YAYASAN PRIMA AGUS TEKNIK

Kecerdasan Buatan Gabungan pada Sistem Operasi Bisnis

Dr. Agus Wibowo, M.Kom, M.Si, MM.

Dr. Agus Wibowo, M.Kom, M.Si, MM.

Kecerdasan Buatan Gabungan pada Sistem Operasi Bisnis

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
JL. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8120-83-3 (PDF)
9 786238 120833

Kecerdasan Buatan Gabungan pada Sistem Operasi Bisnis

Penulis :
Dr. Agus Wibowo, M.Kom, M.Si, MM.

ISBN : 9 786238 120833

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom.

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Segala Puji dan Syukur kami panjatkan selalu kepada Tuhan Yang Maha Esa atas Rahmat dan karunia-Nya yang sudah diberikan sehingga penulis bisa menyelesaikan buku yang berjudul ***"Kecerdasan Buatan (AI) Gabungan pada Sistem Operasi Bisnis"*** dengan baik. Kecerdasan Buatan (AI) memiliki potensi signifikan untuk mengubah cara banyak operasi dan proses dilakukan di dunia nyata. Ada banyak kemajuan signifikan di bidang AI, terutama di bidang Machine Learning (ML), dan hal ini telah menghasilkan banyak utilitas yang berguna, mulai dari pendekatan yang memungkinkan kita berbicara dengan mesin, dan komputer yang menggunakan vision hingga melakukan berbagai fungsi, seperti memungkinkan akses ke tempat yang aman. Namun, terlepas dari semua kemajuan besar yang dicapai di bidang AI, dapat dikatakan bahwa kita baru saja menyentuh permukaannya, dan masih banyak dampak yang akan terjadi. Meskipun beberapa bisnis telah banyak memanfaatkan AI untuk meningkatkan bisnis mereka, banyak bisnis yang sudah ada, seperti bank, jaringan ritel, perusahaan asuransi, dll., hanya melihat penggunaan AI yang terbatas dalam bisnis mereka.

Dalam buku ini, kami mengkaji masalah pembuatan model AI dari data yang disimpan di banyak lokasi berbeda, dan tidak dapat dengan mudah dipindahkan ke satu lokasi. Penerapan AI dalam situasi ini memerlukan pendekatan pembelajaran gabungan. Dalam pendekatan ini, model lokal dibuat di setiap situs dari data yang ada secara lokal dan model ini dipindahkan ke server yang menggabungkan semua model lokal ke dalam satu model. Pendekatan federasi tidak perlu terbatas pada pembuatan model, namun juga dapat digunakan pada tahap inferensi, ketika model digunakan untuk pengambilan keputusan. Dalam pendekatan inferensi gabungan, keputusan dapat dibagikan ke berbagai lokasi berbeda untuk meningkatkan kualitasnya.

Masalah lain yang menghalangi penerapan AI adalah kesenjangan antara teori AI dan penerapan teori tersebut ke dalam solusi dunia nyata. Pendekatan AI/ML sering kali dibahas dari sudut pandang ahli statistik, dan dari sudut pandang ilmuwan data atau pengembang perangkat lunak yang harus mengimplementasikan fungsi-fungsi tersebut. Penerapan AI pada permasalahan bisnis biasanya harus dilakukan oleh analis bisnis atau arsitek teknis senior yang mungkin tidak ingin terjebak dalam detail statistik atau kompleksitas paket pemrograman perangkat lunak. Untuk mendorong adopsi ke dalam bisnis, AI/ML harus dipisahkan dari bidang statistik dan pemrograman dan menggunakan istilah-istilah yang masuk akal bagi pengembang bisnis dan arsitek teknis.

Fokus kami adalah penggunaan AI dalam lingkungan bisnis dibandingkan dengan eksplorasi akademis seputar penanganan federasi. Beberapa tantangan berat yang perlu diatasi oleh sebuah bisnis tidak memungkinkan dilakukannya eksplorasi akademis yang baik. Pada saat yang sama, beberapa aspek pembelajaran gabungan yang sangat menarik dari sudut pandang eksplorasi ilmiah mungkin tidak terlalu menarik dari sudut pandang bisnis karena terdapat solusi sederhana untuk masalah tersebut.

Sebagai contoh dari masalah sebelumnya, penanganan data dengan format dan inkonsistensi yang berbeda memerlukan banyak waktu untuk membangun aplikasi bisnis gabungan, namun bergantung pada pendekatan yang tidak menghasilkan publikasi ilmiah yang menarik. Kami memiliki bab yang dikhususkan untuk membahas isu-isu terkait inkonsistensi data. Sebagai contoh masalah akademis yang tidak relevan dengan bisnis, sejumlah besar publikasi telah membahas isu serangan permusuhan dalam pembelajaran gabungan. Meskipun hal ini memberikan banyak peluang untuk penelitian yang menarik, dunia usaha sering kali dapat menghindari masalah ini dengan

menandatangani perjanjian bisnis dan menerima risiko sisa yang ada setelah perjanjian bisnis tersebut ditandatangani.

Buku ini terdiri dari sembilan bab yang disusun sebagai berikut, Bab 1 memberikan pengenalan non-matematis tentang AI dan Pembelajaran Mesin. Ini menjelaskan keseluruhan proses pembuatan model AI/ML dan penggunaan model tersebut di lingkungan bisnis yang berbeda. Bab 2 membahas berbagai skenario dalam lingkungan bisnis yang memerlukan penggunaan teknik pembelajaran gabungan. Ini membahas motivasi pembelajaran gabungan, dan juga memberikan perbedaan antara pembelajaran gabungan konsumen dan pembelajaran gabungan bisnis. Fokus buku ini adalah pada hal terakhir. Dan Bab 3 membahas serangkaian algoritma untuk pembelajaran gabungan yang dapat digunakan ketika beberapa asumsi penyederhanaan dapat dibuat mengenai lingkungan perusahaan. Asumsi ini mencakup fakta bahwa data tersedia dalam format yang sama di semua situs, data didistribusikan secara merata di semua situs, setiap situs melatih model pada waktu yang sama, dan bahwa semua situs memiliki kepercayaan penuh satu sama lain. Bab-bab berikutnya membahas pendekatan-pendekatan yang dapat digunakan ketika asumsi-asumsi ini tidak terpenuhi.

Bab 4 membahas permasalahan yang muncul ketika format data tidak sama di semua lokasi, dan pendekatan yang dapat digunakan untuk mengatasi tantangan penanganan data dengan format dan kualitas yang bervariasi di berbagai lokasi. Bab 5 membahas permasalahan yang timbul ketika data didistribusikan secara berbeda di berbagai lokasi, yaitu distribusi data tidak merata dan tidak identik. Bab ini membahas pendekatan yang dapat digunakan untuk mengatasi situasi tersebut. Bab 6 membahas isu-isu yang muncul ketika lokasi-lokasi tersebut tidak sepenuhnya percaya satu sama lain, dan mekanisme-mekanisme yang dapat mereka terapkan agar dapat bekerja sama meskipun tidak memiliki kepercayaan penuh satu sama lain.

Bab 7 membahas pendekatan yang dapat menangani pembelajaran bersama model ketika lokasi yang berbeda tidak dapat berkolaborasi dengan mudah untuk melatih model pada saat yang bersamaan. Bab 8 membahas pendekatan untuk berbagi intelijen di seluruh situs yang tidak dapat menyelesaikan ketidakcocokan atau ketidaksesuaian data yang ada di antara mereka sendiri. Hal ini dapat muncul dalam banyak situasi di mana situs berbeda mengamati entitas yang sama namun mengumpulkan informasi yang sangat berbeda tentang entitas tersebut. Bab 9 menunjukkan bagaimana berbagai pendekatan dan algoritme dapat digabungkan dalam beberapa kasus penggunaan spesifik yang muncul dalam bisnis. Kasus penggunaan diambil dari keterlibatan percontohan dan diskusi dengan klien di dunia bisnis.

Kami berharap buku ini bermanfaat bagi para para pembaca dalam memahami kekuatan pembelajaran gabungan dan pendekatan yang dapat diterapkan untuk memecahkan tantangan yang muncul dalam penggunaan AI dalam bisnis.

Semarang, Februari 2024
Penulis

Dr. Agus Wibowo, M.Kom, M.Si, MM

DAFTAR ISI

BAB 1
PENGANTAR KECERDASAN BUATAN

Kecerdasan Buatan (AI) mempunyai potensi untuk membuat peningkatan yang signifikan terhadap efektivitas dan efisiensi proses yang ada di spektrum aplikasi bisnis yang luas. Namun, seperti banyak istilah populer lainnya, definisi pasti mengenai AI tidak tersedia, sehingga menyebabkan perbedaan pendapat yang luas di antara anggota komunitas teknis mengenai teknologi apa saja yang merupakan bagian dari AI, dan apa saja yang harus dipertimbangkan di luar bidang AI. AI. Intelijen sendiri memiliki lebih dari 70 definisi dan setiap teks tentang AI memiliki definisi istilahnya masing-masing. Kami tidak ingin menentukan definisi AI yang lebih luas di masyarakat, namun akan berguna jika kami mendefinisikan Kecerdasan Buatan dalam konteks buku ini.

Fokus buku ini adalah penggunaan AI (dan khususnya penggunaan Kecerdasan Buatan Federasi) untuk meningkatkan operasi dalam skenario dunia nyata yang dihadapi dalam bisnis komersial dan lingkungan militer. Oleh karena itu, kami akan mendefinisikan AI dalam konteks spesifik ini, pertama dengan menggunakan model abstrak yang mewakili operasi dalam skenario dunia nyata, dan menjelaskan bagaimana model ini bekerja dengan dan tanpa AI. Hal ini akan memberikan definisi spesifik mengenai apa yang termasuk dalam cakupan AI dan apa yang berada di luar cakupan AI.

1.1 MODEL OPERASI BISNIS

Untuk cakupan buku ini, kami mendefinisikan skenario dunia nyata sebagai pengambilan keputusan dalam menjalankan operasi bisnis melalui implementasi berbasis perangkat lunak. Sifat sebenarnya dari operasi dan keputusan bisnis akan bergantung pada industri dan organisasi spesifik dalam industri tersebut. Sebagai contoh, bank mungkin ingin menentukan apakah permintaan transfer uang yang dibuat oleh nasabah adalah sah atau palsu. Contoh lain adalah ketika operator telepon menentukan apakah panggilan telepon yang dilakukan oleh penelepon harus dibiarkan tersambung atau diblokir karena penelepon tersebut mungkin melanggar beberapa norma.

Mendeteksi apakah suatu transaksi yang dilakukan oleh klien sah adalah contoh operasi yang diperlukan di banyak jenis bisnis, termasuk namun tidak terbatas pada bank, perusahaan asuransi, perusahaan telepon, dan toko ritel. Pemeriksaan awal terhadap transaksi penipuan dapat dilakukan dengan memeriksa catatan yang ada untuk memastikan bahwa pemrakarsa transaksi adalah klien resmi dari bisnis tersebut, telah menunjukkan kredensial yang tepat untuk mengautentikasi dirinya atas transaksi tersebut dan akunnya bereputasi baik. Pemeriksaan ini diperlukan namun tidak cukup. Kredensial dapat dicuri, beberapa pelanggan mungkin ditipu oleh penipu untuk melakukan transaksi yang tidak pantas dari akun mereka, dan beberapa penjahat mungkin telah membuat akun yang sah untuk melakukan transaksi penipuan. Memeriksa transaksi penipuan memerlukan tes tambahan untuk pola perilaku abnormal. Menentukan apa yang normal dan apa yang tidak normal merupakan bidang dimana penerapan teknik Kecerdasan Buatan bisa sangat berguna.

Keputusan lain yang muncul di banyak industri adalah menentukan pelanggan mana yang bernilai tinggi dan mana yang bernilai rendah. Pelanggan bernilai tinggi adalah pelanggan yang dapat menghasilkan keuntungan besar bagi bisnis, sedangkan pelanggan bernilai rendah mungkin menggunakan sumber daya perusahaan dalam jumlah yang tidak proporsional dan menimbulkan kerugian bagi perusahaan. Secara umum, perusahaan mana pun akan lebih memilih untuk memberikan layanan yang lebih baik kepada pelanggan sehingga menghasilkan lebih banyak keuntungan. Restoran ingin memberikan perhatian khusus kepada pelanggan yang memesan makanan mahal di menu, dan mungkin ingin mencegah pelanggan yang hanya memesan kopi dan menempati tempat duduk dalam waktu lama. Bank mungkin ingin memberikan insentif khusus kepada nasabah yang menyimpan dana dalam jumlah besar di rekening mereka dan tidak menyarankan nasabah yang memiliki saldo sangat sedikit di rekening mereka.

Namun, tidaklah mudah untuk menentukan siapa pelanggan yang bernilai tinggi. Misalnya, sebuah bank mungkin ingin memberikan status pilihan kepada anak seorang nasabah yang memiliki saldo signifikan di bank tersebut, meskipun anak tersebut saat ini adalah seorang pelajar yang tidak memiliki dana dalam jumlah besar di rekening perorangannya. Klasifikasi pelanggan ke dalam nilai tinggi atau rendah (atau sejumlah tingkatan) perlu dilakukan dengan memeriksa sifat transaksi mereka dengan pihak lain, bukan hanya rekening individual mereka.

Jenis keputusan bisnis lain yang perlu diambil oleh bisnis muncul selama operasi terkait dengan layanan dan dukungan pelanggan. Saat pelanggan meminta bantuan, bisnis perlu segera menentukan alasan pelanggan menelepon, dan mengarahkan panggilan tersebut ke sistem yang tepat. Sistem yang tepat dapat berupa salah satu dari banyak sistem otomatis yang menangani permintaan sederhana, atau salah satu dari banyak pakar manusia yang menangani permintaan rumit. Menentukan pakar manusia mana yang memiliki kombinasi keahlian dan ketersediaan terbaik memerlukan perangkat lunak untuk mengambil beberapa keputusan rumit.

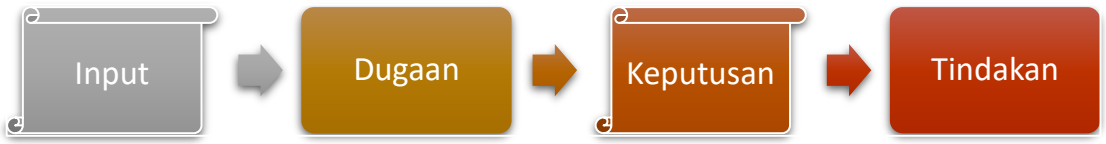
Selain operasi yang berhubungan dengan interaksi antara pelanggan dan bisnis, ada operasi bisnis yang menghadap ke dalam dan dilakukan untuk memastikan kelancaran fungsi perusahaan. Misalnya, ketika beberapa jenis pesan terlihat di jaringan yang menghubungkan mesin yang berbeda di suatu perusahaan, hal tersebut mungkin memerlukan konfigurasi jaringan untuk dimodifikasi agar dapat memblokir perangkat yang tidak berfungsi atau berbahaya di jaringan. Rumah sakit mungkin ingin memeriksa perkembangan pasiennya untuk menentukan jenis pengobatan mana yang lebih efektif, atau apakah jenis penyakit baru atau jenis virus baru sudah mulai muncul. Demikian pula, ketika motor penggerak AC di rumah sakit mulai mengeluarkan suara aneh, akar permasalahannya perlu didiagnosis dan proses perbaikan yang sesuai harus dimulai. Beberapa operasi lain yang sifatnya serupa diperlukan dalam bisnis yang berbeda, masing-masing memerlukan satu atau lebih keputusan yang harus diambil secara teratur.

Ada banyak keragaman dalam jenis keputusan bisnis dan operasi bisnis yang perlu dilakukan. Beberapa jenis operasi bisnis diperlukan di berbagai jenis bisnis, sementara jenis

lainnya mungkin hanya diperlukan di industri tertentu. Namun jenis operasi bisnis lainnya mungkin hanya diperlukan di departemen tertentu dalam industri tertentu. Namun demikian, sebagian besar operasi bisnis ini dapat dicirikan dengan model sederhana yang ditunjukkan pada Gambar 1.1. Operasi bisnis terdiri dari dua langkah, inferensi dan tindakan. Langkah inferensi mengubah masukan menjadi keputusan keluaran. Langkah tindakan mengimplementasikan keputusan yang dibuat. Seseorang dapat melihat langkah inferensi sebagai 'otak' dari operasi bisnis, sedangkan langkah 'tindakan' adalah 'otot' (lengan/kaki/tungkai) yang melakukan apa yang diperintahkan oleh otak.

Dalam contoh yang dibahas sebelumnya, operasi bisnis untuk memeriksa penipuan akan menghasilkan keputusan biner (apakah transaksi tersebut curang atau tidak). Tergantung pada keputusannya, beberapa tindakan akan diambil, misalnya. transaksi mungkin diperbolehkan untuk dilanjutkan, pemberitahuan penolakan dikirimkan, atau permintaan pemeriksaan audit otomatis dibuat.

Tujuan AI adalah untuk meningkatkan langkah inferensi dalam proses bisnis. Langkah inferensi diimplementasikan sebagai modul perangkat lunak yang mengambil masukan dan menghasilkan keputusan sebagai keluarannya. Baik masukan maupun keluarannya mungkin sederhana atau kompleks. Masukan ke modul perangkat lunak dapat berupa nilai tunggal, file yang berisi banyak nilai seperti spreadsheet, umpan dari sensor atau sekumpulan sensor, dan lain-lain. Outputnya juga dapat berupa nilai tunggal, kumpulan beberapa nilai, file dengan sekumpulan catatan, atau file yang dihasilkan dalam teks tidak terstruktur. Output ini adalah contoh keputusan yang dapat dibaca komputer.



Gambar 1.1 Model proses bisnis yang diidealkan.

Operasi yang lebih kompleks dapat disusun dengan menggabungkan banyak operasi sederhana, dengan keluaran dari satu operasi komponen dimasukkan ke dalam operasi komponen lainnya. Masukan yang sama dapat dimasukkan ke dalam beberapa operasi, menghasilkan grafik kompleks yang menghubungkan beberapa keputusan dan tindakan, dan operasi bisnis sangat kompleks yang terdiri dari banyak keputusan bisnis. Namun, kita dapat dengan mudah memahami konsep AI dengan berfokus pada satu komponen, dan implementasinya sebagai modul perangkat lunak.

Ada banyak cara berbeda untuk mengimplementasikan keputusan bisnis dalam perangkat lunak. Masing-masing cara berbeda untuk melakukan konversi akan memetakan masukan ke keluaran, dan semua implementasi akan menyediakan fungsionalitas yang setara. Namun, mungkin terdapat perbedaan yang signifikan dalam atribut bagaimana konversi terjadi, seperti keterampilan orang-orang yang diperlukan untuk mengimplementasikan proses bisnis, jenis informasi yang perlu diberikan kepada mereka, atau kecepatan dalam melakukan konversi. dimana proses tersebut dapat dilakukan. Sebagai contoh, satu

pendekatan untuk melakukan konversi dari input ke output mungkin melibatkan perekrutan tim pemrogram perangkat lunak, pendekatan lain mungkin memerlukan tim ilmuwan data, sedangkan pendekatan ketiga mungkin memerlukan tim yang terdiri dari orang-orang untuk memasukkan data ke dalam suatu sistem. lembar kerja. Pendekatan yang berbeda juga mungkin lebih efektif tergantung pada rentang nilai yang digunakan dalam proses pengambilan keputusan.

Kami akan mengilustrasikan beberapa pendekatan yang berbeda ini dengan versi operasi bisnis yang disederhanakan. Mari kita perhatikan kasus ketika seorang pemohon di bank ingin mengajukan pinjaman. Bank perlu menentukan jenis risiko apa yang harus dikaitkan dengan permohonan pinjaman ini. Dalam kehidupan nyata, penentuan risiko ini didasarkan pada banyak faktor berbeda, termasuk jangka waktu pinjaman, jumlah pinjaman, pendapatan pemohon, wilayah negara tempat orang tersebut tinggal, suku bunga yang berlaku, dan lain-lain. Sebagai gambaran, mari kita sederhanakan permasalahannya dan asumsikan bahwa bank menentukan risiko hanya berdasarkan dua faktor, yaitu jumlah pinjaman yang diminta, dan pendapatan tahunan pemohon. Bank kemudian akan menentukan risiko yang terkait dengan permohonan tersebut sebagai salah satu dari tiga kategori hijau, kuning, dan merah, di mana hijau berarti pinjaman tersebut memiliki risiko yang sangat kecil atau tanpa risiko, kuning berarti pinjaman tersebut memiliki risiko sedang, dan merah berarti pinjaman tersebut. sangat berisiko. Pada modul inferensi, bank akan menentukan peringkat risiko yang akan dikaitkan dengan aplikasi dan menghitung risiko berdasarkan dua nilai input, sedangkan pada modul tindakan, perangkat lunak bank akan menentukan bagaimana tepatnya mengimplementasikan keputusan yang telah diambil.

Pendekatan pertama yang dapat digunakan bank untuk menerapkan modul inferensi adalah dengan menyusun tabel risiko yang terkait dengan perbedaan nilai pendapatan dan jumlah pinjaman. Salah satu contoh tabel pencarian ini adalah yang ditunjukkan pada Tabel 1.1.

Tabel 1.1 Pendekatan tabel pencarian untuk keputusan bisnis. (Dalam RP. X.000)

Jumlah pinjaman	Pendapatan tahunan	Mempertaruhkan
< 10.000	> 20.000	Hijau
10.001 – 50.000	> 100.000	Hijau
> 50.000	< 100.000	Merah
> 50.000	25.000 – 100.000	Kuning
> 50.000	> 200.000	Hijau
> 100.000	< 200.000	Merah

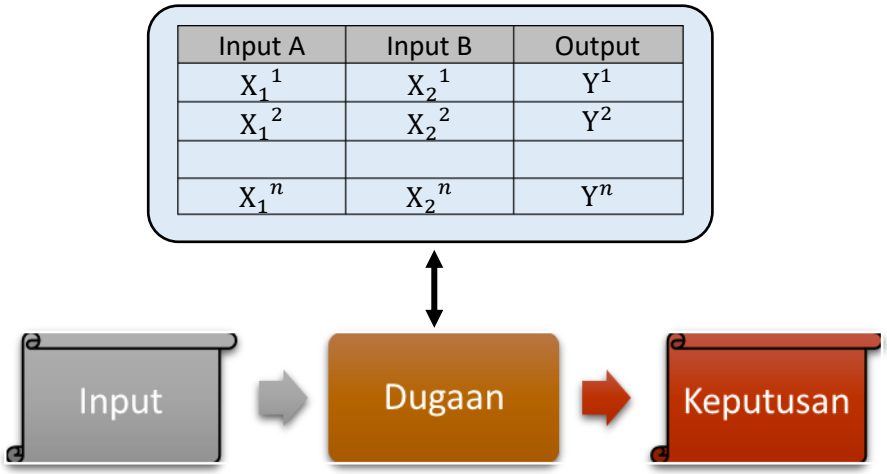
Setelah tabel ditentukan, proses inferensi terdiri dari mencari nilai hasil yang sesuai dengan masukan apa pun yang diberikan dalam tabel.

Mendefinisikan tabel seperti itu mungkin merupakan pendekatan yang baik dalam banyak kasus. Hal ini memungkinkan perangkat lunak untuk sekadar mencari informasi dalam tabel tersebut, dan menentukan peringkat risiko kredit suatu permohonan pinjaman. Operasinya akan mengikuti pendekatan yang ditunjukkan pada Gambar 1.2. Untuk menjaga

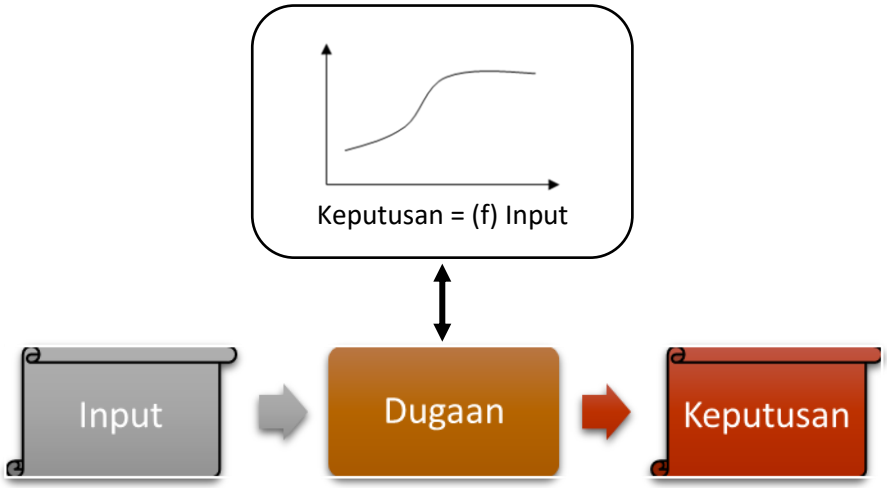
tabel yang menggerakkan perangkat lunak ini, bank perlu mempekerjakan sejumlah personel penilaian risiko pinjaman yang dapat menentukan entri yang tepat dalam tabel. Dibutuhkan juga seorang insinyur perangkat lunak untuk menulis dan memelihara kode yang menarik kesimpulan dari tabel.

