Silang]
- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pra-pelatihan transformator dua arah dalam untuk pemahaman bahasa.arXiv 2018, arXiv:1810.04805.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: Pendekatan Prapelatihan BERT yang Dioptimalkan dengan Kuat. arXiv 2019, arXiv:1907.11692.
- Mathai, Y.; Wang, S.; Li, Y.; Feng, S.; Tian, H.; Wu, H.; Wang, H. ERNIE 2.0: Kerangka kerja pra-pelatihan berkelanjutan untuk pemahaman bahasa. Dalam Prosiding Konferensi AAAI tentang Kecerdasan Buatan, New York, NY, AS, 7–12 Februari 2020; Volume 34, hlm. 8968–8975.
- Ali Zaidi, S.S.; Fraz, M.M.; Shahzad, M.; Khan, S. Kerangka kerja umum multipendekatan untuk saran solusi otomatis tiket dukungan. Int. J. Intel. Syst. 2022, 37, 3654–3681. [Referensi Silang]
- Gouryash, S.; Kataria, S.; Sivagiraradoss, J. Prediksi Pelanggaran Perjanjian Tingkat Layanan di ServiceNow. Dalam Prosiding Konferensi Internasional Ketiga 2021 tentang Penelitian Inovatif dalam Aplikasi Komputasi (ICIRCA), Coimbatore, India, 2–4 September 2021; hlm. 689–698.
- Agarwal, S.; Aggarwal, V.; Akula, A.R.; Dasgupta, G.B.; Sridhara, G. Ekstraksi dan analisis masalah otomatis dari teks tidak terstruktur dalam tiket IT. IJL. Res. Dev. 2017, 61, 4–41.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; dtkr. Scikit-learn: Pembelajaran Mesin dalam Python. J. Mach. Belajar. Res. 2011, 12, 2825–2830.
- Zhao, H.; Lai, Z.; Leung, H.; Zhang, X. Pengantar yang lembut untuk pembelajaran fitur. dalam Pembelajaran dan Pemahaman Fitur; Springer: Berlin/Heidelberg, Jerman, 2020; hlm. 1–12.
- Al Qadi, L.; El Rifai, H.; Obaid, S.; Elnagar, A. Pendekatan pembelajaran dangkal yang dapat diskalakan untuk mendaindri artikel berita bahasa Arab. Yordania, J. Comput. Inf. Technol. 2020, 6, 263–280.
- Ye, Y.; Ma, F.; Liu, Y.; Chiu, M.; Huang, J.Z. iSurfer: A Focused Web Crawler Based on Incremental Learning from Positive Samples. In Proceedings of the Advanced Web Technologies and Applications, 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, 14–17 April 2004; Catatan Kuliah dalam Ilmu Komputer; Yu, J.X., Lin, X., Lu, H., Zhang, Y., Eds.; Springer: Berlin/Heidelberg, Jerman, 2004; Volume 3007, hlm. 122–134. [Referensi Silang]
- Reyina, A.; Buza, K.; Meister, VG Klasifikasi Tiket IT: Semakin Sederhana, Semakin Baik. Akses IEEE 2020, 8, 193380–193395.

25. Revina, A.; Buza, K.; Meister, V.G. Merancang Alur Klasifikasi Teks yang Dapat Dijelaskan: Wawasan dari Kompleksitas Tiket TI Studi Kasus Prediksi Menafsirkan. *Artif. Intell. Sebuah perspektif*. Granul. Komputasi, 2021, 937, 293.
26. Zuev, D.; Zuev, A. Pembelajaran mesin dalam manajemen layanan TI. *Komputasi Prosedur. Sci.* 2018, 145, 675–679.
27. Costa, J.; Pereira, R.; Ribeiro, R. Otomatasi ITSM-Menggunakan pembelajaran mesin untuk memprediksi kategori resolusi insiden. Dalam Prosiding Konferensi Asosiasi Manajemen Informasi Bisnis Internasional ke-33: Keunggulan dan Inovasi Pendidikan Manajemen melalui Visi 2020. IBIMA 2019, Granada, Spanyol, 10–11 April 2019; hlm. 5819–5830.