Meskipun pendekatan berbasis tabel memiliki keuntungan karena mudah diterapkan, keterbatasan pendekatan ini juga harus jelas. Tabel tersebut perlu dibuat dan dipelihara berdasarkan pendapat ahli dari personel penilaian risiko. Pendapat-pendapat ini mungkin bias karena pandangan mereka dan bertentangan dengan risiko sebenarnya yang ada dalam proses pinjaman. Seiring waktu, tabel tersebut mungkin tidak mencerminkan risiko sebenarnya berdasarkan data historis mengenai gagal bayar pinjaman, dan mungkin tidak sinkron dengan kenyataan di lapangan. Selain itu, seiring berjalannya waktu, tabel tersebut mungkin bertambah besar dan mungkin sulit untuk dipelihara secara konsisten. Anda mungkin telah memperhatikan beberapa hal ketidakkonsistenan pada Tabel 1.1, dimana beberapa kombinasi pendapatan dan jumlah pinjaman saling tumpang tindih, dan tabel tersebut tidak mencakup beberapa kombinasi. Jika terdapat beberapa kolom dalam tabel, mendeteksi ketidakkonsistenan tersebut mungkin sulit dilakukan tanpa perangkat lunak untuk menganalisis dan mengidentifikasinya.

Pendekatan alternatif bagi bisnis adalah dengan mendefinisikan fungsi matematika yang memetakan nilai masukan yang berbeda ke skor risiko. Contohnya adalah perusahaan mendefinisikan pinjaman sebagai hijau jika rasio jumlah pinjaman terhadap pendapatan kurang dari 0,5, merah jika lebih dari 2, dan kuning jika rasio berada di antara kedua batas tersebut. Mendefinisikan fungsi seperti itu akan memberikan representasi ringkas dari informasi yang terkandung dalam tabel. Ketika masukan diberikan ke modul inferensi, modul tersebut menghitung fungsinya dan menghasilkan keluaran yang diinginkan seperti yang ditunjukkan pada Gambar 1.3.



Gambar 1.2 Pendekatan berbasis tabel untuk keputusan bisnis.



Gambar 1.3 Pendekatan berbasis fungsi.

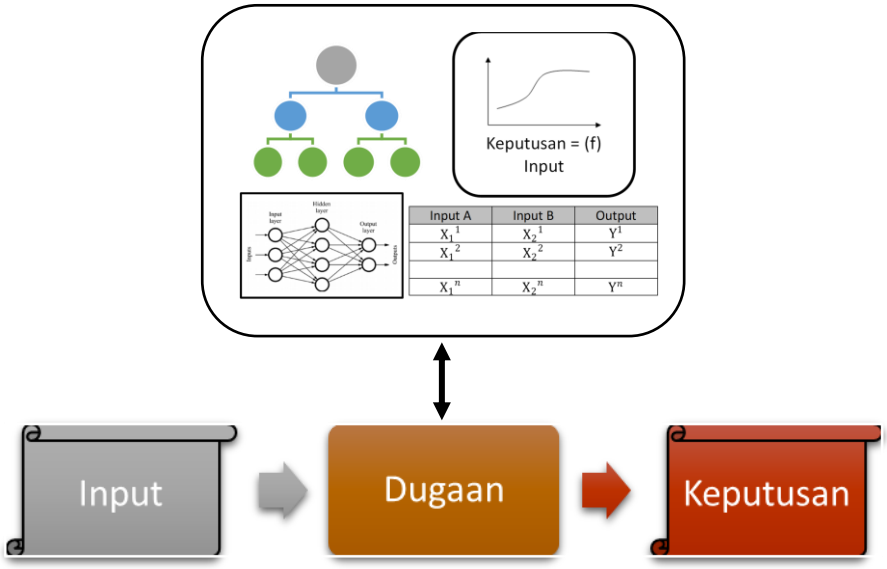
Ketika fungsi tersebut dapat diidentifikasi, pendekatan ini bekerja dengan sangat baik. Modul yang mengimplementasikan suatu fungsi, betapapun rumitnya, dapat dibuat melalui implementasi perangkat lunak dari fungsi tersebut.

Tabel dan fungsinya merupakan representasi alternatif dari 'model' yang berbeda. Sebuah model mewakili hubungan antara masukan dan keputusan. Ada banyak alternatif representasi model untuk merepresentasikan hubungan antara input dan output (keputusan). Beberapa dari berbagai representasi model adalah pohon keputusan, mesin vektor pendukung, jaringan saraf, kumpulan aturan, dll., dan beberapa di antaranya dijelaskan pada bagian selanjutnya dari bab ini. Model mengkodekan logika, yang dengannya modul perangkat lunak dapat menerjemahkan masukan menjadi keputusan untuk digunakan dalam proses bisnis. Penggunaan model untuk mengimplementasikan langkah inferensi ditunjukkan pada Gambar 1.4. Namun, sebelum inferensi dapat menggunakan model tersebut, model tersebut harus didefinisikan dengan cara tertentu.

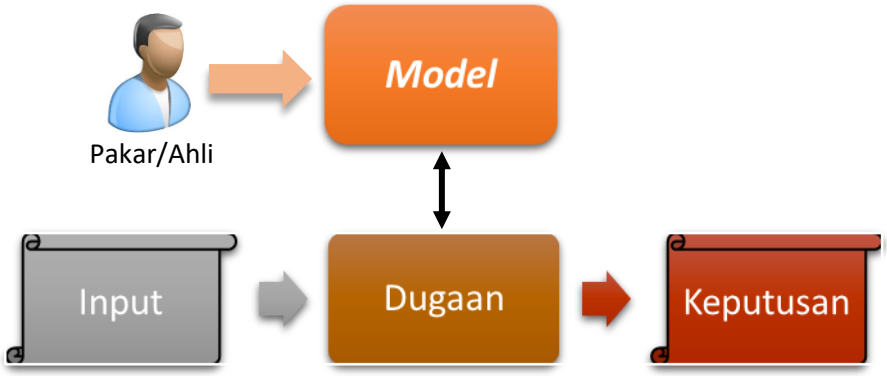
Salah satu pendekatan yang mungkin dilakukan adalah dengan menggunakan manusia ahli, atau tim ahli untuk mendefinisikan model seperti yang ditunjukkan pada Gambar 1.5. Dampaknya, model ini menangkap kecerdasan yang tertanam dalam pikiran para ahli, dan memungkinkan hal tersebut diimplementasikan sebagai modul perangkat lunak. Pendekatan untuk mendefinisikan model seperti itu disebut AI simbolik. Model didefinisikan dalam bentuk simbol-simbol yang mempunyai arti di dunia nyata, dan simbol-simbol tersebut dapat dimanipulasi oleh mesin. Contoh umum AI simbolik adalah sistem berbasis aturan.

Anda mungkin bertanya-tanya bagaimana pakar memperoleh kecerdasan yang mereka tangkap sebagai model. Dalam kebanyakan kasus, keahlian diperoleh melalui pengalaman. Bank yang aktif memberikan pinjaman dalam jangka waktu lama mengetahui tipe orang seperti apa yang lebih besar kemungkinannya untuk gagal membayar pinjamannya, dan tipe orang seperti apa yang lebih besar kemungkinannya untuk melunasi pinjamannya berdasarkan riwayat masa lalu mereka. Pengetahuan sejarah ini ditangkap dalam model mereka, misalnya. Perusahaan asuransi jiwa menyusun tabel ekstensif mengenai

kemungkinan kematian orang-orang pada kelompok umur yang berbeda untuk menilai risiko mereka.



Gambar 1.4 Pendekatan berbasis model.

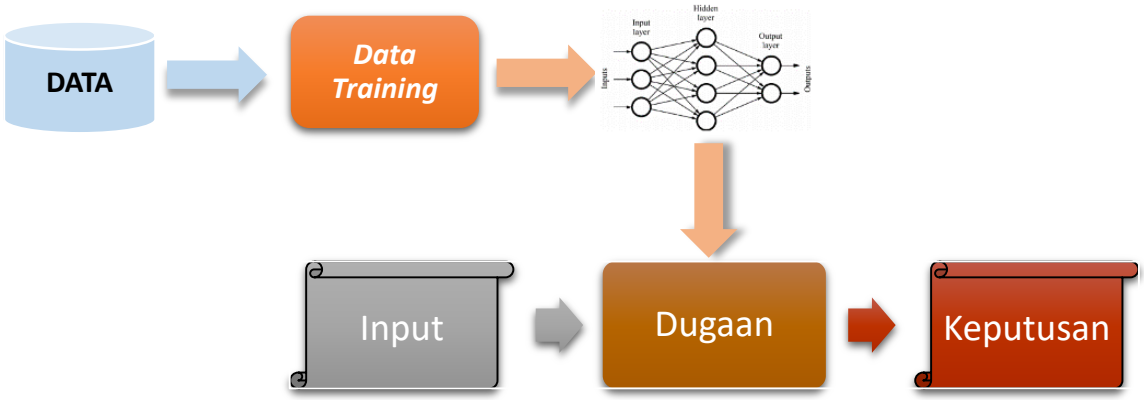


Gambar 1.5 Pendekatan pembelajaran simbolik.

Alat dapat diciptakan untuk membantu manusia dalam mendefinisikan model secara lebih efisien. Sebagai contoh, alat dapat dibuat untuk membantu memeriksa konsistensi dalam suatu model, memeriksa kesalahan umum yang dapat dilakukan suatu model, dan memungkinkan validasi model berdasarkan kebenaran dasar yang diketahui. Namun, dalam AI simbolik, manusia pada akhirnya bertanggung jawab untuk mentransfer pengetahuannya ke dalam model.

Pendekatan AI simbolik telah ada selama beberapa dekade, namun pendekatan ini menjadi sulit digunakan ketika model menjadi sangat kompleks. Dalam situasi kehidupan nyata, logika model biasanya memiliki kompleksitas yang jauh lebih besar dibandingkan contoh sederhana yang telah kita tunjukkan di atas. Mungkin ada lusinan masukan yang diperlukan untuk menilai risiko permohonan pinjaman, misalnya. ketersediaan agunan, riwayat pembayaran masa lalu dan skor risiko kredit pemohon, dll. Mendefinisikan model simbolis dengan sejumlah besar masukan adalah tugas yang tidak sepele bagi manusia mana pun.

Di sebagian besar bisnis, terdapat data historis yang signifikan serta sekumpulan pakar yang memiliki pengetahuan tentang pengoperasian bisnis. Daripada hanya mengandalkan keahlian manusia untuk mendefinisikan model berdasarkan pengalaman mereka, sistem perangkat lunak mungkin bisa menganalisis semua data historis dan menangkap wawasan sejarah sebagai model. Sebagai contoh, sebuah bank mungkin memiliki banyak ahli yang dapat melihat parameter permohonan pinjaman dan menentukan risiko yang melekat dalam permohonan tersebut untuk memutuskan apakah pinjaman tersebut harus disetujui atau tidak. Ia juga memiliki sejarah pinjaman selama bertahun-tahun dan informasi tentang pinjaman mana yang gagal bayar dan pinjaman mana yang telah dibayar kembali sepenuhnya. Data historis ini dapat dianalisis dan diubah menjadi model. Ada banyak keuntungan dalam menganalisis data historis, yang bisa lebih komprehensif dibandingkan keahlian manusia mana pun. Analisis data dapat menghindari bias yang melekat dalam pengambilan keputusan oleh manusia, dapat melihat kumpulan data yang lebih besar dibandingkan pengalaman individu manusia, dan dapat menemukan hubungan dan pola yang mungkin tidak terlihat jelas.



Gambar 1.6 Pendekatan pembelajaran mesin.

Penambangan informasi sejarah menggunakan program komputer otomatis untuk membuat model disebut pembelajaran mesin. Langkah pembuatan model biasanya disebut pelatihan atau pembelajaran, dan model tersebut kemudian dapat digunakan untuk melakukan tugas inferensi. Proses yang terlibat dalam pembelajaran mesin ini ditunjukkan pada Gambar 1.6. Jenis model populer yang sering dibuat adalah jaringan saraf, sehingga teknik pembelajaran mesin juga disebut sebagai pembelajaran saraf.

Baik pakar manusia maupun proses penambangan data memiliki peran yang berharga dan saling melengkapi dalam pengembangan model. Dalam kehidupan nyata, pendekatan hibrid yang mengandalkan kombinasi keahlian manusia (pendekatan AI Simbolik) dan pembelajaran mesin kemungkinan besar akan berguna dalam meningkatkan solusi masalah bisnis. Salah satu jenis pendekatan hibrid yang spesifik adalah kasus dimana manusia menuliskan keahliannya dalam dokumen atau laporan. Perangkat lunak komputer yang mampu memproses pemrosesan bahasa alami kemudian dapat digunakan untuk menganalisis

dokumen-dokumen tersebut dan membuat model. Data untuk pembelajaran mesin dalam hal ini adalah hasil tangkapan pengetahuan manusia yang direpresentasikan dalam format tidak langsung, seperti teks. Terkadang, pengetahuan berbasis teks dapat ditangkap dalam dokumen berbeda yang mungkin ada di lingkungan terdistribusi seperti Internet. Perangkat lunak komputer dapat merayapi Internet, mengumpulkan dokumen yang relevan, mengubahnya menjadi model, dan menambah model tersebut menggunakan analisis data historis. Ada banyak cara berbeda untuk membuat model AI, dan dalam praktiknya semua model yang berguna akan menggunakan pendekatan hibrid yang menggabungkan AI simbolik dengan beberapa pendekatan pembelajaran mesin.

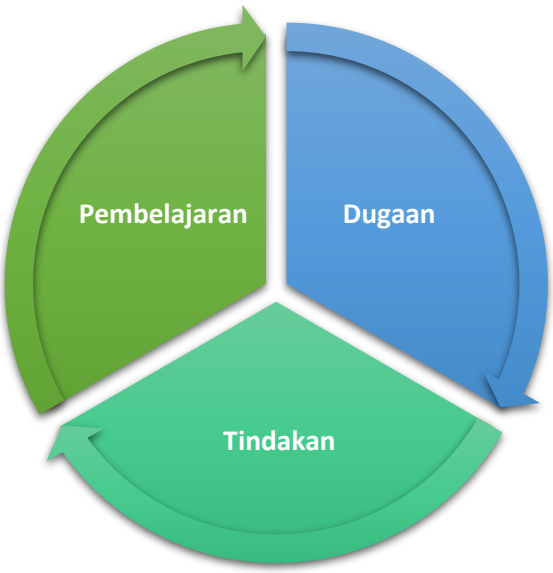
Untuk tujuan buku ini, kami mendefinisikan metode berbasis AI sebagai pendekatan untuk mengimplementasikan proses bisnis yang didorong oleh penggunaan model yang diturunkan menggunakan analisis data berbasis perangkat lunak.

1.2 SIKLUS BELAJAR→MENYIMPULKAN→BERTINDAK

Di hampir setiap perusahaan yang memiliki pengalaman operasi selama beberapa periode, sejumlah besar data dikumpulkan dari operasi sehari-harinya. Namun, kemampuan manusia yang ahli untuk menangkap pola yang muncul dalam data masih terbatas. Hasilnya, pembelajaran mesin menjadi mekanisme yang menjanjikan untuk mengekstrak model dari data. Ekstraksi pola kompleks dari data, bila digabungkan dengan beberapa domain pengetahuan dan keahlian yang tersedia dari manusia, dapat menghasilkan model canggih yang dapat meningkatkan pengoperasian sistem secara keseluruhan.

Pembelajaran mesin sangat efektif ketika seseorang memiliki akses ke data bagus yang dapat digunakan untuk membuat model. Ketika data tersebut sudah tersedia, namun hubungan fungsional antara berbagai komponen data sulit ditentukan melalui pemeriksaan manusia, pembelajaran mesin akan menjadi pendekatan yang lebih disukai. Ketika data pelatihan sulit dikumpulkan, dan pola yang diperlukan untuk inferensi dapat dengan mudah ditentukan oleh seorang ahli, AI simbolik akan menjadi pendekatan yang lebih disukai.

Dalam praktiknya, setiap model tidak sempurna, dan dapat dibuat lebih akurat seiring berjalannya waktu. Model tersebut dapat dipelajari, kemudian digunakan untuk inferensi, dan hasil inferensi tersebut digunakan untuk mengambil tindakan tertentu. Tindakan itu sendiri akan mencerminkannya ke dalam beberapa metrik operasional proses bisnis. Reaksi klien, rekan bisnis, dan karyawan terhadap tindakan tertentu dapat menyebabkan perubahan dalam hubungan antara masukan dan keluaran. Seiring waktu, sistem akan mendapatkan data tambahan yang mungkin memerlukan pelatihan ulang model. Oleh karena itu, lebih baik kita menganggap aplikasi berbasis AI memiliki siklus hidup di mana fase pembelajaran, inferensi, dan tindakan saling menginformasikan. Model siklus hidup ini awalnya diperkenalkan untuk operasi militer namun valid di sebagian besar konteks bisnis ditunjukkan pada Gambar 1.7.



Gambar 1.7 Siklus hidup proses bisnis yang mendukung AI.

Fase Belajar. Pada fase Learn, model AI dibuat ulang atau diadaptasi dari model yang sudah ada. Dengan menggunakan paket perangkat lunak untuk menganalisis data, proses tersebut menciptakan model yang mampu mengubah masukan menjadi keluaran. Konversi dilakukan agar sesuai dengan hubungan yang terdapat pada data pelatihan. Proses pelatihan ini bisa sangat intensif komputasi karena menganalisis data dalam jumlah besar. Setelah model dibangun, proses pembelajaran juga akan mencakup langkah-langkah tambahan yaitu pengujian keakuratan data pada validasi data. Untuk penilaian yang tepat, data validasi tidak boleh merupakan bagian dari data pelatihan.

Fase Kesimpulan. Pada fase Infer, data masukan digunakan untuk menghasilkan keputusan keluaran untuk proses bisnis. Inferensi adalah kemampuan model yang dihasilkan dalam fase pembelajaran untuk diterapkan pada titik data baru untuk menghasilkan keputusan. Langkah inferensi dapat terdiri dari serangkaian aktivitas, di mana tahap awal dari alur tersebut akan terdiri dari konversi masukan yang diberikan ke sistem menjadi kumpulan masukan yang diharapkan oleh model AI. Secara umum, data masukan yang dihasilkan dalam menjalankan bisnis mungkin tidak sesuai dengan asumsi masukan yang dibuat oleh model AI, dan konversi fitur perlu dilakukan secara eksplisit. Tahap terakhir dari pipeline melakukan langkah sebenarnya dalam memanggil model AI.

Fase Bertindak. Pada fase Act, output dari fase Infer digunakan untuk mengambil tindakan atau melaksanakan keputusan. Jenis tindakan/keputusan dapat bervariasi, dan mungkin melibatkan masukan dari manusia, perangkat lunak lain, atau mesin lain.

Pada prinsipnya, model yang sama dapat didefinisikan oleh pakar dan juga dipelajari oleh mesin selama fase Learn. Namun, beberapa model lebih mudah didefinisikan oleh manusia, misalnya. seperangkat aturan atau sekumpulan tabel dengan kolom berbeda. Beberapa jenis model lain, yang memerlukan banyak bobot kompleks dan keputusan percabangan yang rumit, akan sulit didefinisikan oleh manusia, dan lebih baik dihasilkan menggunakan teknik pembelajaran mesin.

1.3 PROSES MEMBUAT MODEL AI

Membuat model AI/ML memerlukan pendekatan sistematis untuk melakukan serangkaian aktivitas. Proses untuk proses bisnis yang merupakan adaptasi dari proses untuk operasi multidomain militer, meliputi:

- Mengidentifikasi tugas AI/ML yang dapat ditingkatkan dengan penggunaan teknik AI/ML. Tidak semua tugas cocok untuk perbaikan tersebut. Secara umum, tugas-tugas yang memerlukan model yang sangat sederhana dapat diimplementasikan melalui program perangkat lunak, atau dengan pendekatan seperti pencarian tabel. Namun, untuk tugas yang modelnya tidak dapat ditangkap dengan mudah, dan jika data berkualitas baik untuk pelatihan tersedia atau dapat diperoleh dengan mudah, penggunaan teknik ML mungkin lebih tepat.
- Merancang AI/ML untuk memenuhi tujuan tugas dalam batasan yang diperkirakan dalam pengaturan operasional yang diharapkan. Desain ini memerlukan pemilihan model, keputusan apakah pendekatan simbolik atau pembelajaran mesin akan digunakan, dan kriteria apa yang harus dipenuhi agar tugas AI/ML dapat digunakan pada model tersebut.
- Dapatkan dan kurasi data pelatihan/validasi yang relevan, jika tugas didasarkan pada ML. Data pelatihan digunakan untuk membuat model, sedangkan data validasi digunakan untuk menilai performa model. Idealnya, data pelatihan dan data validasi harus independen. Namun, karena pengadaan data seringkali merupakan aktivitas yang paling memakan waktu di seluruh proses AI/ML, kumpulan data pelatihan sering kali dibagi menjadi dua kumpulan berbeda, satu digunakan untuk pelatihan dan satu lagi digunakan untuk validasi.
- Melatih model dalam berbagai kondisi, termasuk menggunakan data yang kotor, dinamis, dan menipu yang akan ada di lingkungan operasional. Penting agar pelatihan model mencakup sebanyak mungkin situasi yang mungkin ditemui di lingkungan operasional. Secara umum, model tidak akan berkinerja baik jika diminta untuk memprediksi informasi dalam lingkungan yang sangat berbeda dari lingkungan tempat mereka dilatih.
- Validasi AI/ML dalam kondisi realistis untuk memastikan model sesuai tujuan. Validasi dilakukan dengan menggunakan data validasi, dan diasumsikan bahwa data validasi mencerminkan lingkungan operasional.
- Menganalisis AI untuk memahami dan memprediksi kinerjanya serta menilai keselamatan, keamanan, dan ketahanannya. Pada tahap proses inilah metode simbolik memiliki keunggulan dibandingkan metode berbasis ML, karena analisis model simbolik biasanya lebih mudah dibandingkan dengan model berbasis pembelajaran mesin.
- Menentukan kasus penggunaan yang diperbolehkan untuk tugas yang mendukung AI/ML untuk membatasi cara penggunaannya (termasuk tujuan memastikan tugas tersebut sesuai dengan pedoman etika), menentukan pengaturan pembagian yang diperbolehkan, bagaimana dan kapan model yang dipelajari dapat diadaptasi atau

diperbarui , dan di mana serta kapan model dapat dilatih, kesimpulan dapat dibuat, dan tindakan dapat diambil.

Elemen kunci dari proses ini adalah menilai bagian mana dari proses bisnis yang layak untuk dilengkapi dengan AI, tingkat otonomi apa yang diinginkan, dan apa peran manusia dalam keseluruhan proses bisnis yang digerakkan oleh AI.

1.4 PERAN MANUSIA DALAM AI

Salah satu kekhawatiran yang sering terlihat di media populer mengenai AI adalah bahwa teknik AI dapat menggantikan manusia, sehingga dapat menyebabkan pergolakan sosial yang luar biasa. Oleh karena itu, penting untuk mendiskusikan dan menilai peran manusia dalam kerangka tugas yang didukung AI untuk operasi bisnis. Mengingat Learn - Infer - Act sebagai putaran dasar proses AI, keterlibatan manusia diperlukan pada berbagai tahapan siklus.

1.4.1 Peran Manusia dalam Fase Belajar

Selama fase pembelajaran, sistem perlu mempelajari model dari data pelatihan, atau meminta manusia memberikan pengetahuan yang dipelajari ke dalam model. Peran manusia dalam mendefinisikan model terlihat jelas pada pendekatan AI simbolik, dimana manusia memberikan pengetahuannya kepada sistem. Masukan manusia juga penting dalam pendekatan berbasis ML.

Tantangan utama dalam pembelajaran mesin untuk pembuatan model adalah sensitivitas model yang dipelajari terhadap data pelatihan yang tersedia. Kualitas model bergantung pada kualitas data pelatihan yang disediakan. Keterlibatan manusia yang signifikan diperlukan untuk memproses data dan memastikan bahwa data tersebut memiliki kualitas yang cukup tinggi untuk melatih model yang baik. Upaya manusia dan kebosanan ini dapat dikurangi dengan alat yang membantu pengumpulan dan analisis data yang dikumpulkan, namun langkah pengumpulan data adalah sesuatu yang sulit untuk diotomatisasi dengan kondisi teknologi saat ini.

Tantangan lain yang terkait dengan fase pembuatan model adalah algoritma pelatihan dapat menyesuaikan model dengan data secara berlebihan. Data tersebut mungkin memiliki pola yang dapat diambil oleh model, sehingga dapat memberikan hasil yang salah dan tidak diinginkan. Keterlibatan manusia diperlukan untuk memastikan bahwa model pembelajaran tidak mengambil pola-pola yang tidak diinginkan tersebut. Sebagai contoh, model untuk mengenali buah mungkin menggunakan warna merah sebagai ciri pembeda apel karena semua gambar yang digunakan dalam pelatihan menggunakan gambar apel merah, bukan bentuknya. Intervensi manusia diperlukan untuk memastikan bahwa pola yang dipelajari model dari data pelatihan sudah tepat, dan tidak mengambil pola palsu.

Dengan kondisi teknologi saat ini, AI dalam proses bisnis paling baik digunakan sebagai asisten untuk meningkatkan proses pembangunan model yang akan dilakukan manusia. Analisis berbasis data berbasis komputer dapat mengidentifikasi banyak pola menarik yang sulit diekstraksi oleh manusia ahli dari data. Namun, pakar manusia akan dapat menentukan apakah beberapa pola tersebut palsu dan pola mana yang berguna.

Untuk menilai kegunaan suatu model, model tersebut harus dapat dimengerti oleh manusia. Beberapa jenis model lebih mudah dipahami dibandingkan yang lain, dan kemampuan interpretasi model merupakan aspek penting untuk diperiksa oleh manusia dan untuk digunakan dalam aplikasi bisnis. Akibatnya, beberapa model yang mungkin sangat baik dalam menjalankan fungsinya, namun tidak dapat dipahami atau dijelaskan, mungkin kurang diminati dari sudut pandang bisnis dibandingkan dengan model yang dapat dipahami dan keluarannya dapat dijelaskan, meskipun model tersebut dapat dimengerti. berkinerja buruk pada skala absolut dalam metrik akurasi atau kinerjanya.

1.4.2 Peran Manusia dalam Fase Infer dan Act

Pada tahap Infer (Dugaan) dari siklus Pembelajaran → Dugaan → Kegiatan, langkah inferensi dapat sepenuhnya diotomatisasi. Dapat dikatakan bahwa manusia tidak perlu mempunyai peran apapun dalam proses inferensi yang menghasilkan suatu keputusan. Tahap tindakan dalam siklus ini, bagaimanapun, adalah langkah dimana keterlibatan manusia menjadi diinginkan tergantung pada tugas bisnis yang sedang dilakukan.

Dalam banyak proses bisnis, keluaran dari tahap inferensi adalah keputusan yang bersifat nasihat bagi manusia. Manusia harus memutuskan tindakan yang tepat untuk diambil berdasarkan keluaran dari proses pengambilan keputusan, dan merupakan bagian integral dari keseluruhan siklus. Model Human in the Loop (HIL) ini merupakan norma di banyak proses bisnis yang mendukung AI.

Model alternatif untuk melibatkan manusia pada saat inferensi adalah model Human Over the Loop (HOL). Dalam model ini, fase inferensi dan tindakan yang dihasilkan dilakukan secara otomatis, namun manusia memiliki peran pengawasan atas fase inferensi dan tindakan. Model HOL dapat dilihat pada mobil self-driving otonom dengan manusia di belakang kemudi yang dapat mengambil alih tugas mengemudi kapan saja, atau pada auto-pilot untuk pesawat terbang dimana pilot dapat mengambil alih sesuai kebutuhan. Pemilihan pendekatan yang digunakan untuk tindakan, apakah harus sepenuhnya manual, Human in the Loop, Human Over the Loop, atau sepenuhnya otomatis bergantung pada konsekuensi dari tindakan yang salah yang diambil dan apa risiko relatif dari pemilihan tindakan yang salah. antara manusia dan proses otomatis.

Untuk tindakan di mana manusia lebih mungkin melakukan kesalahan, dan dampak dari tindakan yang salah tidak signifikan, otomatisasi penuh mungkin merupakan pendekatan yang tepat untuk diikuti. Sebagai contoh, untuk tugas yang berulang, seperti mengirim email dengan template yang tetap, manusia lebih cenderung melakukan kesalahan dibandingkan sistem otomatis, dan dampak dari email yang salah tidak signifikan dalam banyak kasus. Tugas itu lebih baik dilakukan secara otomatis.

Dalam tugas di mana mesin lebih mungkin melakukan kesalahan, misalnya. dalam proses pengambilan keputusan yang kompleks untuk menangani pelanggan yang mengalami gangguan emosi, dan dampak kehilangan pelanggan sangat penting bagi bisnis karena pelanggan tersebut merupakan pelanggan bernilai tinggi, penanganan manusia secara menyeluruh mungkin merupakan pilihan terbaik. Di antara kedua ekstrem ini, mekanisme

human in the loop dan human on the loop dapat diikuti untuk memberikan trade-off terbaik antara dampak dan risiko.

1.5 REPRESENTASI MODEL AI

Model AI sebenarnya merupakan cara alternatif untuk merepresentasikan fungsi yang mengambil beberapa masukan dan menghasilkan keputusan keluaran. Ada banyak cara berbeda untuk merepresentasikan fungsi tersebut. Kita telah memeriksa beberapa teknik berbeda untuk merepresentasikan fungsi tersebut, termasuk representasi sebagai tabel dan representasi sebagai fungsi matematika. Di bagian ini, kita melihat berbagai representasi alternatif model AI lainnya.

Mengingat keragaman model AI yang berbeda-beda, bagian ini tidak mempertimbangkan semua kemungkinan variasi model AI, namun hanya mempertimbangkan beberapa jenis umum yang dapat diterapkan secara luas dalam operasi bisnis.

1.5.1 Representasi Fungsional

Representasi paling umum dari model AI adalah sebagai sebuah fungsi. Untuk bagian ini, kita dapat berasumsi bahwa suatu fungsi terdiri dari beberapa nilai masukan, misalkan N masukan nilai $x_1, x_2, \ldots x_N$ dan memiliki keluaran y. Fungsi umum yang direpresentasikan di sini adalah fungsi $y = f\ (x_1, x_2, \ldots x_N)$.

Ketika fungsi ini harus direpresentasikan dan dimasukkan ke dalam perangkat lunak komputer, fungsi tersebut dapat direpresentasikan sebagai beberapa parameter yang memberikan pendekatan yang sesuai untuk menghitung fungsi sesuai kebutuhan. Fungsi tersebut dapat dimodelkan dalam asumsi yang berbeda mengenai bagaimana keluaran bergantung pada masukan. Dalam beberapa kasus, kita dapat berasumsi bahwa terdapat hubungan linier antara fitur masukan dan fitur keluaran, yaitu hubungan tersebut dapat ditangkap melalui persamaan:

$$y = \alpha_0 + \sum_{i=1}^{i=N} \alpha_i x_i$$

Model AI dalam hal ini akan terdiri dari N + 1 parameter dari $\alpha_0, \alpha_1, \ldots \ldots \alpha_N$. Metode umum untuk menemukan model AI untuk hubungan linier adalah dengan menggunakan teknik regresi linier.

Dalam kehidupan nyata, tidak semua hubungan bersifat linier. Hubungan juga dapat digambarkan sebagai persamaan non-linier antara pangkat yang berbeda dari variabel masukan, atau sebagai hasil perkalian atau pembagian nilai masukan yang berbeda. Seperangkat parameter dapat mewakili kombinasi kompleks apa pun dari hubungan antara masukan dan keluaran. Sebagai contoh, hubungan antara input dan output yang sama dimana setiap input mempunyai pangkat M max dapat direpresentasikan dalam hubungan berikut.

$$y = \beta_0 + \sum_{j=1}^{j=M} \beta_1, j{x_1}^j + \sum_{j=1}^{j=M} \beta_2, j{x_2}^j + \cdots + \sum_{j=1}^{j=M} \beta_N, j{x_N}^j$$

Hal ini dapat direpresentasikan sebagai matriks berukuran Nx (M + 1) dengan memecah suku konstanta β_0 menjadi konstanta M + 1 yang menjumlahkannya.

Pendekatan umum lainnya untuk menangkap hubungan non-linier adalah dengan mengambil log atau eksponensial dari variabel keluaran atau masukan. Kemudian, regresi dapat diplot antara logaritma keluaran dan masukan atau kombinasi lainnya. Pendekatan regresi logistik digunakan di banyak bidang dan menyediakan cara untuk menangkap hubungan non-linier.

Ada banyak cara lain untuk merepresentasikan suatu fungsi menggunakan sekumpulan parameter dalam model, sebagai kombinasi fungsi lain, atau sebagai jumlah, produk input, atau sebagai representasi eksponensial atau log. Biasanya, representasi fungsional matematis baik digunakan untuk model yang mewakili fenomena fisik, misalnya ketika seseorang mencoba memahami pengoperasian komponen mekanis seperti kinerja mesin mobil, kondisi peralatan mekanis di lantai pabrik, atau memahami informasi yang dihasilkan oleh sensor dalam skenario aplikasi Internet of Things (IoT).

Representasi fungsional dapat berguna dalam banyak situasi dunia nyata, mulai dari menentukan pengendalian yang tepat dan pengaturan konfigurasi untuk pengendalian fisik peralatan dan mesin, membuat keputusan keuangan seperti memperkirakan risiko gagal bayar yang terkait dengan pinjaman, memperkirakan nilai sebuah rumah, memperkirakan konfigurasi yang akan memaksimalkan kinerja pusat data atau jaringan, dll. Ini juga merupakan representasi paling umum dari hubungan antara masukan dan keluaran, dan dapat digunakan untuk memperkirakan model lainnya. Namun, untuk beberapa kasus tertentu, jenis representasi model lain mungkin lebih cocok, khususnya ketika beberapa nilai masukan atau keluaran bukan numerik.

1.5.2 Tabel Keputusan

Meskipun representasi fungsional adalah cara yang baik untuk mendapatkan representasi umum dari fenomena apa pun dan menangkap hubungannya, banyak pertimbangan bisnis tidak akan dimodelkan oleh hubungan yang mungkin mudah direpresentasikan dalam formulasi matematis matematis. Tabel risiko yang terkait dengan pinjaman usaha yang ditunjukkan pada Tabel 1.1 memiliki tiga kategori merah, kuning dan hijau sebagai keluarannya. Seseorang masih dapat menghitung fungsi matematika dan mengasosiasikan tiga warna untuk rentang output fungsi yang berbeda, namun fungsi tersebut tidak mungkin memiliki representasi yang sangat sederhana. Grafiknya akan berubah secara tiba-tiba, dan memiliki banyak diskontinuitas. Fenomena bisnis yang tidak terlalu bergantung pada perangkat fisik namun pada pengalaman manusia, hubungan sosial, dan firasat sulit untuk dimodelkan sebagai fungsi matematika sederhana yang halus.

Input dan output dapat berupa nilai numerik atau nilai kategorikal. Nilai numerik adalah angka yang biasanya memiliki batas bawah dan atas bergantung pada konteks bisnis, dan tidak ada batas yang menjadi kasus khusus. Nilai kategorikal adalah nilai yang menganggap nilai sebagai salah satu kategori terpisah yang berbeda, mis. nama. Output pada Tabel 1.1 yang terdiri dari tiga nama warna bersifat kategorikal.

Sebaliknya, cara yang lebih baik untuk memodelkan fungsi-fungsi ini adalah dengan mencantumkannya dalam format tabel seperti yang ditunjukkan pada Tabel 1.1. Representasi model AI ini adalah tabel keputusan. Saat digunakan untuk merepresentasikan fungsi dengan input $x_1, x_2, \ldots\ x_N$ dan keluaran y, tabel keputusan akan memiliki hingga $2N + 1$ kolom. Setiap variabel masukan yang bersifat numerik akan memiliki dua kolom, satu menandai nilai terendah dari variabel masukan, dan yang lainnya menandai variabel maksimum dari variabel masukan tersebut. Setiap variabel masukan yang bersifat kategorikal akan memiliki satu kolom dengan kategori nilainya.

Setiap baris dalam tabel memberikan nilai keluaran jika data masukan berada dalam kondisi yang ditentukan oleh batas dan nilai baris tersebut. Tabel keputusan dapat dilihat sebagai representasi kompak dari data pelatihan, yang juga merupakan representasi tabel dengan kolom untuk setiap masukan dan kolom tambahan untuk keluaran. Tabel keputusan merangkum kumpulan data pelatihan yang memiliki jumlah baris jauh lebih besar menjadi representasi yang lebih ringkas. Selama proses pemadatan, mungkin juga terdapat informasi yang bertentangan dalam data pelatihan dalam arti bahwa dua titik data dengan masukan yang identik atau sangat mirip dapat menghasilkan keluaran yang sangat berbeda. Proses pelatihan akan menyelesaikan konflik-konflik tersebut untuk mendapatkan sistem yang konsisten dalam memetakan masukan menjadi keluaran. Ketika keputusan keluaran bersifat kategoris, tabel keputusan dapat memberikan representasi yang baik untuk mendefinisikan model AI.

1.5.3 Pohon Keputusan

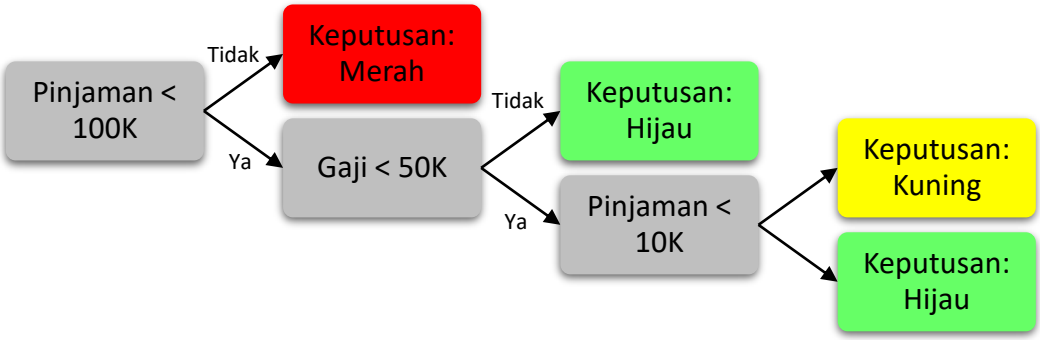
Pohon keputusan terdiri dari struktur mirip pohon yang terdiri dari beberapa node. Struktur seperti pohon berarti terdapat simpul awal atau simpul akar, dan setiap simpul mungkin mempunyai beberapa anak. Node yang tidak mempunyai anak disebut node daun. Setiap node di pohon berisi pengujian pada satu variabel masukan, dan keluaran pengujian menentukan anak mana dari node tersebut yang akan dipilih untuk pengujian berikutnya. Pada setiap node daun, terdapat nilai prediksi keluarannya.

Contoh sederhana ditunjukkan pada Gambar 1.8. Setiap node pada pohon mempunyai dua anak. Untuk mengetahui resiko pinjaman menjadi merah, hijau atau kuning. Pada node akar, yang merupakan node paling kiri dalam diagram, pengujian memeriksa apakah jumlah pinjaman kurang dari ambang batas (dalam contoh 100K). Jika tes gagal, pinjaman ditandai dengan warna merah. Jika tes berhasil, tes kedua dilakukan untuk memeriksa apakah gaji pelamar kurang dari 50K/tahun. Jika tes gagal (yaitu gaji di atas 50K/tahun), pinjaman ditandai hijau. Jika pengujian berhasil, algoritme menjalankan pengujian lain yang memeriksa apakah pinjaman kurang dari 10K. Jika tes berhasil, pinjaman ditandai dengan warna hijau. Jika tes gagal, pinjaman ditandai dengan warna kuning.

Pohon keputusan di dunia nyata memerlukan beberapa lusin parameter sebagai masukan, dan jauh lebih kompleks daripada contoh sederhana yang ditunjukkan di atas. Namun, mereka bekerja berdasarkan prinsip melakukan pengujian pada setiap node dan mengikuti cabang pohon yang berbeda, bergantung pada hasil pengujian.

Pohon keputusan dapat dipandang sebagai representasi tabel keputusan yang kompak dan efisien. Jika pohon keputusan bersifat biner, yaitu setiap node mempunyai dua anak, maka pohon yang melakukan pengujian N dalam satu lintasan pohon dapat mewakili model yang memerlukan $2N$ baris dalam tabel keputusan.

Pelatihan pohon keputusan memerlukan data masukan pelatihan yang dipindai untuk menemukan rangkaian pengujian yang paling efisien yang dapat menghasilkan pengambilan keputusan yang cepat. Ada berbagai algoritma efisien yang mencoba menemukan representasi terbaik berdasarkan perbandingan sifat statistik kumpulan data, dan dampak penambahan node dalam proses pengambilan keputusan.



Gambar 1.8 Contoh pohon keputusan.

1.5.4 Kumpulan Aturan

Kumpulan aturan adalah representasi lain dari model AI, yang terdiri dari sejumlah aturan. Dalam setiap aturan, terdapat serangkaian kondisi yang melibatkan variabel masukan berbeda, dan nilai keluaran jika kondisinya benar. Kondisi adalah hubungan logis yang dapat menampung antara satu atau lebih variabel masukan. Berbeda dengan pohon keputusan di mana setiap baris berisi sesuatu untuk setiap variabel masukan, aturan hanya dapat memuat sebagian dari variabel masukan, dan mengabaikan yang lain.

Beberapa aturan juga dapat memprediksi variabel masukan lain alih-alih memprediksi keluaran secara langsung. Jenis aturan ini dapat digunakan untuk memprediksi nilai masukan baru yang kemudian dapat dimasukkan ke dalam aturan lain untuk memprediksi keluaran pada langkah berikutnya. Aturan juga bisa dirantai. Jika semua aturan dalam kumpulan aturan memprediksi keluaran secara langsung, maka aturan tersebut dapat dengan mudah diubah menjadi tabel keputusan yang setara dengan mengonversi setiap aturan menjadi baris tabel keputusan. Setiap jalur dalam pohon keputusan juga dapat dilihat sebagai aturan tunggal yang terdiri dari semua pemeriksaan sepanjang jalur yang dilakukan secara berurutan.

Kumpulan aturan adalah salah satu mekanisme paling awal yang mewakili model AI. Mereka dapat ditulis dengan tangan oleh seorang ahli atau dapat ditentukan dengan menganalisis data pelatihan yang tersedia. Mereka telah digunakan dalam berbagai jenis sistem, dan telah terbukti bermanfaat dalam banyak sistem bisnis.

1.5.5 Jaringan Syaraf Tiruan

Jaringan saraf merupakan representasi model AI yang terinspirasi oleh cara kerja sistem saraf manusia. Jaringan saraf terdiri dari beberapa node yang saling berhubungan dalam suatu jaringan di mana setiap node (disebut neuron) memiliki banyak masukan dan satu keluaran. Suatu fungsi mendefinisikan bagaimana masukan dipetakan ke dalam keluaran. Node-node tersebut biasanya disusun dalam beberapa lapisan, masing-masing lapisan terdiri dari beberapa node, dan keluaran dari satu lapisan dimasukkan ke dalam masukan dari lapisan berikutnya. Lapisan pertama mengambil variabel masukan yang berbeda $x_1, x_2, \ldots x_N$ dan lapisan terakhir memprediksi nilai keluaran y. Fungsi setiap node ditentukan oleh sekumpulan parameter yang umumnya disebut bobot. Sebagai contoh, sebuah neuron dapat mengambil beberapa masukan bernilai biner dan menghasilkan nol jika jumlah masukan melebihi ambang batas, dan satu sebaliknya. Ambang batasnya akan menjadi bobot neuron. Pengumpulan seluruh bobot pada semua neuron pada lapisan berbeda dapat membuat jaringan saraf mampu memodelkan hampir semua jenis hubungan antara variabel masukan dan keluaran.

Fleksibilitas jaringan saraf semakin diperluas dengan fakta bahwa ada beberapa fungsi yang dapat digunakan dengan setiap neuron, lapisan yang berbeda dapat menggunakan jenis fungsi yang berbeda, dan topologi jaringan yang sangat kaya dapat dihasilkan dari fungsi tersebut. Hasilnya, jaringan saraf dapat mewakili fungsi apa pun dan seiring berjalannya waktu, beberapa algoritma yang efisien untuk melatih fungsi tersebut telah dikembangkan. Baru-baru ini, jaringan saraf dalam yang terdiri dari beberapa lapisan neuron yang terhubung bersama telah terbukti sangat berhasil dalam berbagai aplikasi, termasuk pengenalan gambar, konversi ucapan ke teks, dan pemrosesan bahasa alami. Peningkatan daya komputasi yang terjangkau selama bertahun-tahun dan akses ke data pelatihan dalam jumlah besar telah menyebabkan ledakan penerapan jaringan saraf di banyak domain berbeda.

Jaringan saraf hadir dalam berbagai bentuk, dan bergantung pada topologi interkoneksinya, jumlah lapisan dan jenis fungsi yang dikodekan dalam neuron pada lapisan berbeda didefinisikan ke dalam salah satu dari banyak kategori berbeda. Ini termasuk jaringan saraf feed forward, jaringan saraf berulang, jaringan saraf konvolusional, jaringan saraf memori jangka pendek jangka panjang, mesin Boltzmann, jaringan Hopfield, dll. Setiap varian memiliki fitur yang memungkinkan mereka mengatasi kelas masalah tertentu dengan sangat baik. Sebagai contoh, jaringan saraf konvolusional berfungsi dengan baik untuk memahami konten gambar dan mengklasifikasikannya, sedangkan jaringan memori jangka pendek atau LSTM berfungsi dengan baik untuk memodelkan jenis data deret waktu. Kami tidak akan membahas secara rinci jenis jaringan saraf tertentu. Pendekatan federasi yang dibahas dalam buku ini dapat diterapkan untuk semua jenis jaringan saraf. Kemajuan dalam kemampuan jaringan saraf dalam adalah salah satu pendorong utama minat baru-baru ini terhadap penerapan AI untuk berbagai masalah bisnis.

1.5.6 Representasi Berbasis Transformasi Matriks

Beberapa model AI dapat direpresentasikan sebagai matriks, yaitu array angka NxM yang dikalikan dengan variabel input $x_1, x_2, \ldots x_N$ dan menghasilkan kumpulan variabel transformasi lainnya $v_1, v_2, \ldots v_M$. Serangkaian operasi lain, mis. jumlah tertimbang pada

variabel keluaran kemudian dapat memprediksi keluaran y. Pemetaan variabel yang diubah menjadi keluaran juga dapat dilakukan oleh model AI lain, misalnya. pohon keputusan atau jaringan saraf, yang secara efektif menggabungkan dua model AI.

Penerapan umum model transformasi matriks adalah untuk mengurangi jumlah fitur menjadi sekumpulan variabel yang lebih kecil yang memiliki ketergantungan silang yang lebih kecil satu sama lain. Transformasi ini biasanya dilakukan dengan menggunakan metode analisis komponen utama untuk variabel masukan yang bersifat numerik dan menggunakan analisis korespondensi berganda untuk variabel masukan yang bukan numerik tetapi mengambil beberapa nilai diskrit. Kedua pendekatan ini menghasilkan format matriks untuk representasi model. Representasi matriks kemudian dapat digunakan sebagai langkah pra-pemrosesan sebelum mempelajari model lainnya.

1.5.7 Model Berbasis Jarak

Metode berbasis transformasi matriks juga dapat dipandang sebagai konversi variabel masukan menjadi ruang transformasi dalam banyak dimensi. Ruang yang ditransformasikan kemudian dapat digunakan sebagai representasi dimana jarak antar ruang yang ditransformasikan dapat dihitung. Pendekatan umum untuk menghitung jarak antara dua titik $v_1, v_2, \ldots v_M$ dan $u_1, u_2, \ldots u_M$ dihitung menggunakan apa yang disebut perkalian titik di antara vektor-vektor yang berbeda, yaitu jarak antara dua titik dihitung $v_1.u_1, v_2.u_2, \ldots v_M.u_M$. Transformasi ke ruang di mana jarak ditentukan dapat dilakukan dengan menggunakan transformasi matriks atau, lebih umum, menggunakan jaringan saraf. Pendekatan umum untuk memetakan masukan ke representasi vektor menggunakan jaringan saraf adalah melalui auto-encoder, jaringan saraf yang terdiri dari bagian encoder yang memetakan masukan asli ke vektor, dan decoder yang memetakan vektor kembali ke aslinya. format. Perbandingan masukan yang direkonstruksi dapat digunakan sebagai ukuran kebaikan untuk melatih auto-encoder.

Jika metrik jarak dapat ditentukan di antara titik-titik yang berbeda, maka metode berdasarkan jarak dapat digunakan di antara titik-titik yang berbeda. Salah satu pendekatan spesifik yang digunakan dalam banyak pendekatan adalah menemukan kelompok titik yang berdekatan satu sama lain. Hal ini memungkinkan penentuan masukan yang berdekatan, dan mengatasi fungsi seperti pengelompokan 1.8.2, deteksi anomali 1.8.3, dan pemfilteran 1.8.5, yang dalam banyak kasus dapat dilakukan tanpa mengetahui keluaran apa yang seharusnya.

Metode berbasis jarak juga dapat digunakan untuk mendefinisikan permukaan dalam ruang yang ditransformasikan yang dapat menyediakan mekanisme untuk memprediksi nilai kategorikal dari keluaran y. Algoritme populer untuk menghitung mekanisme ini mencakup konsep mesin vektor pendukung (SVM), yang menemukan permukaan dalam ruang yang ditransformasikan untuk memisahkan berbagai jenis ruang keluaran dengan menggunakan teknik efisien untuk mendukung hubungan linier dan non-linier antara masukan dan keluarannya.

1.5.8 Model Mesin Keadaan Hingga

Model berbasis mesin negara mendefinisikan sistem berada dalam beberapa keadaan, masing-masing negara ditentukan oleh kombinasi input yang dipantau. Sistem dapat

didefinisikan berada di beberapa K keadaan $s_1, s_2, \dots s_K$, dimana setiap keadaan ditentukan oleh beberapa kombinasi variabel masukan $x_1, x_2, \dots x_N$. Setiap keadaan dikaitkan dengan tindakan spesifik yang harus diambil ketika sistem berada dalam keadaan tersebut, dan tindakan tersebut berhubungan dengan keputusan y. Masing-masing negara bagian juga dikaitkan dengan seperangkat aturan yang menentukan kapan sistem harus mengubah negara bagian.

Mesin negara dapat didefinisikan sebagai deterministik atau probabilistik. Dalam mesin negara deterministik, aturan deterministik menentukan keadaan selanjutnya yang akan diambil. Dalam mesin keadaan probabilistik, pilihan tahap selanjutnya bersifat acak, didorong oleh beberapa distribusi probabilitas. Jumlah semua probabilitas untuk keadaan berikutnya haruslah 1. Model berbasis mesin negara telah berhasil digunakan dalam beberapa aplikasi praktis, termasuk kontrol robotik, mobil tanpa pengemudi, permainan komputer, atau pembuatan kasus uji untuk pengujian perangkat lunak. Teknik pelatihan untuk model berbasis mesin negara mencakup pendekatan yang disebut pembelajaran automaton dan pembelajaran penguatan.