28. Wang, Q.; Zeng, C.; Iyengar, S.; Li, T.; Shwartz, L.; Grabarnik, G.Y. AISTAR: Sistem cerdas untuk rekomendasi otomatisasi tiket TI online. Dalam Prosiding Konferensi Internasional IEEE 2018 tentang Big Data (Big Data), Seattle, WA, AS, 10–13 Desember 2018; hlm. 1875–1884.
29. Al-Hawari, F.; Barham, H. Sistem meja bantuan berbasis pembelajaran mesin untuk manajemen layanan TI. *J. King Saud Univ. Comput. Inf. Sci.* 2021, 33, 702–718.
30. Paramesh, S.; Shreedhara, K. Klasifikasi Insiden Meja Bantuan TI menggunakan Ansambel Pengklasifikasi. *ICTACT J. Komputasi Lunak* 2019, 9, 1980–1987.
31. Gupta, R.; Prasad, K.H.; Mohania, M. Mengolokmatikan proses manajemen insiden ITSM. Dalam Prosiding Konferensi Internasional 2008 tentang Komputasi Oltonomi, Chicago, IL, AS, 2–6 Juni 2008; hlm. 141–150.
32. Altintas, M.; Tantur, AC. Klasifikasi tiket berbasis pembelajaran mesin dalam sistem pelacakan masalah. Dalam Prosiding Konferensi Internasional tentang Kecerdasan Buatan dan Ilmu Komputer (AICS 2014). Bandung, Indonesia, 15–16 September 2014; hml. 195–207.
33. Silva, S.; Pereira, R.; Ribeiro, R. Pembelajaran mesin dalam otomatisasi kategorisasi insiden. Dalam Prosiding Konferensi Iberia ke-13 2018 tentang Sisteme dan Teknologi Informasi (CISTI). Cáceres, Spanyol, 13–16 Juni 2018; hml. 1–6.
34. Paramesh, S.; Shreedhara, K. Sistem meja layanan TI otomatis menggunakan teknik pembelajaran mesin. Dalam Analisis Data dan Pembelajaran. Springer: Berlin/Heidelberg, Jerman, 2019; hml. 331–346.
35. Roy, S.; Muni, DP.; Tack Yan, JY.; Budhiraja, N.; Ceiler, A. Pengelompokan dan pelabelan tiket pemeliharaan TI. Dalam Prosiding Konferensi Internasional tentang Komputasi Berorientasi Layanan, Banff, AB, Kanada, 10–13 Oktober 2016; hml. 829–845.
36. Schad, J.; Sambasivan, R.; Woodward, C. Memprediksi penetapan ulang tiket meja bantuan dengan jaringan konvolusional grafik.Mach., Pelajar. *Aplikasi* 2022, 7, 100237.
37. Aglibar, K.D.; Alegre, G.C. Del Mundo, G.; Duro, K.F.; Rodelas, N. Sistem Tiket: Penelitian Deskriptif tentang Penggunaan Sistem Tiket untuk Manajemen Proyek dan Pelafakatan Masalah di Perusahaan TI. *arXiv* 2022, arXiv:2202.06213.
38. YAYAH, F.C.; Ghauth, K.I.; TING, CY. Pendekatan klasifikasi pembelajaran mesin otomatis pada kumpulan data tiket masalah telekomunikasi. *J. Eng. Sci. Technol.* 2021, 16, 4263–4282.
39. Bajpai, H. Membangun Sistem Cerdas Berbasis ML untuk Menganalisis LSI Produksi (Insiden Situs Langsung). *Int. J. Eng. Adv. Technol. (IJEAT)* 2021, 10, 41–46. [Referensi Silang]
40. Shorten, C.; Khoshgoftaar, T.M.; Furti, B. Augmentasi Data Teks untuk Pembelajaran Mendalam. *J. Data Besar* 2021, 8, 101. [Referensi Silang] [PubMed]
41. Chawla, N.V.; Bowyer, K.W.; Aula, L.O.; Kegelmeyer, W.P. SMOTE: Teknik over-sampling minoritas sintetis. *J. Artif. Intell. Res.* 2002, 16, 321–357.