1.5.9 Kesetaraan Model AI

Meskipun kami telah memberikan gambaran singkat tentang beberapa jenis model AI yang populer, ada beberapa jenis model AI lainnya. Ini termasuk model stokastik yang meliputi rantai Markov, Model Markov Tersembunyi, jaring Petri, dll. Dalam setiap jenis model AI, terdapat beberapa variasinya sendiri. Jenis model yang sama, bila digunakan dalam domain aplikasi berbeda, dapat disebut dengan nama berbeda.

Meskipun ada banyak variasi berbeda dalam cara merepresentasikan model, semua model adalah ekuivalen dan seseorang dapat mengubah satu jenis model menjadi jenis model lainnya. Efektivitas proses pelatihan model, akses terhadap paket perangkat lunak yang ada yang menghasilkan jenis model tertentu, dan latar belakang personel bisnis yang membuat model merupakan faktor penentu utama dalam memilih jenis model yang akan digunakan.

Kesetaraan ini tidak berarti bahwa semua model AI akan bekerja dengan baik pada kumpulan data yang sama. Performa setiap model bergantung pada banyak aspek berbeda, termasuk hyper-parameter spesifik yang dipilih untuk mendefinisikan arsitektur model. Hyper-parameter tersebut mencakup jumlah node dalam pohon keputusan, atau jumlah lapisan dan neuron di setiap lapisan yang dipilih untuk jaringan saraf, atau jumlah dimensi ruang transformasi yang dipilih dalam metode berbasis matriks, dll. Waktu yang dibutuhkan untuk melatih model, kemampuan untuk menangkap model untuk menangkap pola-pola berbeda yang ada dalam data, dan kemampuan untuk memahami dan menjelaskan cara kerja model merupakan faktor-faktor yang akan berbeda untuk model yang berbeda. Namun, model apa pun dapat digunakan sebagai bagian dari proses inferensi, dan definisi model terbaik untuk tugas apa pun akan sangat bergantung pada spesifikasi tugas tersebut. Karena fleksibilitasnya dalam merepresentasikan berbagai jenis fungsi, dalam buku ini kami akan fokus pada representasi fungsional model dan jaringan saraf sebagai contoh model AI yang perlu difederasi. Namun, konsep dan pendekatan federasi yang sama juga berlaku untuk jenis model AI lainnya.

1.6 MODEL PENDEKATAN PEMBELAJARAN

Ada banyak pendekatan berbeda untuk melatih model AI. Setiap pendekatan menghasilkan penciptaan model AI. Dalam beberapa kasus, representasi model AI sangat erat kaitannya dengan jenis pendekatan pembelajaran. Dalam setiap pendekatan, terdapat banyak algoritme berbeda yang dapat digunakan untuk melatih model, dan setiap algoritme memiliki kelebihan dan kekurangannya masing-masing.

Pendekatan pembelajaran suatu model dicirikan oleh data pelatihannya dan atribut data pelatihan yang tersedia. Beberapa pendekatan model pembelajaran yang umum adalah sebagai berikut:

- ***Pembelajaran yang Diawasi:*** Ketika data pelatihan terdiri dari data masukan dan keluaran yang sesuai (label), seseorang dapat menggunakan metode pembelajaran yang diawasi untuk membuat model. Label dapat mengidentifikasi kelas diskrit, nilai komputasi berkelanjutan, status, dll. Pembelajaran yang diawasi dianggap berdasarkan tugas karena penggunaannya bergantung pada jenis tugas yang dilakukan. Beberapa tugas yang berhasil menggunakan gaya ini adalah klasifikasi (misalnya gambar) dan regresi (misalnya prediksi pelacakan target).
- ***Pembelajaran Tanpa Pengawasan***: Ketika data pelatihan terdiri dari informasi masukan tanpa label, seseorang dapat menggunakan metode pembelajaran tanpa pengawasan untuk membuat model. Contoh permasalahannya meliputi pengelompokan, deteksi anomali, pembelajaran aturan asosiasi, reduksi dimensi (mengidentifikasi fitur data yang paling signifikan). Pembelajaran tanpa pengawasan didasarkan pada data.
- ***Pembelajaran semi-supervisi***: Jika sebagian data pelatihan memiliki label, dan ada pula yang tidak diberi label, skema pembelajaran semi-supervisi dapat digunakan. Ini berfokus pada penggabungan data berlabel dan tidak berlabel, biasanya menggunakan heuristik untuk memberi label pada data tidak berlabel. Pembelajaran transfer tanpa pengawasan dapat digunakan ketika tugasnya sama tetapi domain operasionalnya berbeda dari domain yang dilatih. Contoh soal meliputi: analisis ucapan, pengorganisasian pengetahuan, klasifikasi, regresi. Pembelajaran semi-supervisi didorong oleh data tugas hibrid.
- ***Pembelajaran Transfer:*** Ketika data pelatihan tersedia di satu domain, namun hanya data pelatihan terbatas yang tersedia di domain terkait, pembelajaran transfer dapat digunakan. Hal ini melibatkan adaptasi model yang telah dilatih sebelumnya ke masalah target baru yang terkait. Hal ini memperluas pembelajaran transfer tanpa pengawasan ke kasus-kasus di mana tugas-tugas yang dilatih dan operasional berbeda (pembelajaran transfer induksi) atau ketika tugasnya sama tetapi tugas operasionalnya berbeda dari domain yang dilatih (pelatihan transduktif). Contoh permasalahannya meliputi: klasifikasi dokumen dan pembelajaran dari simulasi. Pembelajaran transfer ditujukan untuk domain baru.
- ***Reinforcement Learning***: Ketika keluaran terdiri dari tujuan sistem, seseorang dapat menggunakan pembelajaran penguatan untuk mencoba membuat model yang

bergerak menuju tujuan tersebut. Hal ini berfokus pada pembelajaran dari pengalaman, biasanya dengan memberikan hadiah, dan menyeimbangkan eksplorasi dan eksploitasi pengetahuan saat ini. Pembelajaran penguatan dapat dianggap sebagai cara untuk bereaksi terhadap lingkungan untuk mencapai suatu tujuan. Contoh permasalahannya meliputi: mobilitas robot, permainan, pemeliharaan prediktif. Pembelajaran penguatan didorong oleh tujuan.

Ringkasan pendekatan pembelajaran mesin dan kapan menerapkannya berdasarkan karakteristik data pelatihan yang tersedia ditunjukkan pada Tabel 1.2.

Tabel 1.2 Data pelatihan dan pendekatan pembelajaran mesin.

Karakteristik Data Pelatihan	Pendekatan Pembelajaran
Berisi label keluaran	Pembelajaran yang Diawasi
Tidak mengandung label keluaran	Pembelajaran Tanpa Pengawasan
Data terbatas dengan label keluaran dan data besar tanpa label	Pembelajaran semi-supervisi
Data terbatas tetapi data besar tersedia di area terkait	Pembelajaran Transfer
Serangkaian tindakan dengan dampaknya terhadap sistem	Pembelajaran Penguatan

1.7 ASPEK PEMBELAJARAN SPASIAL DAN TEMPORAL

Proses pembelajaran siklus belajar-menyimpulkan-bertindak dapat diterapkan dalam beberapa cara berbeda. Dalam satu mode, pembelajaran bisa bersifat statis, hal-hal dipelajari satu kali biasanya dalam lingkungan terkendali atau perusahaan, dan dianggap statis atau kuasi-statis. Dalam mode lainnya, pembelajaran harus berkelanjutan dan model diperbarui secara konstan dalam sistem yang diterapkan. Cara umum untuk penerapan pembelajaran adalah:

- *Pembelajaran Offline atau Batch*. Dalam hal ini, model AI dibuat dalam pengaturan offline. Jika metode simbolik digunakan, masukan manusia dikumpulkan dalam metode offline. Jika pembelajaran mesin digunakan, data pelatihan dikumpulkan secara offline, model dapat dilatih pada sebagian data pelatihan yang tersedia, dan diuji performanya pada bagian lain dari data tersebut. Selanjutnya, model yang dilatih dibekukan untuk diterapkan di mana inferensi menggunakan model yang telah dilatih sebelumnya. Pendekatan ini paling baik digunakan dalam situasi yang kurang dinamis/tidak pasti karena pendekatan ini memiliki verifikasi yang lebih baik dan performa yang diketahui, namun tidak beradaptasi dengan situasi baru atau ketidaksesuaian antara model yang dilatih dan lingkungan. Model tersebut, yang bersifat statis dan tidak dapat diubah selama penerapan, dapat diperbarui setelah beberapa waktu dengan model yang lebih baru yang dilatih secara offline.

- *Pembelajaran Online atau Berkelanjutan.* Latih/uji secara offline, namun algoritme pembelajaran diterapkan untuk mengadaptasi model dengan data masukan baru. Oleh karena itu, perusahaan terus belajar dengan beradaptasi secara bertahap seiring dengan tersedianya data setelah diterapkan ke dalam konteks operasionalnya. Pendekatan ini paling baik digunakan dalam situasi yang dinamis/tidak pasti dimana risiko adaptasi tidak sebanding dengan kebutuhan untuk bereaksi terhadap keadaan yang tidak diketahui atau berubah.

Dari perspektif spasial, pembelajaran dapat terjadi di satu lokasi pusat atau banyak lokasi berbeda. Pilihan utama untuk belajar adalah sebagai berikut:

- ***Terpusat:*** Pembelajaran Terpusat membangun modelnya di satu lokasi dimana semua data pelatihan dan validasi berada. Setelah model AI divalidasi dan diverifikasi, model tersebut dibekukan sebelum diterapkan. Bentuk pembelajaran ini sangat berguna dengan kumpulan data besar yang dapat memanfaatkan komputasi kinerja tinggi dan jaringan berkecepatan tinggi, namun tidak sesuai bila ada kendala berbagi data pelatihan.
- ***Terdistribusi***: Pembelajaran Terdistribusi memperluas pendekatan terpusat dengan pemrosesan terdistribusi untuk mengatasi kendala komputasi dan/atau penyimpanan, sering kali menggunakan teknik pemrosesan paralel. Hal ini sering terjadi digunakan di lingkungan perusahaan dengan jaringan area lokal cepat yang dapat mendistribusikan data ke banyak prosesor.
- ***Federasi***: Pembelajaran Federasi adalah pelatihan kolaboratif dimana data pelatihan tidak dipertukarkan. Dalam pembelajaran Federasi, pelatihan didistribusikan tanpa pertukaran data pelatihan berbeda dengan ML Terdistribusi di mana data pelatihan dipindahkan untuk diproses oleh beberapa komputer. ML gabungan digunakan untuk mengatasi kendala dalam berbagi data pelatihan (kebijakan, keamanan, kendala berbagi) dan/atau kapasitas jaringan yang tidak mencukupi.

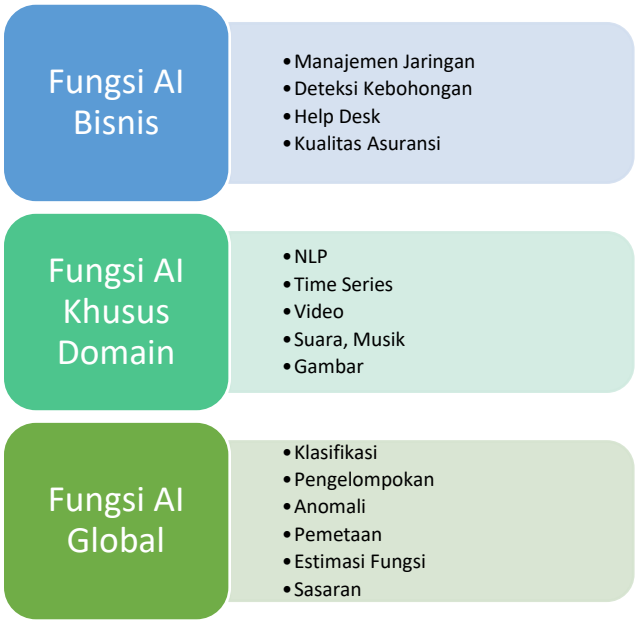
Dari perspektif spasial yang sama, pilihan utama untuk inferensi adalah sebagai berikut:

- ***Terpusat:*** Inferensi terpusat melakukan inferensinya di satu lokasi dimana model untuk inferensi tersedia. Biasanya lokasi yang sama digunakan untuk tugas pelatihan. Jika lokasi tersebut memiliki banyak mesin, kemungkinan server yang digunakan untuk inferensi tidak sama dengan server yang digunakan untuk melatih model, namun perbedaannya biasanya tidak terlihat oleh sistem mana pun di luar lokasi.
- ***Edge:*** Inferensi tepi melakukan inferensinya di lokasi yang dekat dengan lokasi penghasil data. Situs inferensi biasanya berupa sistem yang terletak di beberapa jenis batas, misalnya. sistem yang menghubungkan lokasi terpencil ke lokasi pusat yaitu tepi. Dalam inferensi tepi, model dipindahkan dari lokasi pelatihan ke sistem di tepi dan digunakan untuk inferensi. Banyak situs edge mungkin secara bersamaan menggunakan sistem mereka untuk situs lokalnya.
- ***Terfederasi***: Inferensi terfederasi adalah variasi inferensi tepi di mana situs tepi yang berbeda dapat berkolaborasi untuk meningkatkan proses pengambilan keputusan.

Fokus buku ini adalah pada AI gabungan, yang mencakup pembelajaran gabungan dan inferensi gabungan. AI gabungan merupakan persyaratan penting dalam banyak situasi dunia nyata. Pada bab berikutnya, kita melihat beberapa skenario umum yang mungkin memerlukan penerapan solusi pembelajaran mesin gabungan.

1.8 FUNGSI YANG DIAKTIFKAN AI

Teknik berbasis AI dapat digunakan untuk banyak operasi bisnis yang berbeda. Beberapa dari operasi bisnis ini bersifat spesifik pada industri atau perusahaan tertentu, sementara beberapa operasi bisnis dapat dipandang dapat diterapkan pada sejumlah besar perusahaan di beberapa industri. Banyak dari operasi bisnis ini didasarkan pada beberapa fungsi umum. Kita dapat membagi fungsi-fungsi ini menjadi tiga kategori besar yang ditunjukkan dalam arsitektur berlapis seperti yang ditunjukkan pada Gambar 1.9.



Gambar 1.9: Arsitektur fungsi yang mendukung AI.

- *Fungsi AI Umum: Ini* adalah fungsi yang memberikan kemampuan umum berdasarkan penggunaan teknik AI/ML.
- *Fungsi berbasis tipe data:* Fungsi ini menerapkan AI/ML pada tipe data tertentu. Fungsi-fungsi ini mungkin menggunakan kembali beberapa fungsi umum, namun sering kali menyesuaikan fungsi umum dengan cara yang dioptimalkan untuk menangani tipe data tertentu.
- *Fungsi Bisnis:* Fungsi bisnis dibangun berdasarkan fungsi umum serta fungsi berbasis tipe data untuk memecahkan masalah bisnis tertentu. Fungsi bisnis dapat menggunakan kembali algoritma dan pendekatan dari fungsi umum dan fungsi berbasis tipe data.

Perhatikan bahwa masing-masing fungsi ini cukup rumit untuk dijadikan buku yang lengkap. Tujuan kami di bagian ini adalah untuk memberikan gambaran luas kepada pembaca dengan menggunakan terminologi yang konsisten sehingga konsep utama dalam buku ini, yang melakukan pembelajaran model dan inferensi di berbagai situs, dapat dijelaskan secara rinci selanjutnya. Untuk menjelaskan pengoperasian pembelajaran mesin gabungan, kami akan fokus pada fungsi AI umum, bukan fungsi berbasis tipe data atau fungsi bisnis.

Fungsi berbasis tipe data adalah fungsi berbasis AI/ML yang mendukung satu tipe data input tertentu. Fungsi-fungsi ini mencakup pemrosesan data deret waktu, data grafik, suara ucapan, suara non-ucapan, gambar, video, dokumen teks, dan pemrosesan lalu lintas jaringan. Dalam setiap pemrosesan tipe data, beberapa kemampuan mungkin diturunkan dari properti spesifik tipe data input. Hal ini kemudian dapat dikombinasikan dengan teknik berbasis AI/ML untuk memecahkan masalah tertentu.

Fungsi bisnis memecahkan masalah spesifik yang mungkin timbul dalam industri tertentu. Masalah seperti ini dibangun berdasarkan fungsi umum serta fungsi tipe data, dan mungkin menggunakan kemampuan berbasis AI/ML hanya untuk aktivitas tertentu. Beberapa fungsi bisnis ini dapat digunakan di berbagai industri. Contoh fungsi tersebut mencakup penilaian risiko pinjaman, dan deteksi transaksi penipuan di industri keuangan, optimalisasi kualitas produksi di industri manufaktur, dan Intelijen, Pengawasan, dan Pengintaian (ISR) di lingkungan militer. Beberapa fungsi yang dapat diterapkan di berbagai industri termasuk mendeteksi serangan intrusi jaringan, menangani permintaan layanan pelanggan, mencocokkan resume pelamar dengan lowongan pekerjaan terbuka, dan meningkatkan manajemen jaringan dan sistem untuk instalasi TI. Ada banyak aplikasi khusus domain dan banyak fungsi bisnis AI, yang masing-masing terlalu banyak untuk dijelaskan. Kami hanya akan menjelaskan beberapa jenis umum fungsi AI generik lebih lanjut pada berbagai subbagian di bawah ini:

1.8.1 Klasifikasi

Klasifikasi adalah penerapan teknik AI di mana keputusan keluaran mengidentifikasi masukan sebagai milik salah satu di antara beberapa kelas yang biasanya telah ditentukan sebelumnya. Klasifikasi diperlukan dalam banyak aplikasi bisnis, seperti menandai permohonan pinjaman sebagai berisiko tinggi atau berisiko rendah, menentukan apakah gambar suatu produk sesuai dengan produk bagus atau produk cacat, menentukan apakah suara yang keluar dari perangkat adalah bunyi mencicit, bunyi dentingan atau pekikan, memeriksa gambar radiologi untuk menentukan apakah gambar tersebut menunjukkan kondisi sehat atau salah satu dari sejumlah penyakit, memeriksa apakah perangkat lunak yang diunduh dari Internet merupakan salah satu dari banyak jenis virus yang dikenal, dll.

Untuk klasifikasi, data pelatihan terdiri dari beberapa contoh input milik masing-masing kelas. Selama proses pelatihan, sistem memeriksa masukan yang berbeda untuk mengekstrak pola yang mencirikan masukan yang dimiliki setiap jenis kelas, dan menyimpannya sebagai model AI. Banyak jenis model AI yang dapat digunakan untuk menangkap pola yang mendefinisikan setiap kelas, mulai dari jaringan saraf dan pemisahan

statistik hingga pohon keputusan dan aturan. Beberapa model dapat didefinisikan oleh manusia dan bukan dipelajari dari data pelatihan.

Efektivitas klasifikasi diukur melalui kemampuannya mengklasifikasikan secara akurat hasil pendekatan pembelajaran mesin dengan apa yang akan dilakukan manusia. Selama masa pengujian algoritma, hal ini dapat dilakukan dengan memeriksa kinerja algoritma pada set pengujian. Jika kumpulan data pelatihan tersedia, kumpulan data tersebut dapat dibagi menjadi kumpulan pelatihan dan kumpulan pengujian, yang kemudian digunakan untuk memeriksa ukuran akurasi.

1.8.2 Pengelompokan

Clustering adalah penerapan teknik AI di mana keputusan output melakukan tugas membagi semua data input menjadi satu atau lebih cluster, di mana setiap cluster berisi input-input yang dekat menurut beberapa definisi kedekatan. Jika kita membandingkan proses pengelompokan dengan klasifikasi, selama tahap pembelajaran dalam siklus hidup, model pengelompokan tidak memerlukan definisi kelas yang sudah ada sebelumnya. Selama fase inferensi, pengelompokan mengambil masukan dan mengidentifikasi klaster yang paling dekat dengan masukan.

Ada banyak jenis algoritme pengelompokan yang berbeda, ada yang melihat kumpulan tetangga yang paling dekat dengan suatu titik, ada pula yang melihat ukuran kepadatan titik untuk menentukan cara mendefinisikan kluster yang berbeda, sementara yang lain mungkin mencoba mengidentifikasi hubungan dan keterkaitan di antara mereka. titik masukan yang berbeda. Terdapat hubungan antara pengelompokan dan klasifikasi, di mana masing-masing kelompok yang diidentifikasi sebagai hasil pelatihan dapat dianggap sebagai definisi kelas yang berbeda. Oleh karena itu, pengelompokan dapat digunakan sebagai alat untuk membantu orang dalam menyiapkan dan mendefinisikan label untuk klasifikasi. Ini selain menggunakannya untuk memetakan atau mengklasifikasikan masukan apa pun ke dalam cluster terdekat.

Algoritme pengelompokan dapat didasarkan pada seberapa dekat tetangga suatu titik masukan, seberapa besar kepadatan titik-titik yang tersebar berdekatan, seberapa cocok titik-titik masukan tersebut dengan asumsi distribusi statistik, atau seberapa baik keterhubungan titik-titik masukan yang berbeda, berdasarkan pada definisi konektivitas.

Salah satu jenis pengelompokan yang spesifik adalah penambangan asosiasi, di mana objek masukan yang memiliki atribut yang sama dikelompokkan bersama. Jika hubungan antara objek masukan yang berbeda dapat direpresentasikan menggunakan hubungan grafis, maka algoritma pengelompokan grafik dapat digunakan untuk penambangan asosiasi. Dalam hubungan grafis, informasi direpresentasikan antara node yang menunjukkan entitas berbeda dan link menghubungkan node bersama-sama. Algoritme pengelompokan grafik menemukan node dalam grafik yang mirip satu sama lain.

Teknik clustering dapat digunakan untuk berbagai jenis operasi bisnis, misalnya. untuk memahami gambar untuk membedakan berbagai jenis jaringan dalam gambar medis, untuk menganalisis lalu lintas jaringan untuk memahami mode perilaku yang berbeda dari perangkat yang berbeda, untuk memodelkan topik yang mungkin menonjol dalam transkrip panggilan

dukungan pelanggan, mengidentifikasi perangkat yang mungkin beroperasi di cara serupa dalam jaringan, dll.

1.8.3 Deteksi Anomali

Dalam banyak jenis operasi bisnis, terdapat konsep operasi normal dan ketika ada sesuatu yang tidak normal, misalnya kondisi abnormal atau anomali mungkin sedang terjadi. Perilaku anomali dapat digunakan untuk mengidentifikasi potensi serangan keamanan yang mungkin dilakukan pada jaringan suatu perusahaan, mesin yang mungkin mengalami situasi bermasalah dan mungkin memerlukan pemeliharaan, pelanggan yang mungkin sangat tidak puas, upaya penipuan, dan beberapa hal lainnya. kasus penggunaan lainnya.

Perilaku anomali dapat dipandang terkait dengan pengelompokan dan klasifikasi. Hal ini dapat dilihat sebagai upaya untuk mengklasifikasikan semua masukan ke dalam dua kelas, normal dan abnormal. Hal ini juga dapat dianggap sebagai pengidentifikasian titik masukan yang tidak mendekati cluster mana pun yang diidentifikasi selama pengelompokan. Namun, deteksi anomali juga dapat dilakukan tanpa menggunakan pendekatan pengelompokan atau klasifikasi apa pun, seperti memeriksa sistem selama pengoperasian, menentukan garis dasar dari nilai yang diukur, dan kemudian melaporkan ketika nilai mulai berada di luar kisaran yang diamati selama garis dasar. Deteksi anomali juga dapat dilakukan dengan membandingkan masukan dengan sekumpulan kelompok, dimana kelompok dipilih berdasarkan beberapa kriteria, dan perilaku anomali ditandai jika masukan tidak sesuai dengan norma yang ditentukan oleh perilaku kelompok tersebut. Deteksi anomali dapat digunakan sebagai bagian dari persiapan data untuk jenis pembelajaran mesin lainnya, di mana data outlier atau anomali dapat dihapus dari set pelatihan.

1.8.4 Pemetaan

Pemetaan adalah transformasi masukan menjadi keluaran dimana masukan dan keluaran tersebut berada pada dua domain berbeda, dimana kedua domain tersebut merupakan dua cara berbeda untuk merepresentasikan entitas yang sama. Sebagai contoh, frase kata dapat dinyatakan sebagai sinyal suara atau representasi tekstual. Terjemahan sampel suara ke representasi tekstual akan menjadi pemetaan. Terjemahan informasi di berbagai representasi, mis. penerjemahan antar bahasa yang berbeda dimana teks yang sama direpresentasikan ke dalam bahasa yang berbeda, merupakan contoh pemetaan.

Dalam banyak kasus, pemetaan dilakukan untuk meningkatkan kecepatan operasi. Menemukan kesamaan di antara rangkaian teks memerlukan perbandingan rumit antara rangkaian huruf yang bisa lambat dan tidak efisien. Namun, jika kata-kata dipetakan ke representasi vektor matematika dalam ruang vektor, perbandingan kemiripan menjadi lebih efisien. Akibatnya, banyak aplikasi pemrosesan teks yang mengandalkan pemetaan kata ke vektor, kalimat ke vektor, paragraf ke vektor, dll. Demikian pula, urutan objek dapat direpresentasikan sebagai vektor, dan rangkaian vektor dapat digunakan untuk menyediakan komposisi otomatis pekerjaan bisnis.

Pemetaan juga dapat digunakan dalam aplikasi manajemen sistem dan jaringan, misalnya. peristiwa yang diamati dalam lalu lintas jaringan atau log sistem dapat dipetakan ke akar penyebab yang mungkin menyebabkan peristiwa tersebut. Teknik berbasis AI untuk

memetakan kejadian hingga akar permasalahan dapat membantu meringankan beban manajemen sistem dan jaringan.

Operasi pemetaan berkaitan dengan klasifikasi, di mana kita dapat melihat masing-masing nilai keluaran dalam representasi yang dipetakan sebagai kelas tempat masukan tersebut dipetakan. Algoritma yang biasanya dikaitkan dengan klasifikasi akan bekerja dengan baik ketika jumlah kelas sedikit. Namun, karena jumlah kelas menjadi sangat besar, algoritma pemetaan mungkin menjadi lebih efisien dan berkinerja lebih baik. Pemetaan digunakan secara luas di banyak proses yang mendukung AI sebagai komponen yang membantu operasi end-to-end dijalankan dengan lebih baik.

1.8.5 Penyaringan

Istilah pemfilteran, yang berasal dari bidang analisis sinyal, namun digunakan dalam banyak proses bisnis, mengacu pada tindakan menghilangkan komponen atau fitur yang tidak diinginkan dari suatu sinyal. Banyak proses bisnis yang didorong oleh sinyal yang dipantau melalui sensor, dan keputusan dibuat dengan menganalisis keluaran dari sensor tersebut. Sinyal dari sensor biasanya diubah menjadi peristiwa yang dimasukkan sebagai masukan ke dalam proses bisnis. Banyak dari peristiwa-peristiwa ini mungkin berlebihan dan terduplikasi, dan pemfilteran menghilangkan atau memampatkan peristiwa-peristiwa tersebut ke subset yang lebih kecil untuk pemrosesan yang efisien.

Filter juga dapat digunakan untuk menghaluskan sinyal dan menghilangkan informasi yang salah. Jika Anda menggunakan sistem navigasi berbasis GPS di mobil Anda, Anda mungkin memperhatikan beberapa saat ketika sistem meyakini bahwa mobil Anda tidak berada di jalan yang tepat, namun berada di titik off-road yang dekat. Hal ini disebabkan kesalahan bawaan dalam memperkirakan posisi menggunakan sistem GPS. Menghaluskan lokasi pada beberapa pengukuran sebelumnya menghilangkan kemungkinan kesalahan tersebut.

Kegunaan lain dari pemfilteran adalah untuk melakukan pengukuran tidak langsung terhadap variabel yang mungkin tidak dapat diamati secara langsung dari data itu sendiri. Salah satu jenis filter, Filter Kalman, menggunakan informasi tentang model sistem, serangkaian kemungkinan pengukuran kebisingan, dan konfigurasi sistem untuk memperkirakan keadaan suatu sistem, dan digunakan secara luas di banyak bidang termasuk pengendalian kendaraan otonom dan drone.

1.8.6 Pemodelan Fungsi

Jenis fungsi AI yang sangat umum adalah tugas memperkirakan fungsi yang memodelkan hubungan antara masukan dan keluaran. Tujuannya adalah untuk menghasilkan fungsi yang dapat memprediksi keluaran berdasarkan masukan.

Dalam banyak kasus, masukan terdiri dari beberapa komponen, misalnya. parameter berbeda yang membentuk atribut kartu kredit atau permohonan pinjaman, dan outputnya adalah keputusan yang dibuat di masa lalu, misalnya. apakah pinjaman itu diperbolehkan atau tidak, dan tugasnya adalah memperkirakan fungsi yang dapat memprediksi keputusan ini berdasarkan nilai masukan. Setelah fungsi ini diperkirakan dari data historis, fungsi ini dapat digunakan untuk mengambil keputusan di masa depan.

Dalam beberapa kasus, masukannya bergantung pada waktu, yaitu terdiri dari serangkaian pengukuran sepanjang waktu. Dalam kasus tersebut, estimasi fungsi dapat mencakup prediksi nilai masa depan dari suatu pengukuran berdasarkan nilai masa lalu dan nilai prediksi dapat digunakan untuk mengambil keputusan. Misalnya, jumlah permintaan yang dilihat di sebuah situs web dapat digunakan untuk memprediksi perkiraan tingkat permintaan di masa depan, dan perkiraan tersebut digunakan untuk mengalokasikan sumber daya, misalnya. jumlah server yang dibutuhkan untuk menangani perkiraan beban kerja. Pemodelan fungsi dapat dipandang sebagai fungsi AI yang paling umum dan sebagian besar fungsi AI lainnya dapat didefinisikan ulang sebagai kasus pemodelan fungsi yang spesifik.

1.8.7 Pencapaian Tujuan

Pencapaian tujuan adalah fungsi khusus dalam AI yang menentukan tujuan sistem, dan sistem diharapkan mencapai tujuan tersebut secara otomatis dengan masukan manusia yang minimal. Baik pembelajaran penguatan maupun pembelajaran Automaton menggunakan mesin keadaan terbatas dirancang secara eksplisit untuk mencapai suatu tujuan, namun model AI lainnya juga dapat digunakan untuk mencapai tujuan yang telah ditentukan.

Salah satu contoh spesifik pencapaian tujuan adalah sistem menang dalam suatu permainan, baik permainan itu Catur, Go, atau Checkers. Untuk mencapai tujuan tersebut, sistem dapat dilatih untuk memodelkan dampak dari berbagai gerakan yang dilakukan oleh dirinya sendiri, lawannya (atau lawannya), dan untuk memilih gerakan terbaik yang memungkinkannya memenangkan permainan.

Contoh lain dari pencapaian tujuan adalah masalah optimasi sistem, dimana tujuan yang diberikan kepada sistem adalah mempertahankan beberapa properti dari sistem sambil memaksimalkan atau meminimalkan beberapa atribut dari sistem. Contohnya adalah meminta pengontrol pusat data untuk meminimalkan konsumsi daya (misalnya dengan menggunakan jumlah server sesedikit mungkin) sambil memastikan bahwa semua aplikasi berjalan dengan ambang batas atas yang diinginkan pada waktu responsnya.

1.9 RINGKASAN

Dalam bab ini, kami telah memberikan gambaran umum tingkat tinggi tentang AI dan penggunaannya dalam proses bisnis. Kami telah memodelkan operasi bisnis dan pengambilan keputusan bisnis serta memperkenalkan siklus belajar-menyimpulkan-tindakan sebagai mekanisme dasar penggunaan AI dalam bisnis. Kami telah memberikan gambaran umum tentang jenis model AI yang digunakan dalam praktiknya, dan taksonomi tentang bagaimana model ini dapat dilatih, dan daftar beberapa fungsi umum AI yang dapat digunakan.

Pada bab berikutnya, kita akan melihat tantangan yang terkait dengan pembuatan model AI yang mendorong perlunya pembelajaran gabungan.

BAB 2
SKENARIO UNTUK AI FEDERASI

Pada Bab 1, kita membahas garis besar operasi bisnis berbasis AI dan membahas bagaimana hal tersebut dapat diimplementasikan menggunakan siklus Learn Infer Act. Dalam bab ini, kami memperluas aspek spasial tentang bagaimana proses Pembelajaran dan inferensi terjadi, yang telah dibahas secara singkat di Bagian 1.7. Sebagian besar situasi dunia nyata dalam bisnis berhubungan dengan sistem yang didistribusikan pada wilayah yang luas dan saling terhubung oleh jaringan komunikasi komputer. Komponen yang berbeda mungkin berlokasi di berbagai belahan dunia yang terhubung melalui Internet, atau berada dalam lokasi berbeda di suatu perusahaan yang terhubung melalui intranet perusahaan. Kami akan mengelompokkan situs-situs ini ke dalam salah satu dari tiga kategori (a) situs tepi (b) situs proxy dan (c) situs pusat.

Definisi pasti dari edge, proxy, dan situs sentral akan bergantung pada spesifikasi bisnis. Misalnya, jika kita mempertimbangkan sebuah pabrik yang mempunyai beberapa pabrik produksi. Masing-masing lokasi pabrik akan menjadi lokasi tepi. Pabrikan mungkin memiliki pusat data yang merupakan situs pusatnya. Tidak ada situs proxy dalam konfigurasi ini, yang terdiri dari beberapa situs tepi dan satu situs pusat.

Jika Anda mempertimbangkan sebuah bank internasional dengan banyak cabang di berbagai negara, setiap cabang bank tersebut adalah situs edge. Cabang-cabang tersebut mungkin mengirimkan data mereka ke pusat data khusus negara yang akan menyimpan data di pusat-pusat tersebut sesuai dengan peraturan data yang diwajibkan untuk situs tersebut. Cabang-cabang ini dapat dianggap sebagai tempat perantara bank. Bank mungkin juga memiliki pusat data internasional tempat mereka melakukan sebagian besar pengembangan dan pemeliharaan perangkat lunak. Pusat data internasional itu adalah situs pusatnya.

Bisnis lain, misalnya sebuah perusahaan asuransi, mungkin telah memutuskan untuk menyelenggarakan seluruh operasinya di penyedia layanan cloud. Cabang-cabang perusahaan asuransi menghasilkan berbagai transaksi yang dibuat oleh agen perusahaan untuk pemrosesan asuransi. Cabang-cabangnya adalah situs tepi sedangkan lokasi perusahaan asuransi yang dihosting di cloud menjadi situs pusatnya.

AI didorong oleh data, dan masing-masing situs ini dapat memiliki lima peran yang mungkin dimainkan dalam pemrosesan data. Sebuah situs dapat memiliki:

a) peran pembuatan data
b) peran pengumpulan data
c) peran pelatihan model
d) peran inferensi dan
e) peran tindakan.

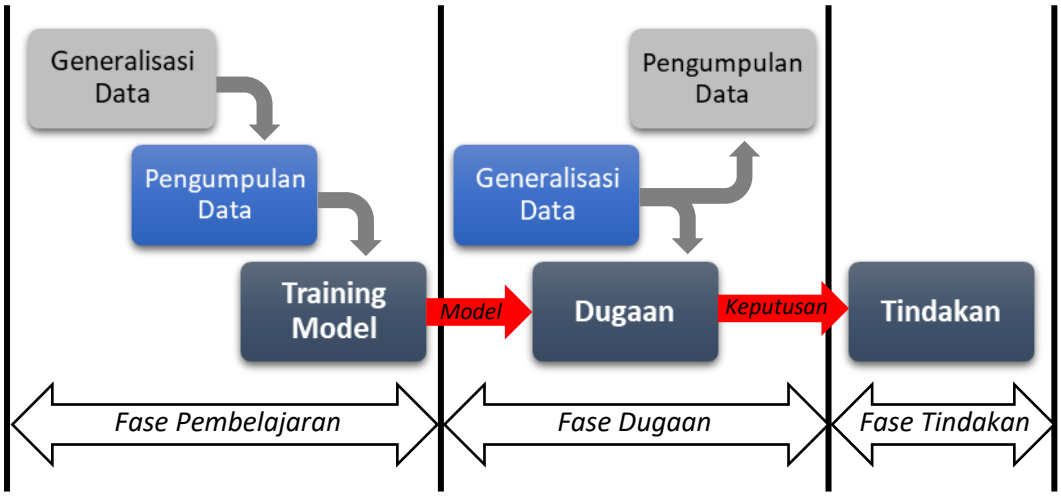
Situs dengan peran pembuatan data menghasilkan data yang diperlukan untuk proses bisnis. Situs dengan peran pengumpulan data memelihara data historis yang dihasilkan dalam perusahaan. Situs dengan peran pelatihan model menggunakan algoritme pembelajaran

mesin untuk mengonversi data menjadi model AI, dan harus memiliki kapasitas komputasi yang memadai serta akselerator perangkat keras yang diperlukan untuk mempercepat tugas pembuatan model. Situs dengan peran Inferensi menggunakan data yang dihasilkan untuk mencapai keputusan dan situs dengan peran tindakan adalah lokasi di mana tindakan yang disarankan diambil.

Selama fase Pelajari siklus Pelajari Infer Act, situs dengan peran pengumpulan data dan peran pelatihan model adalah situs yang aktif. Situs dengan peran pembuatan data mungkin aktif dalam beberapa kasus, jika situs tersebut menghasilkan data pelatihan, dan dalam kasus lain mungkin tidak aktif, misalnya. jika data pelatihan diperoleh dari sumber lain. Situs dalam peran pengumpulan data mengumpulkan data pelatihan dalam jumlah yang memadai, baik secara real-time atau data yang dikumpulkan sebelumnya. Jika situs tidak memiliki peran pelatihan model, data yang dikumpulkan dikirim ke situs dengan peran pelatihan model. Situs tersebut akan memproses data untuk membuat model AI.

Model yang dibuat di situs dengan peran pelatihan model dikirim ke situs dengan peran inferensi, yang diaktifkan selama fase inferensi dari siklus Learn Infer Act. Selama fase inferensi, situs lain yang aktif adalah situs yang berperan dalam pembuatan data. Data yang dibuat di situs dengan situs pembuatan data dikirim ke situs dengan peran inferensi yang akan mengubah data yang dihasilkan menjadi suatu keputusan. Situs dengan peran pengumpulan data mungkin aktif atau tidak dalam fase ini tergantung pada kasus penggunaan. Jika kasus penggunaan mengumpulkan data selama fase inferensi untuk memperbarui model nanti, peran pengumpulan data harus aktif. Namun, jika kasus penggunaannya sedemikian rupa sehingga data inferensi tidak dapat dikumpulkan, situs dengan peran ini mungkin tidak aktif.

Setelah keputusan dibuat, keputusan tersebut dikirim ke situs dengan peran tindakan, yang akan mengambil tindakan. Tentu saja, jika situs inferensi dan situs tindakan sama, aktivitas transfer apa pun dapat dihindari. Perbedaan peran lokasi dan kapan peran ini diaktifkan selama fase siklus yang berbeda ditunjukkan pada Gambar 2.1.



Gambar 2.1 Peran aktif selama fase berbeda dalam siklus Belajar→Dugaan→Tindakan.

Penetapan peran yang berbeda pada situs tepi, situs proksi, dan situs pusat menghasilkan pola yang berbeda untuk AI perusahaan. Dalam beberapa pola AI perusahaan ini, kita perlu menggunakan teknik AI gabungan.

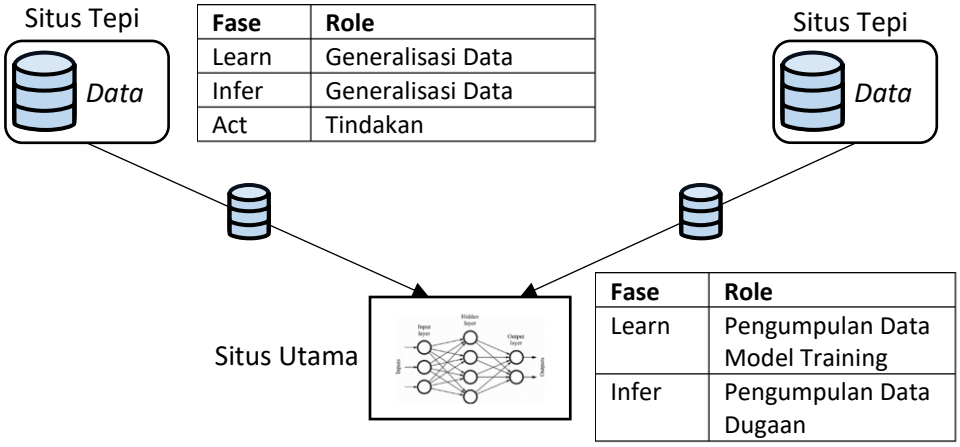
2.1 POLA ABSTRAK UNTUK AI PERUSAHAAN

Setiap pola pembelajaran mesin perusahaan memberikan pemetaan yang berbeda dari lima peran pembuatan data, pengumpulan data, pelatihan model, inferensi, dan tindakan ke situs edge, proxy, dan pusat. Di hampir semua kasus, peran pembuatan dan tindakan data akan berada di lokasi edge. Peran pengumpulan data, pelatihan model, dan inferensi dapat ditugaskan ke salah satu dari tiga lokasi lainnya.

2.1.1 AI terpusat

Pendekatan konvensional untuk pembelajaran mesin di industri adalah pembelajaran mesin terpusat. Dalam pembelajaran terpusat, peran pengumpulan data, pelatihan model, dan inferensi semuanya ditugaskan ke situs pusat. Seperti dalam semua pola, pembuatan data dan peran tindakan ditugaskan ke situs edge. Situs proksi tidak perlu ada dalam pembelajaran terpusat, atau situs proksi mungkin hadir dengan peran tambahan sebagai pengumpul data hanya selama fase pembelajaran dan penyimpulan.

Penggunaan situs pusat untuk melatih model memiliki beberapa keuntungan. Ini akan memungkinkan penggunaan sistem yang memiliki akselerator khusus untuk mempercepat proses pelatihan. Ini menyederhanakan tugas pembaruan model, jika diperlukan di masa depan. Situs pusat kemungkinan besar memiliki kapasitas komputasi yang diperlukan untuk pelatihan model, serta keahlian manusia untuk mengoptimalkan dan membuat model pembelajaran mesin terbaik. Data pelatihan yang dikumpulkan dapat dikurasi dan dibersihkan di situs tersebut.



Fase	Role
Learn	Generalisasi Data
Infer	Generalisasi Data
Act	Tindakan

Fase	Role
Learn	Pengumpulan Data Model Training
Infer	Pengumpulan Data Dugaan

Gambar 2.2 Pola AI terpusat.

Pola abstrak AI terpusat tanpa proxy apa pun ditunjukkan pada Gambar 2.2. Data untuk membuat model pembelajaran mesin dikurasi dari banyak situs edge yang berperan dalam pembuatan data. Dua situs tepi tersebut ditunjukkan dalam gambar. Data yang dihasilkan dari

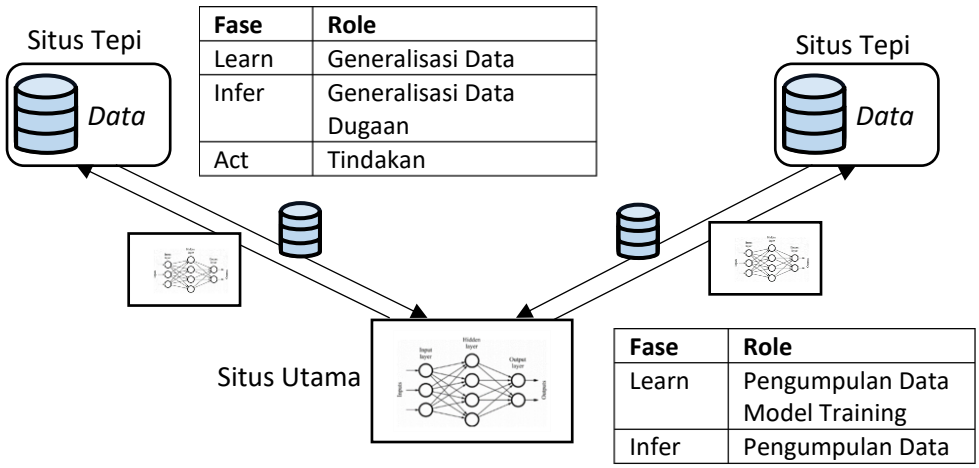
masing-masing situs dikirim ke situs pusat. Situs pusat juga merupakan situs pengumpulan data.

Selama fase pembelajaran, situs edge mengirimkan data yang dihasilkan ke lokasi terpusat untuk membuat model AI. Model ini dibuat, dipelihara dan diperbarui di situs pusat. Selama fase inferensi, data yang dihasilkan di lokasi tepi dikirim ke lokasi pusat di mana model yang telah dilatih sebelumnya digunakan untuk menentukan keputusan yang tepat untuk diambil. Keputusan yang dihasilkan dikirim kembali ke lokasi tepi dimana tindakan diambil. Peran pengumpulan data selama fase infer akan dilakukan oleh situs pusat.

Secara umum, beberapa situs tepi mungkin mengambil peran sebagai penghasil data hanya selama fase pembelajaran, beberapa situs tepi mungkin mengambil peran sebagai penghasil data hanya selama fase inferensi, dan beberapa situs tepi mungkin mengambil peran peran selama kedua fase. Namun, untuk Gambar 2.2, kami berasumsi bahwa semua situs tepi mempunyai peran ini dalam kedua fase. Pada pola ini terjadi perpindahan data dari situs tepi ke situs pusat. Namun, model tersebut tidak pernah berpindah dari lokasi pusat.

2.1.2 Inferensi Tepi

Dalam pola inferensi tepi yang ditunjukkan pada Gambar 2.3, situs pusat berperan sebagai pengumpulan data dan pelatihan model selama fase pembelajaran dalam siklus Learn Infer Act. Namun, model yang dilatih dipindahkan kembali ke lokasi tepi, yang berperan sebagai inferensi selama fase inferensi siklus. Situs proxy mungkin tidak ada, atau jika ada, memiliki peran pengumpulan data selama fase pembelajaran dan penyimpulan.



Fase	Role
Learn	Generalisasi Data
Infer	Generalisasi Data Dugaan
Act	Tindakan

Fase	Role
Learn	Pengumpulan Data Model Training
Infer	Pengumpulan Data

Gambar 2.3 Pola inferensi tepi.

Perbedaan utama dari pola pembelajaran terpusat adalah bahwa model berpindah dari lokasi pusat ke lokasi tepi antara fase pembelajaran dan fase inferensi. Keuntungannya adalah, selama inferensi, latensi jaringan antara situs tepi dan situs pusat dapat dihindari. Hal ini memungkinkan pengambilan keputusan dan tindakan yang lebih responsif selama tahap inferensi. Jika jaringan antara situs tepi dan situs pusat mahal, pendekatan ini dapat menghemat biaya jaringan secara signifikan selama fase inferensi.

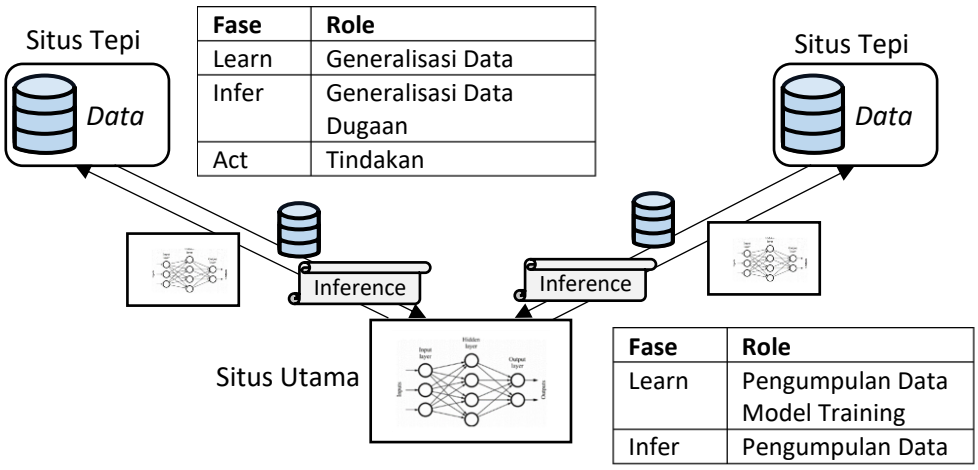
Dalam beberapa kasus penggunaan, koneksi antara situs tepi dan situs pusat mungkin terputus-putus, dan tidak dapat diandalkan selama fase penyimpulan. Dalam kasus tersebut, pola inferensi tepi merupakan pendekatan yang layak, sedangkan pola AI terpusat tidak akan berfungsi. Contohnya adalah pesawat terbang atau kendaraan udara tak berawak yang mungkin perlu beroperasi tanpa koneksi jaringan bandwidth tinggi yang tersedia selama penerbangan. Mereka dapat memuat model dari situs pusat sebelum penerbangan mereka, dan beroperasi secara independen dengan melakukan inferensi menggunakan model lokal selama penerbangan mereka.

Jika model perlu diperbarui, data perlu dikumpulkan dan dikirim kembali ke lokasi pusat. Peran pengumpulan data selama fase infer dapat dilakukan di situs tepi atau situs pusat. Situs tepi mungkin melakukan pengumpulan data pada skala waktu yang lebih lambat dibandingkan pengambilan keputusan yang diperlukan selama tahap inferensi. Sebagai contoh, situs tepi dapat mengumpulkan data yang dihasilkan dan mengunggahnya ke situs pusat di lain waktu, misalnya. selama jam-jam di luar jam sibuk ketika jaringan mungkin tidak terlalu padat atau lebih murah.

2.1.3 Inferensi Tepi Federasi

Pola inferensi tepi gabungan merupakan modifikasi terhadap pola inferensi tepi, dengan perbedaan bahwa beberapa atau semua situs tepi bertukar informasi satu sama lain selama fase inferensi dari siklus Pelajari→Inferensi→Bertindak. Mereka mungkin memilih untuk melakukan hal ini dengan bantuan dari lokasi pusat, dengan asumsi mereka mempunyai konektivitas dengan lokasi pusat, atau beroperasi tanpa bantuan apa pun dari lokasi pusat.

Gambar 2.4 menunjukkan pola ini ketika situs pusat tersedia selama siklus inferensi. Bagian dari peran inferensi ditugaskan ke situs pusat yang membantu situs tepi dalam peran inferensi. Contoh inferensi tepi gabungan dapat dilihat pada pemeriksaan retakan pada jembatan menggunakan drone. Segerombolan drone mungkin akan dikirim untuk memeriksa kondisi jembatan. Mereka mungkin mencari retakan di permukaan, dan jika drone menemukan area yang diduga terdapat retakan, drone mungkin akan meminta drone lain untuk memeriksa permukaannya juga. Jika drone yang berbeda membawa jenis peralatan penginderaan yang berbeda, memeriksa hasil inferensi dari drone yang berbeda dengan menggunakan peralatan penginderaan yang berbeda dapat menghasilkan inferensi keseluruhan yang lebih baik. Drone dapat beroperasi dengan atau tanpa keterlibatan lokasi pusat untuk mengoordinasikan inferensi mereka. Penggunaan situs pusat membuat tugas koordinasi menjadi lebih sederhana, namun juga memerlukan koneksi jaringan yang baik antara drone (edge sites) dan situs pusat.