42. Webb, G.I.; Keogh, E.; Milkulainen, R.; Naïf Bayes. *Encycl. Mach. Belajar.* 2010, 15, 713–714.
43. Mayr, A.; Pengikat, H.; Gefeller, O.; Schmid, M. Evolusi algoritma pengukur. *Metode Inf. Med.* 2014, 53, 419–427.
44. Chen, T.; Dia, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K.; Mitchell, R.; Kano, I.; Zhou, T. Xgboost: Peningkatan Gradien Eksplorasi. Paket R; Tim Kode R: Wina, Austria, 2015; hml. 1–4.
45. Corrao, R.; Probst, P.; Boulesteix, A.L. Huban akas versus regresi logistik: Eksperimen tolok ukur skala besar. *BMC Bioinform.* 2018, 19, 270.
46. Dorogush, A.V.; Ershov, V.; Gulin, A. CatBoost: Peningkatan gradien dengan dukungan fitur kategoris. *arXiv* 2018, arXiv:1810.11363.
47. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: Pohon keputusan peningkatan gradien yang sangat efisien. *Adv. Proses Inf. Neural. Syst.* 2017, 30, 1–9.
48. Suthaharan, S. Mendukung mesin vektor dalam model pembelajaran mesin dan algoritme untuk klasifikasi data besar; Springer-Berlin/Heidelberg, Jerman, 2016; hml. 207–235.
49. Schuster, M.; Paliwal, KK. Jaringan saraf berulang dua arah. *IEEE Trans. Proses Sinyal.* 1997, 45, 2673–2681.
50. O’Shea, K.; Nash, R. Pengantar Jaringan Saraf Konvolusional. *arXiv* 2015, arXiv: 1511.08458.
51. Collobert, R.; Weston, J. Arsitektur terpadu untuk penerapan bahasa alami: Jaringan saraf dalam dengan pembelajaran multitugas. Dalam Prosiding Konferensi Internasional ke-25 tentang Pembelajaran Mesin, Helsinki, Finlandia, 5–9 Juli 2008; hml. 160–167.
52. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. Tentang sifat-sifat terjemahan mesin saraf: Pendekatan encoder-decoder. *arXiv* 2014, arXiv:1409.1259.
53. Burung, S.; Loper, E. NLTK. Toolkit Bahasa Alami. dalam prosiding sesi poster dan demonstrasi interaktif ACL; Asosiasi Linguistik Komputasi: Barcelona, Spanyol, 2004; hml. 214–217.
54. Ling, C.X.; Huang, J.; Zhang, H. AUC: Ukuran yang Lebih Baik daripada Akurasi dalam Membandingkan Algoritma Pembelajaran. Dalam Kemajuan dalam Kecerdasan Buatan; Xiang, Y., Chaib-draa, B., Eds.; Springer: Berlin/Heidelberg, Jerman, 2003; hml. 329–341.
55. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Peningkatan yang tidak bias dengan fitur kategoris. *Adv. Proses Inf. Neural. Syst.* 2018, 31, 1–11.

56. Lagoudakis, M.G.; Parr, R. Pembelajaran pengukuran sebagai klasifikasi: Memanfaatkan pengklasifikasi modern. Dalam Prosiding Konferensi Internasional ke-20 tentang Pembelajaran Mesin (ICML-03), Washington, DC, AS, 21–24 Agustus 2003; hlm. 424–431.
57. Ramamurthy, R.; Sifa, R.; Bauchage, C. NLPgym—Toolkit untuk mengevaluasi agen RL pada Tugas Penrosesan Bahasa Alami. 2020. Tersedia secara online: <http://xxx.lanl.gov/abs/2011.08272> (diakses pada 30 Agustus 2022).

Penafian/Catatan Penerbit: Pernyataan, pendapat, dan data yang terkandung dalam semua publikasi semata-mata milik penulis dan kontributor individu dan bukan MDPI dan/atau editor. MDPI dan/atau editor menyangkal tanggung jawab atas cedera apa pun pada orang atau properti yang diakibatkan oleh ide, metode, instruksi atau produk apa pun yang dirujuk dalam konten.