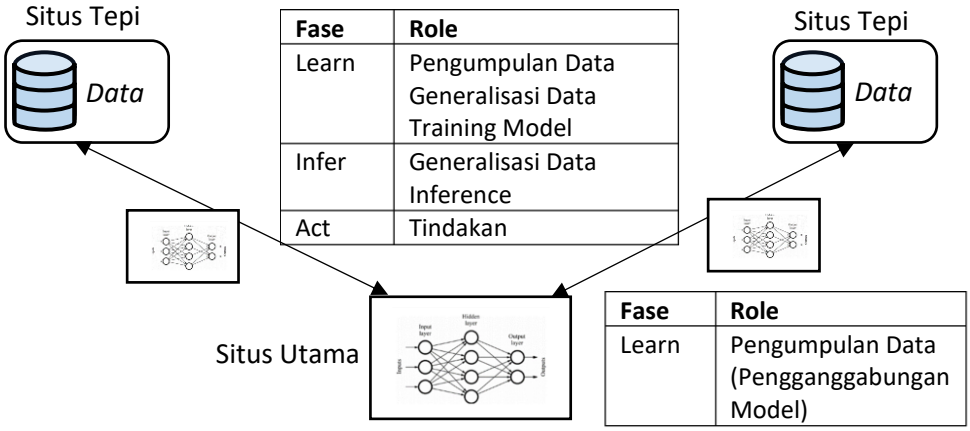
Fase	Role
Learn	Generalisasi Data
Infer	Generalisasi Data Dugaan
Act	Tindakan

Fase	Role
Learn	Pengumpulan Data Model Training
Infer	Pengumpulan Data

Gambar 2.4 Pola inferensi tepi gabungan.

2.1.4 Pembelajaran Tepi

Pola pembelajaran tepi berguna ketika konektivitas jaringan antara situs tepi dan situs pusat tidak terlalu stabil, mahal, atau tidak cukup cepat. Dalam pola ini, situs tepi tidak mengirimkan data apa pun ke situs pusat selama fase siklus Pelajari - Infer - Bertindak - apa pun. Situs tepi mengambil peran pengumpulan data dan pelatihan model selama fase pembelajaran, peran pengumpulan dan inferensi data selama fase inferensi, selain peran biasa dalam pembuatan data selama fase pembelajaran dan inferensi, dan untuk tindakan selama fase tindakan.



Fase	Role
Learn	Pengumpulan Data Generalisasi Data Training Model
Infer	Generalisasi Data Inference
Act	Tindakan

Fase	Role
Learn	Pengumpulan Data (Pengganggabungan Model)

Gambar 2.5 Pola pembelajaran tepi.

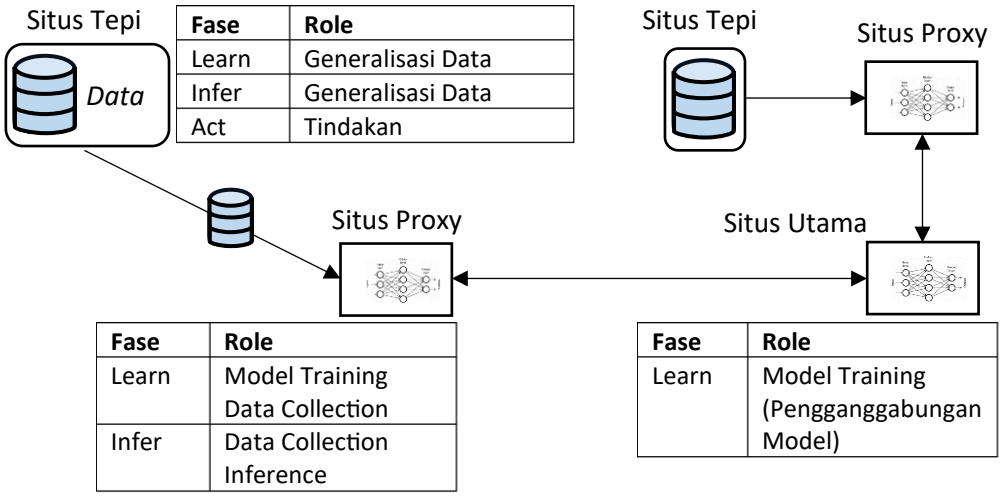
Pola ini ditunjukkan pada Gambar 2.5. Dalam pola ini, situs tepi tidak mengirimkan data mentah ke situs pusat. Sebaliknya, data diubah menjadi model AI di situs edge itu sendiri. Model AI dipindahkan ke situs pusat, di mana model dari situs tepi yang berbeda digabungkan menjadi satu model yang diperoleh dengan menggabungkan semua model menjadi satu. Dalam pola ini, peran situs pusat selama fase pembelajaran adalah membantu penggabungan model yang dihasilkan oleh situs tepi yang berbeda. Situs pusat tidak memiliki peran selama fase lainnya. Edge Learning akan memerlukan daya komputasi yang cukup di setiap situs edge.

Selain itu, algoritme untuk menggabungkan model cenderung lebih kompleks dibandingkan sekadar melatih model berdasarkan data yang tersedia.

2.1.5 Pembelajaran Proksi

Pola pembelajaran proksi berguna ketika situs tepi tidak memiliki kapasitas komputasi yang cukup untuk melatih model, atau mungkin tidak memiliki data yang cukup untuk melatih model AI yang baik. Kualitas model AI sangat bergantung pada ketersediaan data pelatihan yang memadai. Jika pada saat yang sama, situs edge tidak dapat mengirim data ke situs pusat karena alasan apa pun, akan berguna jika memiliki situs proxy yang dapat melakukan pelatihan model untuk situs edge.

Pola ini ditunjukkan pada Gambar 2.6. Peran yang diberikan pada situs yang berbeda juga ditunjukkan pada gambar. Ada dua situs proxy yang ditampilkan, meskipun mungkin ada beberapa situs proxy lainnya. Situs proxy akan melatih model dari data yang mereka kumpulkan, dan bertukar model hanya dengan situs pusat. Situs pusatnya akan menggabungkan model-model ini bersama-sama. Situs-situs tepi secara efektif direduksi menjadi peran yang mereka miliki dalam pola pembelajaran terpusat, sementara sebagian besar peran yang seharusnya mereka ambil dalam pola pembelajaran tepi dilakukan oleh situs-situs proksi.



Gambar 2.6 Pola pembelajaran proxy.

Di awal bab ini, kami telah menjelaskan skenario bank internasional. Setiap cabang bank merupakan situs edge, pusat data dalam negeri adalah situs proksi, dan pusat data internasional adalah situs pusatnya. Jika peraturan membatasi transfer data perbankan ke luar negeri, situs proxy dapat melatih model AI lokal yang spesifik untuk suatu negara. Situs pusat dapat menggabungkan model-model tersebut untuk menciptakan model global yang menggabungkan pola-pola yang ada dalam data di semua negara.

Pola pembelajaran edge dan pola pembelajaran proksi menunjukkan pendekatan pembangunan model yang memerlukan federasi selama fase pembelajaran. Inferensi tepi gabungan menunjukkan pendekatan yang memerlukan federasi selama fase inferensi. Kita dapat menyebut ketiga pola ini sebagai pendekatan umum untuk AI gabungan. Pada bagian

selanjutnya, kita melihat beberapa motivasi mengapa pembelajaran gabungan dan inferensi gabungan mungkin diperlukan.

2.2 MOTIVASI PEMBELAJARAN FEDERASI

Ada banyak faktor yang mendorong perlunya pembelajaran gabungan. Faktor-faktor ini mencakup biaya operasional, kendala jaringan, batasan peraturan, serta masalah privasi dan keamanan data.

2.2.1 Biaya Operasional

Di sebagian besar perusahaan, sejumlah besar data dihasilkan selama operasi normal. Untuk menganalisis dan memproses data ini, mereka perlu memindahkan data ke beberapa lokasi yang memiliki daya komputasi dan penyimpanan yang memadai untuk memproses data tersebut. Pergerakan data, penyimpanan data, dan daya komputasi yang diperlukan untuk memprosesnya menambah biaya pengelolaan dan pengoperasiannya. Dengan kemajuan teknologi penyimpanan dan kapasitas komputasi, biaya pemrosesan data telah turun secara signifikan. Namun, biaya komunikasi jaringan tidak turun sebesar biaya penyimpanan dan komputasi. Hasilnya, solusi yang dapat menghindari perpindahan data seringkali dapat mengurangi biaya operasional.

Mengacu kembali pada Gambar 2.2, situs tepi tipikal adalah lokasi dimana data dihasilkan. Lokasi tepi juga merupakan lokasi di mana tahap tindakan dilakukan. Dibandingkan dengan pola pembelajaran terpusat, pola pembelajaran tepi mempunyai keuntungan yaitu hanya memindahkan model yang dibuat setelah pemrosesan data antara situs tepi dan situs pusat. Karena model biasanya jauh lebih kecil dibandingkan data sebenarnya, biaya bandwidth jaringan untuk pembelajaran gabungan bisa jauh lebih rendah. Manfaat yang sama berlaku untuk pola pembelajaran proxy kecuali pengurangan bandwidth jaringan antara situs proxy dan situs pusat.

Memindahkan jumlah data yang lebih kecil juga memiliki manfaat sampingan yaitu mengurangi jumlah daya pemrosesan yang diperlukan di lokasi pusat. Secara umum, pelatihan model jauh lebih mahal secara komputasi dibandingkan dengan penggabungan model. Pengurangan biaya di lokasi pusat akan diimbangi dengan peningkatan kebutuhan komputasi di lokasi tepi atau situs proksi untuk melatih model.

Dalam banyak situasi, kapasitas komputasi yang diperlukan sudah tersedia di lokasi tepi. Dalam kasus ini, pembelajaran edge (atau pembelajaran proksi) memberikan keuntungan biaya yang jelas. Keuntungan biaya dapat lebih terasa ketika lokasi pusat mempunyai biaya yang terkait dengan daya pemrosesan tambahan, misalnya. dalam komputasi awan, biaya dibayarkan sesuai dengan jumlah sumber daya yang digunakan. Jika suatu perusahaan menggunakan sumber daya cloud dan memiliki akses ke komputasi di situs edge/proxy, pembelajaran edge atau pembelajaran proxy dapat menghemat biaya secara signifikan.

Namun, jika kapasitas baru perlu ditempatkan di lokasi edge atau proxy, biaya penyediaan infrastruktur tersebut perlu diimbangi dengan penghematan biaya operasional di lokasi pusat. Dalam kasus tersebut, pembelajaran edge/pembelajaran proxy belum tentu menghemat biaya operasional. Jawaban akhirnya akan bergantung pada biaya relatif

komputasi, konektivitas, dan penyimpanan. Namun, dalam banyak kasus, pendekatan pembelajaran gabungan ternyata merupakan pendekatan yang lebih murah untuk membuat dan melatih model AI.

2.2.2 Kendala Jaringan

Dalam banyak kasus, suatu perusahaan mungkin ingin menggunakan AI gabungan karena konektivitas jaringan antara situs tepi dan situs pusat terbatas. Batasan tersebut dapat berupa bandwidth terbatas, latensi tinggi, atau keandalan rendah. Dengan konektivitas jaringan yang terbatas, pemindahan data mentah dari situs pembuatan data ke situs lain mungkin tidak dapat dilakukan. Pendekatan yang lebih baik, dan dalam beberapa kasus, satu-satunya pendekatan yang layak, adalah memproses data secara lokal di lokasi tepi.

Situasi ini sangat umum terjadi ketika situs tepi berada di lokasi terpencil, misalnya. di industri kehutanan atau di pertambangan atau rig lepas pantai. Dalam industri kehutanan, edge site adalah sebuah komputer kokoh yang dipasang pada kendaraan penebangan kayu yang beroperasi di lokasi terpencil. Rig lepas pantai sering kali hanya memiliki konektivitas melalui jaringan satelit. Tautan satelit memiliki latensi yang relatif tinggi, yaitu sekitar 500 ms waktu pulang pergi dibandingkan dengan 150 ms untuk jaringan seluler 4G atau 40 ms untuk jaringan kabel tergantung pada lokasinya. Bandwidth yang tersedia pada sambungan satelit juga jauh lebih kecil, biasanya tersedia sekitar 1 Mbps per detik, berbeda dengan infrastruktur kabel yang dapat dengan mudah memberikan kecepatan 1 Gbps dengan biaya yang sangat rendah.

Lingkungan lain dengan jaringan yang sangat terbatas adalah industri perjalanan dan transportasi. Kapal-kapal di laut dapat memiliki kapasitas komputasi di dalamnya, namun konektivitas mereka ke seluruh dunia sangat terbatas. Pesawat terbang, ketika sedang terbang, juga mempunyai konektivitas yang sangat terbatas dengan infrastruktur darat, meskipun konektivitasnya jauh lebih baik ketika berada di bandara.

Dalam operasi militer, konektivitas jaringan selalu dibatasi. Hal ini dikarenakan sebagian besar operasi militer terjadi di wilayah yang terpencil, memiliki infrastruktur jaringan komunikasi yang terbatas, dan infrastruktur komunikasi yang tersedia selalu terancam terganggu akibat tindakan musuh. Militer modern juga terdiri dari kapal, pesawat, helikopter, dan drone, yang semuanya memiliki konektivitas jaringan yang sangat terbatas saat beroperasi.

Kendala jaringan dapat menjadi masalah bahkan pada jaringan kabel. Jaringan seluler, misalnya, memiliki infrastruktur luar biasa untuk memindahkan data. Namun, karena luasnya wilayah geografis yang harus dilayani oleh operator jaringan seluler, penyedia jaringan seluler di negara-negara besar mengoperasikan selusin atau lebih pusat data untuk menyimpan catatan panggilan dan koneksi pelanggan mereka. Operator jaringan seluler di Amerika Serikat akan memiliki selusin atau lebih pusat data. Di India, jaringan seluler dibagi menjadi dua puluh dua lingkaran, dan biasanya operator jaringan seluler memiliki pusat data khusus untuk setiap lingkaran. Volume catatan yang dihasilkan setiap hari di masing-masing pusat data ini sangat besar, sehingga pengumpulan semua data di satu situs memakan waktu lama, meskipun terdapat tautan jaringan kabel yang bagus.

Ketika perpindahan data menjadi masalah, pola yang mendukung AI gabungan dapat memberikan keuntungan yang signifikan dibandingkan pola pembelajaran terpusat.

2.2.3 Peraturan Privasi Data

Dalam beberapa kasus, peraturan dapat mencegah perpindahan data dari situs tepi ke situs pusat. Hal ini sering terjadi pada industri dengan tingkat regulasi yang tinggi, seperti layanan kesehatan atau keuangan.

Sebagai contoh, jaringan rumah sakit mungkin mengoperasikan beberapa klinik tempat pasien mengunjungi dokternya, dan mungkin juga memiliki pusat penelitian tempat para ilmuwan peneliti mencoba menganalisis pola penyebaran penyakit atau mempelajari efektivitas berbagai jenis pengobatan. Undang-undang privasi pasien di banyak negara mungkin melarang pembagian data mentah pasien kepada ilmuwan penelitian. Meskipun para profesional layanan kesehatan di klinik mungkin dapat menyimpan data pasien mereka dan melihatnya tanpa batasan, para ilmuwan peneliti mungkin tidak diizinkan untuk melihat data ini. Jika para ilmuwan penelitian ingin membangun model yang didasarkan pada data yang ada di berbagai lokasi klinis, pembelajaran gabungan mungkin dapat memberikan mereka pendekatan yang layak.

Pembatasan serupa terhadap pembagian informasi juga terjadi pada lembaga pemerintah. Lembaga yang berbeda mempunyai peraturan yang berbeda, dan mereka mungkin tidak diperbolehkan berbagi data secara bebas satu sama lain. Sebagai contoh, catatan pajak di Amerika Serikat hanya dapat dibagikan kepada sejumlah lembaga pemerintah tertentu. Demikian pula, beberapa negara bagian di Amerika Serikat mungkin melarang akses terhadap beberapa informasi, seperti catatan mengemudi, ke beberapa lembaga federal. Peraturan-peraturan ini harus dipatuhi, namun bisa menjadi penghalang ketika berbagai lembaga perlu bekerja sama untuk melakukan kegiatan yang mempunyai kepentingan bersama.

2.2.4 Pertimbangan Keamanan dan Kepercayaan

Dalam beberapa kasus, pertimbangan keamanan dapat mencegah pergerakan data dari situs tepi dan situs pusat. Situasi ini sering muncul ketika lokasi yang berbeda dimiliki oleh organisasi yang berbeda. Dalam lingkungan komputasi saat ini, penggunaan teknologi komputasi awan sangat populer sebagai pendekatan yang hemat biaya dan elastis untuk memperoleh berbagai layanan analisis data. Namun, hal ini memerlukan penyerahan kendali kepada infrastruktur yang dikelola oleh organisasi lain, yaitu penyedia layanan cloud. Dalam hal ini, beberapa data yang dianggap sensitif banyak yang tidak dikirim ke situs cloud, dan perlu diproses secara lokal.

Situasi umum lainnya di mana lokasi yang berbeda mungkin dimiliki oleh organisasi atau organisasi yang berbeda terjadi dalam lingkungan konsorsium, di mana banyak organisasi berencana untuk bekerja sama dan berkolaborasi dalam tugas yang sama. Namun, pembagian data di antara seluruh anggota konsorsium mungkin tidak sepenuhnya terbuka, dan organisasi mungkin khawatir tentang keamanan beberapa bagian data mereka. Dalam kasus ini, anggota bersedia membagikan sebagian datanya kepada orang lain, namun tidak membagikan seluruh datanya.

Secara umum, berbagai situs bersedia berbagi informasi satu sama lain karena mereka memiliki tingkat kepercayaan satu sama lain, dan mereka percaya bahwa berbagi informasi akan saling membantu. Namun, kepercayaan tersebut mungkin tidak mutlak, dan dalam kasus tersebut, mengubah data sehingga tidak sepenuhnya terlihat oleh pihak lain, misalnya. dengan mengubahnya menjadi model, akan memberikan pendekatan yang lebih baik. Pembelajaran gabungan memungkinkan pihak-pihak yang memiliki kepercayaan terbatas satu sama lain untuk berbagi wawasan yang diperoleh dari analisis data mereka.

2.3 PEMBELAJARAN FEDERASI KONSUMEN DAN PERUSAHAAN

Meskipun pembelajaran gabungan dapat berguna dalam banyak konteks yang berbeda, masuk akal untuk membedakan antara dua kategori pembelajaran gabungan, yang masing-masing perlu mengatasi serangkaian masalah dan tantangan yang sangat berbeda. Kami mendefinisikan dua kategori ini sebagai pembelajaran gabungan konsumen dan pembelajaran gabungan perusahaan.

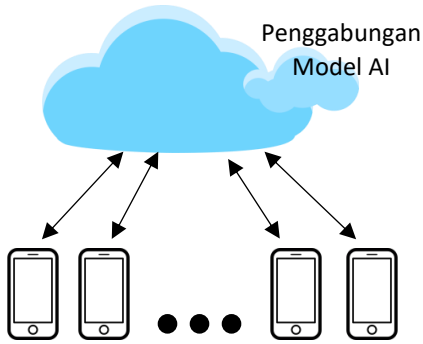
2.3.1 Pembelajaran Federasi Konsumen

Pembelajaran gabungan konsumen mengacu pada situasi ketika pembuatan data terjadi pada perangkat yang dimiliki oleh individu dan bukan milik organisasi. Contoh paling umum dari perangkat tersebut adalah ponsel pintar. Ponsel pintar berisi sejumlah besar data yang sensitif, seperti lokasi dan riwayat lokasi pengguna, informasi pribadi seperti ulang tahun dan anggota keluarga, riwayat penjelajahan web, riwayat jejaring sosial, dll. Informasi ini dapat sangat berguna bagi berbagai jenis pengembang aplikasi seluler, yang dapat menggali informasi untuk meningkatkan layanan mereka. Layanan yang ditingkatkan ini dapat meningkatkan akurasi dalam memprediksi apa yang kemungkinan besar akan diketik pengguna di penelusuran web mereka, jenis film apa yang ingin mereka tonton berdasarkan riwayat mereka, atau jenis rekomendasi produk apa yang harus mereka dapatkan.

Konfigurasi tipikal untuk pembelajaran gabungan konsumen ditunjukkan pada Gambar 2.7. Konfigurasinya terdiri dari situs tempat terjadinya fusi model AI, dan beberapa ponsel pintar yang pada dasarnya merupakan situs edge. Aplikasi seluler di ponsel pintar akan menghasilkan data dari konsumsi lokalnya. Semua ponsel pintar biasanya menjalankan aplikasi yang sama yang seharusnya menghasilkan data dalam format yang konsisten dan umum yang dirancang dan diterapkan oleh pengembang aplikasi. Untuk aplikasi ponsel pintar populer mana pun, jumlah ponsel pintar yang menjalankan aplikasi tersebut bisa mencapai ribuan atau jutaan.

Dalam lingkungan saat ini, merupakan praktik umum bagi banyak mesin pencari, situs jejaring sosial, dan pengecer untuk mengumpulkan data ini di situs pusat dan menggalinya untuk mendapatkan pola umum di seluruh pengguna. Namun, terdapat juga rasa cemas terhadap pelanggaran privasi yang terjadi ketika informasi sensitif dikumpulkan dan dianalisis oleh penyedia layanan. Hasilnya, terdapat dorongan untuk menciptakan teknologi yang memungkinkan eksplorasi pola-pola umum di berbagai pengguna tanpa perlu memindahkan data mereka ke lokasi terpusat. Teknik pembelajaran gabungan memberikan pendekatan untuk melakukan tugas ini.

Dengan pendekatan pembelajaran gabungan, perangkat konsumen tidak perlu memindahkan datanya ke lokasi pusat. Sebaliknya, model-model tersebut akan dibuat pada perangkat lokalnya, dan model-model ini digabungkan di situs data back-end penyedia layanan. Pembelajaran gabungan konsumen ditandai dengan adanya sejumlah besar situs penghasil data (atau lebih tepatnya perangkat), format umum untuk data yang dianalisis ditentukan oleh pengembang aplikasi seluler yang menyertakan teknik pembelajaran gabungan, dan pengoperasian pada telepon pintar.



Gambar 2.7 Pembelajaran gabungan konsumen.

Namun, untuk membangun layanan semacam ini, pendekatan pembelajaran gabungan perlu mengatasi beberapa masalah mendasar. Masalah-masalah tersebut antara lain:

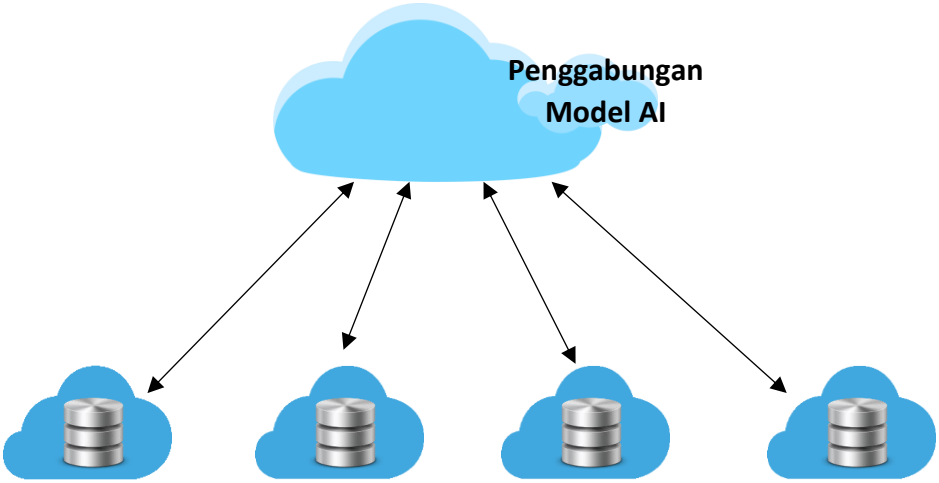
- ⊗ ***Ukuran Data Kecil:*** Penyedia layanan biasanya menyediakan layanannya kepada jutaan konsumen, atau setidaknya beberapa ribu konsumen. Dalam kasus ini, keseluruhan data yang akan ditambang untuk pola dipecah menjadi beberapa unit dengan ukuran kecil. Secara umum, algoritma pembelajaran model AI kurang baik dalam mengekstraksi pola dari data yang berukuran kecil. Akibatnya, model yang dibangun di setiap situs kemungkinan besar tidak mewakili perilaku pengguna secara keseluruhan. Masalahnya kemudian adalah menggabungkan sejumlah besar model yang berbeda, yang masing-masing modelnya sangat tidak akurat, menjadi sebuah model gabungan.
- ⊗ ***Pihak Berbahaya***: Dengan ribuan dan jutaan pengguna, ada kemungkinan bahwa beberapa pengguna akan bertindak jahat, baik sebagai lelucon atau dengan tujuan yang lebih jahat. Meskipun kita dapat berasumsi bahwa sebagian besar orang akan menggunakan perangkat dan data mereka secara normal, tidak semua orang dapat diasumsikan bertindak jujur. Ada beberapa insiden penting di mana orang memasukkan data berbahaya untuk membuat sistem berbasis AI berperilaku tidak menentu. Misalnya, chat-bot berkemampuan AI dilatih untuk berperilaku menjengkelkan dengan memberikan data buruk dalam eksperimen crowdsourcing, algoritme visi komputer dapat ditipu dengan manipulasi gambar masukan, dan serangan serupa dapat diluncurkan pada sistem analisis ucapan. Meskipun serangan ini tidak ditujukan secara eksklusif pada pengaturan pembelajaran gabungan, lingkungan pembelajaran gabungan lebih rentan terhadap serangan tersebut ketika sekelompok pengguna mencoba memanipulasi proses pelatihan mesin masing-masing.

⊗ ***Daya Komputasi yang Terbatas:*** Karena pelatihan model merupakan proses komputasi yang intensif, pelatihan model pada ponsel mungkin mempunyai efek samping yang tidak diinginkan seperti menguras baterai, atau menghabiskan terlalu banyak kapasitas komputasi yang tersedia. Meskipun ketersediaan data terbatas, persyaratan untuk tidak mengganggu pengoperasian normal ponsel juga dapat membatasi pilihan model AI yang dapat dilakukan secara praktis pada ponsel pintar.

Karena tantangan-tantangan ini, pembelajaran gabungan pada perangkat konsumen belum menunjukkan serapan yang signifikan dalam penerapannya pada aplikasi konsumen mana pun yang digunakan secara luas. Sejumlah besar peneliti telah berupaya untuk mengatasi masalah yang terkait dengan pembelajaran gabungan tingkat konsumen, sehingga mereka mungkin masih menemukan beberapa penerapan dalam kasus penggunaan di dunia nyata. Namun, pembelajaran gabungan di tingkat perusahaan, yang akan kami definisikan selanjutnya, kemungkinan besar merupakan bidang yang pertama kali melihat penerapan praktis dari teknologi pembelajaran gabungan.

Pembelajaran gabungan perusahaan mengacu pada situasi di mana situs pembuatan data dimiliki oleh suatu perusahaan. Situs-situs ini umumnya terjadi di kantor, pabrik, atau lingkungan lain di mana data dihasilkan oleh banyak perangkat berbeda, dan dikumpulkan serta disimpan ke dalam beberapa jenis database di lokasi pembuatan data. Pengaturan umum pembelajaran gabungan perusahaan ditunjukkan pada Gambar 2.8 Ini menunjukkan beberapa situs, masing-masing dengan sejumlah besar data yang disimpan dalam database, yang semuanya perlu ditambang untuk membuat model AI secara keseluruhan. Setiap situs dapat berupa bangunan individual suatu perusahaan, atau dapat berupa pusat data, atau partisi logis data lainnya yang dikelola dalam suatu perusahaan, seperti data lake, atau gudang data. Berbagai istilah ini merujuk pada bagian infrastruktur TI perusahaan yang bertanggung jawab terutama untuk menyimpan dan memelihara data yang dihasilkan dalam operasi bisnis normal.

Berbeda dengan pembelajaran gabungan konsumen, yang terdiri dari jutaan perangkat, jumlah situs dalam pembelajaran gabungan perusahaan akan jauh lebih kecil, biasanya berkisar dalam satu digit hingga beberapa puluh lokasi. Masing-masing situs ini biasanya memiliki sejumlah besar data untuk melatih dan mempelajari modelnya. Kapasitas komputasi juga tidak menjadi masalah karena situs biasanya memiliki server dengan kapasitas yang memadai untuk melatih model. Perhatikan bahwa situs fusi itu sendiri mungkin berlokasi bersama dengan salah satu situs yang berisi data lokal. Lokasi bersama ini akan menjadi konfigurasi umum jika semua model AI dilatih di pusat data yang dimiliki oleh suatu perusahaan. Dalam beberapa kasus, lokasi fusi mungkin berlokasi di lokasi fisik yang berbeda dan mungkin tidak ada data lokalnya. Konfigurasi tersebut akan menjadi tipikal jika layanan yang dihosting di cloud digunakan untuk membuat model AI gabungan.



Gambar 2.8 Pembelajaran gabungan perusahaan.

2.3.2 Pembelajaran Federasi Perusahaan

Pembelajaran gabungan perusahaan mempunyai serangkaian tantangan uniknya sendiri. Ini termasuk:

- ***Format Data Heterogen:*** Karena sumber data perusahaan mungkin dikumpulkan secara independen, data mungkin dikumpulkan dalam berbagai format berbeda, meskipun semuanya mengacu pada jenis data yang sama. Misalkan data mengacu pada catatan personel Sumber Daya Manusia, dan dikelola oleh perusahaan yang mengakuisisi perusahaan lain. Catatan dapat disimpan dalam format yang berbeda, meskipun mungkin mengacu pada jenis data yang sama.
- ***Memvariasikan Kualitas Data:*** Lokasi yang berbeda mungkin mempunyai praktik pencatatan dan penyimpanan data yang berbeda. Akibatnya, data yang disimpan di satu situs mungkin dikurasi dengan baik dan memiliki kualitas yang sangat tinggi, sementara data yang disimpan di situs lain mungkin memiliki banyak kesalahan atau ada bagian catatan yang hilang. Terkadang, perbedaan kualitas data mungkin hanya disebabkan oleh perbedaan format penyimpanan informasi, misalnya. suatu lokasi mungkin menyimpan informasi seperti audio dan video dalam format yang hanya mendukung konten beresolusi rendah, sementara lokasi lain mungkin menyimpan informasi dalam konten beresolusi tinggi.
- ***Perbedaan Pola:*** Data di setiap situs mengkodekan beberapa pola, yaitu hubungan implisit dalam data yang dapat dipikirkan oleh algoritma AI atau pembelajaran mesin. Namun, pola yang terdapat di situs berbeda bisa sangat berbeda. Sebagai contoh, situs yang berisi transaksi perbankan untuk pantai Timur Amerika Serikat mungkin menunjukkan beberapa pola aktivitas perbankan yang mungkin berbeda dengan pola aktivitas perbankan di bagian tengah negara tersebut. Saat menggabungkan data dari kedua situs, kehati-hatian harus diberikan agar pola satu situs tidak digabungkan secara tidak tepat. Jika data di pantai timur terutama mengacu pada transaksi yang dilakukan di lingkungan perkotaan sedangkan di negara bagian tengah memiliki

transaksi di lingkungan pedesaan, penggabungan transaksi tersebut dapat menyebabkan pola yang menyesatkan.

- ***Perbedaan Kepercayaan:*** Di banyak lingkungan perusahaan, kepercayaan di antara berbagai organisasi mungkin tidak lengkap. Hal ini dapat mengakibatkan pembatasan pada cara data dipindahkan ke seluruh situs. Perbedaan kepercayaan mungkin disebabkan oleh peraturan yang berlaku di industri, misalnya peraturan perundang-undangan. peraturan privasi pasien mungkin membatasi jenis data yang dapat dibagikan antara departemen rawat jalan seperti rumah sakit dari organisasi layanan kesehatan, dan departemen penelitiannya yang mencoba mengeksplorasi studi tentang dampak pengobatan terhadap pasien. Pembatasan kepercayaan serupa mungkin terjadi di berbagai lembaga di suatu entitas pemerintah. Dalam konteks lain, suatu perusahaan mungkin merupakan konsorsium longgar yang terdiri dari berbagai organisasi, dan hubungan saling percaya di antara organisasi-organisasi ini mungkin tidak bersifat mutlak.

Perbedaan asumsi pengoperasian antara dua jenis pembelajaran gabungan ini dirangkum dalam Tabel 2.1. Jumlah situs dalam pembelajaran gabungan konsumen mempunyai urutan besaran yang berbeda dengan pembelajaran gabungan perusahaan. Akibatnya, setiap situs dalam pembelajaran gabungan konsumen memiliki sebagian kecil data dan volume yang relatif kecil, sedangkan setiap situs dalam pembelajaran gabungan perusahaan akan memiliki jumlah data yang jauh lebih besar. Perangkat yang umum digunakan untuk pembelajaran gabungan konsumen adalah telepon pintar yang kemampuan komputasi dan kemampuan penyimpanannya jauh lebih rendah dibandingkan perangkat umum untuk pembelajaran gabungan perusahaan, yang berupa server komputer. Perangkat berbahaya tidak dapat dikesampingkan dalam kasus konsumen, namun pengaturan bisnis akan membangun kepercayaan yang terbatas di antara situs-situs dalam pembelajaran gabungan perusahaan, yang juga akan mencegah perilaku jahat. Format datanya akan sama dalam pembelajaran gabungan konsumen, namun mungkin berbeda dalam pembelajaran gabungan perusahaan.

Seperti dapat dilihat pada Tabel 2.1, tantangan pembelajaran gabungan perusahaan dan pembelajaran gabungan konsumen sangatlah berbeda. Fokus buku ini adalah pada pembelajaran gabungan perusahaan, yang muncul dalam berbagai skenario berbeda, seperti yang dijelaskan dalam beberapa bagian berikutnya.

Tabel 2.1 Perbandingan pembelajaran gabungan konsumen dan perusahaan.

Aspek	Pembelajaran Federasi Konsumen	Pembelajaran Federasi Perusahaan
Jumlah Situs	ribuan-juta	di bawah seratus
Perangkat Khas	Telepon genggam	Server Kelas Atas
Hubungan Kepercayaan	Tidak terpercaya	Kepercayaan Terbatas
Perilaku Berbahaya	Mengharapkan	Tidak terduga
Volume Data di setiap Situs	Kecil	Besar
Format data	Konsisten	Tidak konsisten

2.4 SKENARIO PEMBELAJARAN FEDERASI PERUSAHAAN

Ada banyak situasi di mana suatu perusahaan mungkin ingin menerapkan kekuatan pembelajaran gabungan dan inferensi gabungan. Situasi ini akan didorong oleh motivasi yang dijelaskan dalam Bagian 2.2.

Setiap kali ada bisnis terdistribusi dengan pembatasan pergerakan data, skenario untuk pembelajaran gabungan dibuat. Di Bagian 2.2, kami menyinggung beberapa skenario tersebut, yang mencakup namun tidak terbatas pada:

- *Kehutanan dan Penebangan Kayu:* dimana banyak aset yang didistribusikan berlokasi di daerah terpencil dengan komputer yang terpasang pada peralatan luar ruangnya, namun dengan konektivitas jaringan yang terbatas dan mahal.
- *Perusahaan Multinasional:* yang memiliki pusat data di banyak negara berbeda, namun memiliki batasan dalam memindahkan data melintasi batas negara.
- *Pabrik Manufaktur:* yang mungkin memiliki banyak fasilitas manufaktur, dan memindahkan data dari fasilitas manufaktur ke lokasi pusat memerlukan biaya yang mahal.
- *Toko Ritel:* yang banyak memiliki gerai perbelanjaan dengan peralatan komputasi di bagian belakang toko ritel, dan ingin terus melanjutkannya tanpa terlalu bergantung pada konektivitas jaringan.

Skenario tambahan di mana AI gabungan dapat digunakan dengan sukses melibatkan situasi di mana perangkat cerdas bergerak dan mobilitas membuat jaringan komunikasi menjadi lemah di beberapa lokasi. Hal ini termasuk mobil cerdas dengan komputer terpasang, drone atau kendaraan udara tak berawak (UAV), kapal dan pesawat terbang, truk yang bergerak di jalan raya, dll. Skenario ini akan muncul baik di ranah militer, di mana pesawa militer, kapal laut, dan aset tak berawak berada. seluler, serta sipil (domain non-militer). Masing-masing skenario ini memiliki sumber daya komputasi yang memadai di lokasi tepi untuk melatih model, dan beberapa motivasi untuk tidak memindahkan data tersebut ke lokasi pusat. Di subbagian berikut, kami menjelaskan beberapa skenario tambahan.

2.4.1 Anak Perusahaan dan Waralaba

Banyak perusahaan diatur sebagai konglomerasi lepas dari anak perusahaan. Ini adalah pengaturan yang cukup umum dilakukan oleh perusahaan multinasional. Umumnya, perusahaan-perusahaan tersebut dapat diatur sebagai satu perusahaan di setiap negara, sesuai dengan peraturan di negara tempat mereka beroperasi. Kepemilikan penuh, atau dalam beberapa kasus, sebagian kepemilikan anak perusahaan di negara mana pun selain negara asal perusahaan tersebut. perusahaan dipegang oleh perusahaan di negara asal. Meskipun anak perusahaan tersebut mungkin dimiliki sepenuhnya oleh perusahaan induk, namun anak perusahaan tersebut mungkin masih tidak dapat mengambil data dari anak perusahaan tersebut karena peraturan yang berlaku.

Skenario ini sangat umum terjadi pada perusahaan yang mungkin memiliki negara induk di Amerika Serikat dan mungkin memiliki anak perusahaan di Uni Eropa. Secara umum, Eropa memiliki peraturan yang lebih ketat mengenai cara penyimpanan dan penanganan data pribadi dibandingkan Amerika Serikat. Akibatnya, pergerakan data mentah tentang klien

perusahaan di Eropa ke Amerika mungkin dibatasi. Dalam kasus tersebut, jika perusahaan induk ingin membangun model yang mengkarakterisasi pola perilaku kliennya di Eropa, maka perusahaan induk tidak akan dapat membangun model tersebut dengan memindahkan data ke pusat data di Amerika Serikat. Namun, jika suatu model dilatih secara lokal, dan hanya model tersebut yang dipindahkan ke Amerika Serikat, model tersebut mungkin dapat memenuhi persyaratan peraturan di negara setempat.

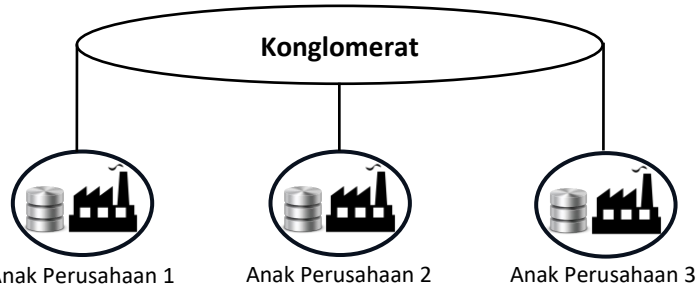
Bentuk umum anak perusahaan adalah seperti yang ditunjukkan pada Gambar 2.9. Konglomerat yang lebih besar akan memiliki kepentingan di banyak anak perusahaan. Setiap anak perusahaan akan bertanggung jawab atas operasinya sendiri, dan akan menghasilkan datanya sendiri. Setiap anak perusahaan perlu menjalankan operasi TI sendiri. Biasanya konglomerat juga memiliki pusat operasi TI. Beberapa fungsi TI mungkin dikendalikan secara terpusat dan dilaksanakan oleh konglomerat, sementara beberapa fungsi TI mungkin dikendalikan secara independen oleh anak perusahaan.

Di beberapa lingkungan, format data dapat diasumsikan sama untuk semua situs yang berbeda. Mereka akan ditentukan dan ditentukan oleh pusat operasi TI konglomerat tersebut. Namun, volume data yang dikumpulkan untuk masing-masing lokasi mungkin sangat berbeda. Tidak semua anak perusahaan memiliki volume transaksi yang sama. Perbedaan volume data ini perlu diperhitungkan ketika melakukan tugas pembelajaran gabungan. Seseorang ingin mengekstrak pola yang lebih jelas dan lebih mudah diekstraksi ketika volume datanya besar. Pada saat yang sama, jika ada pola unik dalam data anak perusahaan dengan volume data yang lebih kecil, akan berguna untuk mengekstrak pola tersebut, dan tidak membiarkan pola tersebut tenggelam dalam volume anak perusahaan dengan jumlah data yang lebih besar.

Karakteristik data yang tersedia di berbagai lokasi tambahan mungkin sangat bervariasi, misalnya. satu anak perusahaan mungkin memiliki sebagian besar klien profesional muda, sementara anak perusahaan lainnya mungkin memiliki sebagian besar pensiunan tua sebagai klien. Ketika lokasi anak perusahaan berada di lokasi internasional, pelanggan dan operasi mereka mungkin sangat dipengaruhi oleh demografi populasi yang mereka layani. Oleh karena itu, setiap lokasi mungkin mengumpulkan data yang berisi pola-pola tertentu yang mungkin dimiliki bersama dan serupa dengan anak perusahaan lainnya, sementara mereka mungkin juga mempunyai beberapa pola dalam datanya yang unik sesuai dengan lingkungan dan konteksnya.

Penerima waralaba juga terlihat sangat mirip dengan anak perusahaan, hanya saja waralaba biasanya berukuran lebih kecil dan lebih mandiri dalam operasinya. Karena skala operasinya yang kecil, waralaba cenderung lebih mengandalkan layanan TI yang disediakan oleh personel TI konglomerat. Namun, beberapa waralaba mungkin memilih untuk memiliki operasional TI sendiri, khususnya jika mereka ingin memiliki catatan independen dan tidak bergantung pada konglomerat untuk sebagian operasional TI mereka. Ketika waralaba beroperasi dengan departemen TI mereka sendiri, mereka mungkin menyimpang dari format dan konvensi yang digunakan oleh kantor TI pusat konglomerat, serta fungsinya. Waralaba yang menangani operasinya sendiri secara independen mungkin ingin melindungi data bisnisnya dengan merahasiakannya. Kebutuhan akan kemandirian ini lebih mungkin terlihat

ketika pewaralaba mengoperasikan sejumlah kantor di suatu wilayah. Sebagai contoh, konglomerat mungkin merupakan jaringan makanan cepat saji, dan pewaralaba mungkin mengoperasikan beberapa toko di wilayah geografis tertentu berdasarkan lisensi dan perjanjian bisnis dengan konglomerat.



Gambar 2.9 Pendirian anak perusahaan dan waralaba.

2.4.2 Merger dan Akuisisi

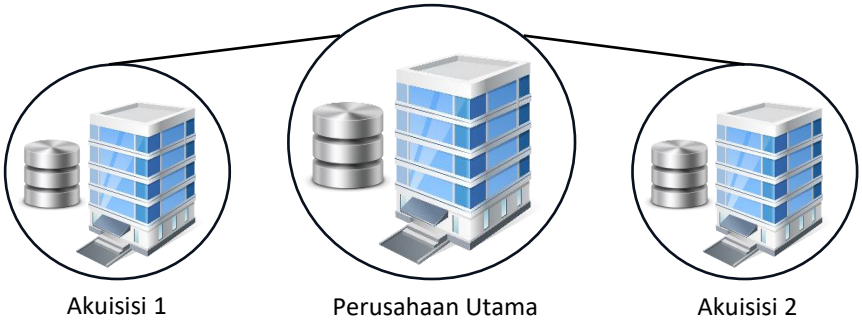
Ada sejumlah besar aktivitas merger dan akuisisi yang terjadi setiap tahun di antara perusahaan-perusahaan, khususnya perusahaan-perusahaan besar atau menengah yang mengakuisisi perusahaan-perusahaan kecil. Hampir seribu merger dan akuisisi terjadi setiap tahun [26], dengan merger yang menyatukan dua perusahaan dengan ukuran yang kira-kira sama menjadi satu unit, dan akuisisi yang menyatukan perusahaan yang lebih kecil ke dalam perusahaan yang lebih besar. Ketika perusahaan yang diakuisisi mempunyai ukuran yang masuk akal, akan sulit untuk mengkonsolidasikan sistem TI yang berbeda dari berbagai komponen perusahaan secara bersamaan. Meskipun beberapa sistem TI pada akhirnya akan digabungkan menjadi satu komponen TI, konvergensi tersebut seringkali memerlukan waktu bertahun-tahun.

Ketika integrasi sedang berlangsung, skema dan data yang dikumpulkan oleh berbagai komponen perusahaan bisa sangat berbeda. Dalam kasus ini, skema data yang dikumpulkan di lokasi yang berbeda mungkin berbeda, ditambah lagi dengan tantangan lain yang dihadapi oleh suatu perusahaan yang terdiri dari beberapa anak perusahaan. Dalam beberapa kasus, pihak yang mengakuisisi mungkin diminta untuk mempertahankan operasinya agar independen dan terpisah bahkan setelah mereka menjadi satu kesatuan. Misalnya, perusahaan yang menjalankan bisnis apotek dan telah mengakuisisi bisnis asuransi mungkin perlu memisahkan data dari kedua sisi akuisisi.

Jika kita membandingkan struktur akuisisi/merger dengan anak perusahaan, hubungan antar entitas yang berbeda lebih bersifat hubungan sejawat dibandingkan dengan anak perusahaan/penerima waralaba. Meskipun perusahaan yang mengakuisisi, yang ditunjukkan sebagai Perusahaan Utama pada Gambar 2.10, mungkin memiliki posisi yang lebih kuat untuk mengendalikan operasi perusahaan yang diakuisisi, perusahaan tersebut sering kali mendapati bahwa biaya yang terkait dengan perubahan operasi akuisisi relatif mahal. . Tergantung pada tingkat integrasi antara perusahaan utama dan akuisisinya, data yang disimpan dalam akuisisi yang berbeda dapat disimpan secara terpisah. Dalam jangka panjang,

ada kemungkinan bahwa semua proses TI dan data terkait akan tersedia di satu lokasi pusat yang dikendalikan oleh perusahaan utama. Namun, hingga integrasi akhirnya terwujud, akan terdapat banyak silo data dan informasi independen yang berbeda dalam keseluruhan bisnis.

Perusahaan utama dapat memilih untuk menyalin data dari akuisisinya ke gudang data pusat atau data lake untuk menganalisis dan mengeksplorasi pola di seluruh data mereka. Namun, karena perpindahan data tersebut seringkali mahal dan memakan waktu, dan dalam beberapa kasus tidak dapat dilakukan karena adanya pembatasan peraturan, pembelajaran gabungan menyediakan mekanisme untuk mengekstraksi pola dari data yang didistribusikan di banyak lokasi berbeda.



Gambar 2.10 Pengaturan merger dan akuisisi.

2.4.3 Operasi yang Dialihdayakan

Dalam bisnis modern, outsourcing operasi adalah praktik yang umum. Banyak perusahaan fokus pada operasi yang mereka anggap sebagai kompetensi inti bisnis mereka dan melakukan outsourcing semua operasi lainnya ke perusahaan lain. Dukungan infrastruktur Teknologi Informasi yang diperlukan untuk menjalankan operasional bisnis seringkali menjadi sasaran outsourcing tersebut. Dalam kasus ini, perusahaan akan memilih perusahaan lain (perusahaan penyedia) yang memiliki keahlian dan pengetahuan operasional untuk melakukan dukungan TI guna mengelola infrastrukturnya. Penyedia biasanya menyediakan layanan ini kepada banyak perusahaan klien yang berbeda, dan tidak bergantung pada satu perusahaan saja. Dengan kata lain, satu penyedia spesialis mungkin mendukung banyak perusahaan berbeda. Selain dukungan dan operasional TI, outsourcing semacam itu juga cukup umum untuk banyak aspek operasional lainnya, termasuk pemeliharaan fasilitas, manajemen dan pemeliharaan inventaris dan aset, analisis data, pengiriman, penggajian, dll.

Penyedia memiliki akses ke informasi yang berasal dari banyak klien berbeda. Secara umum, data dari satu klien dapat dianggap sebagai hak milik klien tersebut, dan merupakan praktik bisnis yang buruk jika penyedia berbagi data dari satu klien dengan klien lainnya. Namun, penyedia juga dapat memperoleh keuntungan yang signifikan dari pola pembelajaran yang umum di seluruh data klien yang didukungnya, dan menggunakannya untuk meningkatkan proses bisnisnya sendiri.

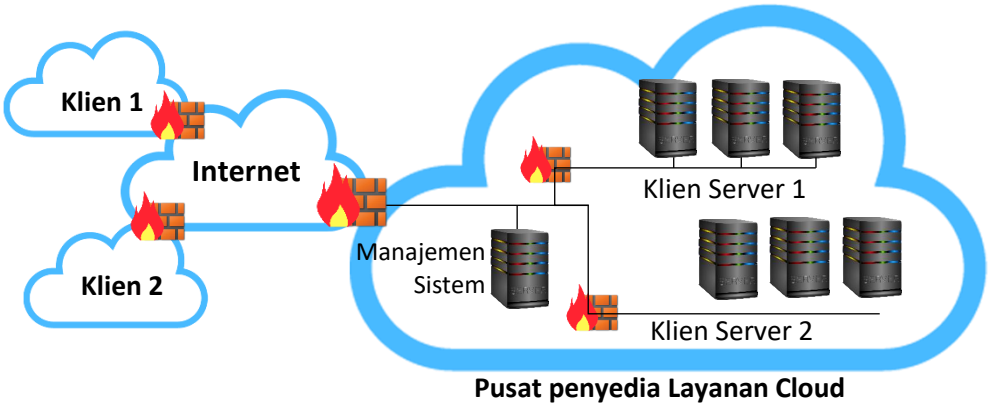
Sebagai contoh, mari kita ambil kasus pengelolaan aset fisik sebuah perusahaan, sebuah tugas yang seringkali lebih baik didelegasikan kepada perusahaan lain yang dapat

menjalankan layanan manajemen aset berbasis cloud. Layanan manajemen aset fisik akan melacak aset yang dimiliki oleh perusahaan, catatan layanan dan pemeliharaannya, dan kapan aset tersebut mengalami kegagalan. Pengaturan umum yang digunakan oleh perusahaan penyedia ditunjukkan pada Gambar 2.11. Server yang digunakan untuk mendukung klien yang berbeda dipisahkan dan dibuat aman melalui firewall yang dapat mengisolasi klien yang berbeda satu sama lain di pusat cloud atau pusat data penyedia. Sistem manajemen digunakan oleh penyedia untuk mendukung pengoperasian berbagai server klien yang berbeda, dan dilindungi oleh firewall. Jaringan pribadi virtual dapat diatur antara kampus masing-masing klien dan server yang mungkin mereka miliki di pusat cloud penyedia. Dalam pengaturan sebenarnya, beberapa tingkatan firewall lainnya dapat digunakan untuk memberikan keamanan tambahan.

Jika perusahaan penyedia di cloud menghosting layanan pemeliharaan aset dari banyak perusahaan berbeda, perusahaan tersebut akan dapat menggabungkan pengetahuan yang diperoleh dari hosting di satu perusahaan untuk meningkatkan proses hosting di perusahaan lain. Hal ini akan menguntungkan semua perusahaan yang menjadi tuan rumah. Model AI untuk meningkatkan layanan dapat dibuat pada sistem manajemen penyedia dengan mengumpulkan semua informasi yang tersedia dari klien yang berbeda. Namun, perusahaan yang bekerja sama dengan penyedia mungkin mewaspadai penggunaan informasi aset mereka. Penyedia mungkin menjadi tuan rumah kompetisi mereka, dan mereka mungkin khawatir dengan kompetisi yang mempelajari riwayat aset mereka. Mereka mungkin tidak ingin informasi tentang aset mereka tercampur dengan informasi umum tentang seluruh aset. Namun, mereka mungkin bersedia membiarkan penyedia layanan mengembangkan model yang memprediksi kapan pemeliharaan harus dijadwalkan, atau apa saja indikator prediksi kegagalan. Berbagi model ini dan menggabungkannya ke seluruh klien penyedia mungkin dapat diterima.

Dalam kasus layanan cloud hosting, situasi serupa muncul bukan pada data yang dimiliki oleh klien, namun dalam hal karakteristik pengoperasian mesin yang digunakan untuk menghosting layanan tersebut. Untuk menyediakan isolasi antar klien, penyedia mungkin menggunakan pendekatan seperti mesin fisik yang berbeda untuk setiap klien, mesin virtual terpisah untuk setiap klien, atau wadah terpisah untuk setiap klien. Metrik kinerja untuk masing-masing mekanisme pemisahan ini dapat digunakan oleh penyedia untuk mendapatkan wawasan tentang pengelolaan infrastruktur yang lebih baik, dan mengotomatiskan pemicu yang dapat menandai perilaku layanan yang tidak wajar. Namun, perjanjian berbagi data antara klien dan penyedia mungkin menghalangi pembagian data ini dengan klien lain, atau pencampuran data dari klien yang berbeda.

Penggerak penggunaan pembelajaran gabungan dalam hal ini adalah pembatasan kontrak yang ada antara perusahaan klien dan penyedia, dan membatasi aliran data melintasi batas-batas organisasi.



Gambar 2.11 Struktur operasi yang dialihdayakan.

2.4.4 Jaringan Telekomunikasi

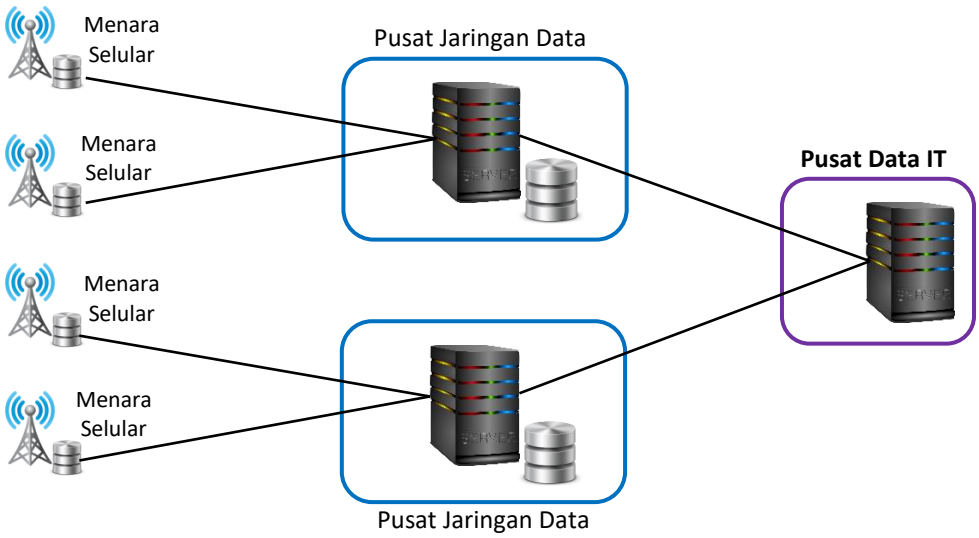
Kadang-kadang, kebutuhan akan pembelajaran gabungan mungkin muncul hanya karena skala data yang dihasilkan, skala tersebut bersifat sedemikian rupa sehingga pemindahan data ke lokasi pusat untuk dianalisis tidak mungkin dilakukan. Hal ini terjadi ketika data dihasilkan terlalu cepat untuk diangkut melalui jaringan, atau data yang disimpan terlalu besar untuk dipindahkan.

Pengaturan umum dimana situasi ini muncul adalah dalam konteks jaringan seluler. Jaringan seluler biasanya berisi banyak lokasi menara seluler, yang mendukung berbagai telepon seluler. Selama pengoperasiannya, perangkat tersebut mungkin berisi data dari berbagai telepon berbeda, termasuk namun tidak terbatas pada informasi tentang sinyal radio yang diterimanya, serta variasi kekuatan dan intensitas sinyal. Meskipun topologi fisik jaringan yang berbeda mungkin berbeda, data apa pun yang dikumpulkan dari menara seluler biasanya akan dikirim ke pusat data jaringan, yang masing-masing pusat data jaringan mendukung beberapa lokasi menara seluler. Informasi dari pusat data jaringan dapat digabungkan menjadi pusat operasi jaringan atau dibawa ke pusat data TI. Hal ini menghasilkan struktur seperti pohon logis sejauh menyangkut pemrosesan data di lingkungan seluler, yang ditunjukkan pada Gambar 2.12.

Volume sinyal frekuensi radio di setiap menara seluler sangat besar, sehingga informasi ini tidak dapat dikirim melalui menara seluler ke pusat data jaringan. Analisis atau model AI yang akan dilakukan terhadap sinyal tersebut perlu dilakukan di menara seluler itu sendiri. Jika model yang mencakup pola-pola yang termasuk dalam lebih dari satu menara seluler perlu dibuat, maka pembuatan model gabungan adalah satu-satunya pilihan yang layak.

Dalam jaringan telekomunikasi, catatan tentang koneksi data atau panggilan suara yang dilakukan oleh klien dicatat di pusat data jaringan, dan disalin secara berkala ke pusat TI untuk tujuan pencatatan dan penagihan. Sebuah perusahaan telekomunikasi mungkin memiliki jutaan pelanggan, dan mungkin mencatat miliaran panggilan setiap bulannya. Mereka mungkin juga memiliki pusat data TI yang berbeda, masing-masing pusat data TI mencakup wilayah geografis yang berbeda. Karena volume data di setiap pusat data, maka diperlukan biaya dan waktu yang lama untuk memindahkan catatan panggilan yang berbeda

secara bersamaan, dan akan lebih bijaksana jika menggunakan teknik pembelajaran gabungan untuk mengekstrak pola dari catatan panggilan.



Gambar 2.12 Struktur jaringan telekomunikasi.

Meskipun contoh-contoh ini diberikan dalam konteks jaringan telekomunikasi, sejumlah besar data yang disimpan di banyak lokasi berbeda dapat muncul di banyak industri berbeda untuk perusahaan besar.

2.4.5 Konsorsium dan Koalisi

Sebuah konsorsium terdiri dari banyak perusahaan berbeda yang berkumpul untuk berbagi informasi atau rincian di bidang tertentu. Sebagai contoh, konsorsium bank mungkin tertarik untuk berbagi wawasan agar dapat mendeteksi penipuan dengan lebih baik. Mereka mungkin dapat berbagi dan memberikan model satu sama lain, namun tidak dapat membagikan seluruh jumlah data secara jelas. Dalam kasus lain, konsorsium dapat berupa sekelompok perusahaan medis, yang dapat berbagi data satu sama lain, namun hanya jika data tersebut dianonimkan dan digabungkan. Mereka ingin bekerja satu sama lain sambil menjaga privasi anggota yang disimpan dalam catatan mereka.

Konsorsium layanan kesehatan adalah representasi khas dari kebutuhan anggota untuk menjaga anonimitas dan privasi catatan yang mereka simpan sambil tetap mencoba berbagi data yang dapat bermanfaat bagi seluruh anggota konsorsium. Konsorsium tersebut dapat dibentuk antara negara-negara berbeda di suatu kawasan, dan negara-negara tersebut mungkin ingin berbagi informasi tentang penyebaran penyakit menular di rumah sakit mereka, atau berbagi informasi tentang kemanjuran obat-obatan untuk mengendalikan penyakit tertentu. Model terbaik untuk konsorsium dapat dibuat dengan menggabungkan data dari seluruh anggota, namun berbagi data mentah mungkin tidak dapat dilakukan berdasarkan peraturan privasi negara anggota. Namun, membuat model dari repositori data individual mungkin memberikan alternatif yang dapat diterima.

Contoh spesifik dari konsorsium adalah koalisi militer, yang terdiri dari angkatan bersenjata dari berbagai negara yang bekerja sama untuk mencapai tujuan militer tertentu.

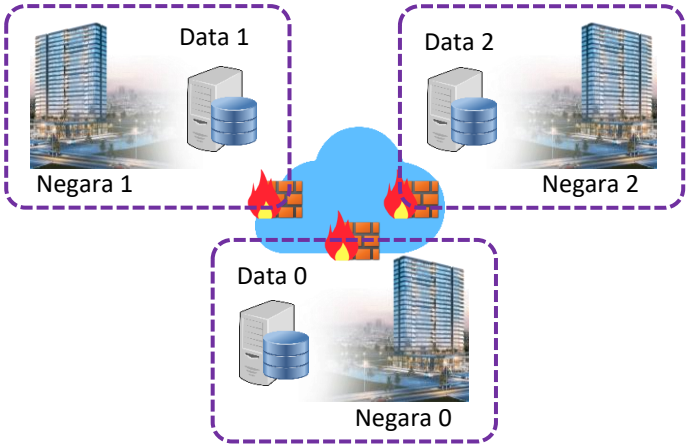
Koalisi mungkin tertarik untuk menjaga perdamaian di wilayah yang mungkin sedang mengalami kerusuhan regional, atau koalisi mungkin tertarik untuk menjaga keseimbangan kekuatan strategis melawan koalisi lain. Anggota koalisi akan berbagi data satu sama lain, namun mereka mungkin tidak berbagi semua data yang mereka miliki dengan anggota lain. Sebagai contoh, seorang anggota koalisi mungkin telah mengembangkan teknologi untuk pencitraan resolusi sangat tinggi dan mungkin tidak ingin mengungkapkan fakta tersebut kepada anggota koalisi lainnya. Daripada membagikan gambar mentah beresolusi tinggi, anggota koalisi mungkin lebih memilih untuk membagikan gambar beresolusi rendah menggunakan teknologi yang dapat diakses oleh semua anggota koalisi. Jika berbagi gambar menghabiskan terlalu banyak bandwidth, anggota koalisi mungkin lebih memilih untuk bertukar model AI karena bandwidth dan konektivitas jaringan untuk anggota koalisi yang beroperasi di daerah terpencil biasanya terbatas dan terbatas.

Skenario umum dalam koalisi militer yang memerlukan pembelajaran gabungan ditunjukkan pada Gambar 2.13. Ini menunjukkan tiga negara, yang masing-masing memiliki pasukannya yang ditempatkan di base camp mereka. Tugas koalisi adalah memantau wilayah tersebut untuk memastikan bahwa gencatan senjata yang disepakati oleh pihak-pihak yang bertikai di wilayah tersebut tetap dipertahankan. Untuk melakukan pemantauan, setiap anggota koalisi akan mengerahkan berbagai peralatan pengawasan untuk memantau wilayah sekitar base camp mereka. Mereka dapat mengotomatiskan pemrosesan informasi pengawasan dengan berbagi data satu sama lain sehingga semua negara dapat membangun model AI mereka untuk menganalisis informasi yang dikumpulkan. Namun, mereka mungkin tidak selalu bersedia berbagi data, dan bahkan jika mereka bersedia berbagi data, jaringan yang menghubungkan base camp mereka mungkin tidak memiliki bandwidth yang memadai.

Demikian pula, kapal angkatan laut atau pesawat terbang milik suatu negara dapat mengumpulkan berbagai data menggunakan instrumen yang mereka miliki. Jika kapal ingin menganalisis data kolektif yang telah mereka kumpulkan, akan lebih efisien bandwidth jika mereka berbagi model, dibandingkan mengirimkan data dalam jumlah besar melalui jaringan nirkabel atau satelit yang menghubungkan kapal-kapal yang berbeda. Hal yang sama juga berlaku untuk kapal angkatan laut atau pesawat terbang yang berkolaborasi bersama sebagai koalisi dari berbagai negara, dengan kompleksitas tambahan bahwa pembacaan sensor mungkin tidak dibagikan secara bebas ke seluruh negara, meskipun mereka adalah bagian dari koalisi yang sama.

Konteks militer yang sangat mirip dengan operasi koalisi adalah operasi multidomain. Bagi negara besar mana pun, angkatan bersenjatanya dikhususkan untuk beroperasi secara efektif di berbagai wilayah peperangan, yang suatu wilayahnya terdiri dari wilayah operasi tertentu. Domain operasi yang umum meliputi darat, laut, udara, luar angkasa, udara, dan dunia maya. Domain darat akan terdiri dari sensor, kendaraan dan tentara yang beroperasi di darat, domain laut akan terdiri dari kapal dan pelaut, domain udara akan terdiri dari UAV, pesawat terbang dan kendaraan udara lainnya, domain luar angkasa akan terdiri dari sinyal satelit, dan domain cyber akan terdiri dari komputer dan jaringan komunikasi, termasuk Internet. Di setiap domain, militer negara mana pun pasti ingin mengambil posisi lebih unggul

dibandingkan kekuatan musuhnya. Dalam operasi multi-domain, tentara menggabungkan informasi yang tersedia di semua domain untuk mencoba mendapatkan keuntungan di setiap domain.



Gambar 2.13 Skenario koalisi.

Karena jaringan komunikasi antara domain yang berbeda akan dibatasi dibandingkan dengan permintaan yang dibuat oleh data yang dihasilkan di lapangan, mengekstraksi pola sebagai model AI dari setiap domain dan kemudian menggabungkannya akan memberikan pendekatan terbaik untuk menciptakan model AI di seluruh domain. melakukan listrik.

2.4.6 Industri yang Diatur

Banyak industri beroperasi di bawah pedoman peraturan ketat yang diberlakukan oleh pemerintah. Pedoman peraturan biasanya dibuat demi kepentingan masyarakat yang lebih luas, dan untuk mencegah penyalahgunaan informasi yang dikumpulkan oleh badan usaha di industri, dan untuk memastikan bahwa hak dan manfaat masyarakat umum tetap terjaga. Namun, pedoman peraturan sering kali menghalangi pembagian informasi di berbagai bagian organisasi.

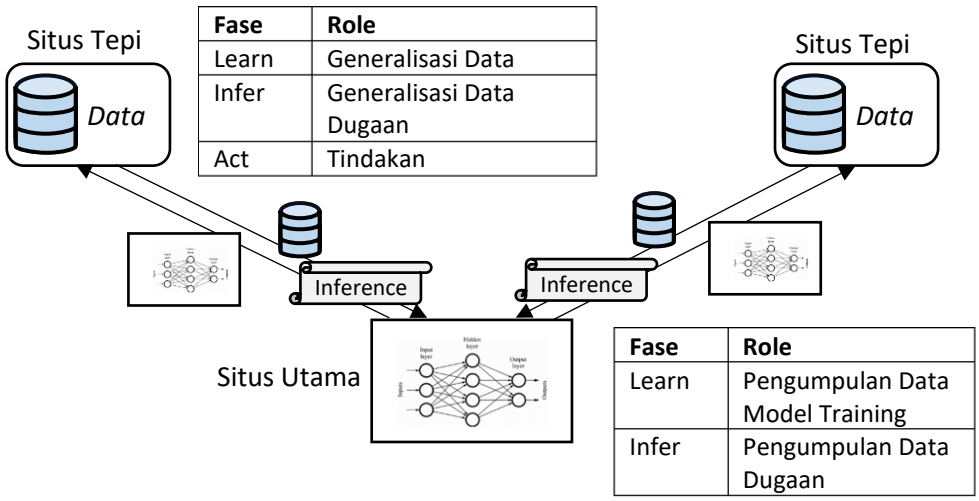
Contoh industri yang menerapkan peraturan ketat dalam berbagi informasi adalah layanan kesehatan. Catatan klinis pasien hanya dapat dibagikan dalam kondisi terbatas dengan persetujuan pasien. Meskipun dokter dapat mengakses catatan untuk memberikan perawatan medis bagi pasien, akses dibatasi untuk sebagian besar entitas lain. Hal ini membantu mencegah penyalahgunaan informasi pasien, misalnya. menghindari situasi di mana perusahaan asuransi jiwa dapat mengakses catatan kesehatan pelanggannya dan membatalkan polis asuransi jiwa siapa pun yang mungkin mengalami cedera atau kecelakaan. Namun, pembatasan pembagian data ini juga dapat menghalangi akses untuk tujuan yang sah dan bermanfaat.

Salah satu penggunaan yang mungkin tidak diperbolehkan di beberapa negara ditunjukkan pada Gambar 2.14. Sistem rumah sakit yang besar dapat menangani pasien serta melakukan penelitian medis. Biasanya, organisasi penelitian berbeda dengan organisasi klinis (rawat jalan). Data yang dikumpulkan pada kelompok klinis yang berbeda dapat disimpan secara terpisah tergantung pada jenis kontrol akses yang diperlukan oleh sistem. Model AI

dalam kasus ini akan dijalankan oleh organisasi penelitian, yang biasanya memiliki kapasitas pemrosesan dan keahlian untuk membuat model. Namun, organisasi penelitian tidak diizinkan mengakses data mentah. Skenario yang ditunjukkan pada Gambar 2.14 memiliki dua lokasi klinis dan satu pusat analisis, meskipun rantai rumah sakit sebenarnya kemungkinan besar memiliki lebih banyak lokasi klinis serta beberapa pusat analisis.

Salah satu cara untuk memenuhi persyaratan ini adalah dengan mengubah data untuk menyembunyikan informasi pribadi apa pun sebelum membuat model AI darinya. Namun, penerapan variasi Fusion AI, yang mampu memproses data mentah di setiap lokasi klinis, kemungkinan besar akan memberikan model yang lebih akurat.

Peraturan juga mengontrol pengoperasian banyak industri lainnya. Dalam hal keuangan, mungkin terdapat kontrol ketat terhadap pembagian informasi tentang transaksi keuangan klien dengan entitas lain. Di antara pembatasan lainnya, hal ini mungkin menghalangi pembagian catatan melintasi batas negara. Karena sebagian besar perusahaan keuangan adalah organisasi multinasional, hal ini dapat menimbulkan tantangan jika mereka ingin membuat model yang mencakup data dari lebih dari satu negara. Federated Learning memberikan satu solusi yang mungkin untuk membantu mereka membuat model seperti itu tanpa mentransfer data mentah melintasi batasan peraturan. Peraturan juga dapat membatasi lembaga-lembaga pemerintah yang berbeda untuk saling berbagi informasi. Dalam situasi ini, pembelajaran gabungan dapat memberikan pendekatan bagi berbagai lembaga untuk membuat model yang menangkap pola di seluruh data mereka, tanpa bertukar data mentah.



Fase	Role
Learn	Generalisasi Data
Infer	Generalisasi Data Dugaan
Act	Tindakan

Fase	Role
Learn	Pengumpulan Data Model Training
Infer	Pengumpulan Data Dugaan

Gambar 2.14 Skenario perusahaan layanan kesehatan.

Secara umum, ketika peraturan menghalangi pertukaran data secara bebas antar lokasi yang berbeda, namun mungkin ada manfaatnya dalam membangun model AI yang mencakup pola dari data di banyak lokasi, maka pembelajaran gabungan dapat memberikan solusi yang tepat.

2.5 RINGKASAN

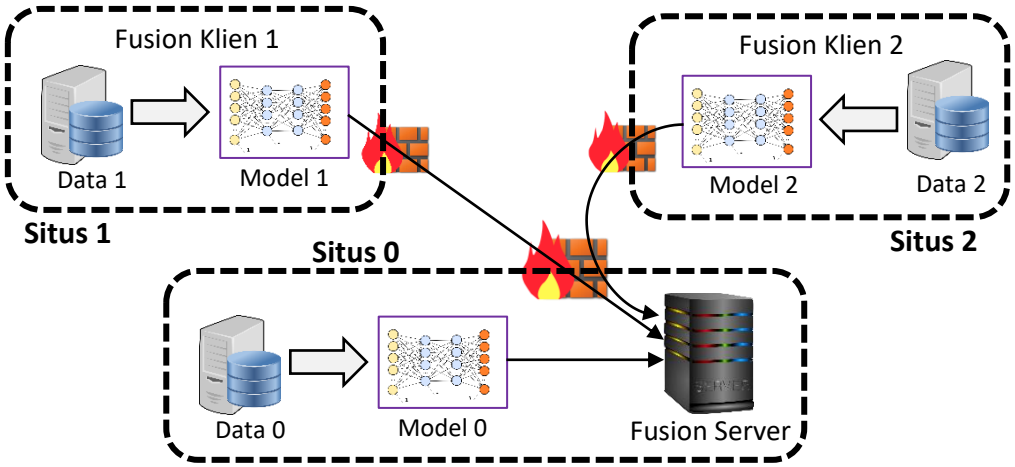
Dalam bab ini, kita telah mengeksplorasi beberapa pola penerapan solusi berbasis AI dan bagaimana berbagai situs dapat mengambil peran berbeda dalam berbagai pola tersebut. Beberapa dari pola tersebut menunjukkan perlunya federasi antar lokasi yang berbeda selama fase pembelajaran, sementara pola lainnya menunjukkan perlunya federasi antar lokasi yang berbeda selama fase inferensi. Kami meninjau beberapa motivasi yang mendorong perlunya federasi ini. Kami telah menjelaskan perbedaan antara pembelajaran gabungan konsumen dan pembelajaran gabungan perusahaan, dan menjelaskan beberapa skenario di mana pembelajaran gabungan untuk perusahaan akan berguna.

BAB 3
PENDEKATAN PEMBELAJARAN FEDERASI YANG NAIF

Seperti disebutkan dalam bab sebelumnya, proses pembuatan model AI menangkap pola yang ada dalam data pelatihan. Selama pembelajaran gabungan, pola-pola yang ditangkap dari banyak lokasi berbeda digabungkan. Dalam bab ini, kita melihat pendekatan dasar yang dapat digunakan untuk menggabungkan pola dari lokasi yang berbeda. Pendekatan yang sangat populer untuk menggabungkan model adalah pendekatan federasi dengan rata-rata. Pendekatan ini telah menjadi fokus penelitian akademis yang signifikan.

Dalam bab ini, kita melihat prinsip rata-rata gabungan dan mendiskusikan pendekatan yang mendasari pendekatan ini. Kami menyebut pendekatan ini sebagai pendekatan pembelajaran gabungan yang Naif, karena pendekatan ini membuat beberapa asumsi yang mungkin sulit dipenuhi dalam konteks perusahaan.

Konfigurasi sistem di mana kami menyajikan berbagai algoritma yang dijelaskan dalam bab ini dan bab selanjutnya diilustrasikan pada Gambar 3.1. Kami berasumsi bahwa data untuk perusahaan berlokasi di banyak situs berbeda, yang kami sebut sebagai lokasi klien Fusion. Pada gambar, kami menunjukkan dua klien fusi tetapi akan ada lebih banyak lagi tergantung pada pengaturan spesifik sistem. Terdapat lokasi server Fusion yang ditampilkan sebagai situs 0. Penamaan situs sebagai klien dan server dilakukan karena kami berasumsi bahwa pendekatan klien-server yang ada di mana-mana untuk komputasi terdistribusi sedang digunakan dalam lingkungan.



Gambar 3.1 Konfigurasi pembelajaran gabungan.

Konfigurasi yang ditunjukkan pada Gambar 3.1 adalah konfigurasi yang diperlukan selama pelatihan model AI selama fase pembelajaran siklus Learn→Infer→Act dalam pola edge learning (dijelaskan di Bagian 2.1.4 Bab 2) dan pembelajaran proksi (dijelaskan dalam Bagian 2.1.5 Bab 2) dan selama fase inferensi dari inferensi tepi gabungan (dijelaskan dalam Bagian 2.1.3 Bab 2).

Tujuan dari keseluruhan sistem adalah untuk melakukan tugas pada contoh data yang terdistribusi tanpa memindahkan data mentah di setiap klien fusi. Tugasnya mungkin mempelajari model AI, menggunakan model AI untuk membuat inferensi, atau menghitung beberapa metrik lain pada data yang didistribusikan. Pendekatan umum untuk melakukan tugas ini adalah setiap klien fusi menghitung model, inferensi, atau metrik lokal dan berbagi metrik dengan server fusi. Server fusi kemudian akan menggabungkan atau memadukan model, kesimpulan, atau metrik yang disediakan oleh semua klien fusi dan memberikan versi gabungannya kembali kepada mereka.

3.1 PEMBELAJARAN METRIK GABUNGAN

Sebagai langkah komponen dalam pelatihan dan penggunaan model AI, seringkali diperlukan beberapa metrik yang berlaku untuk semua data yang didistribusikan di berbagai lokasi. Metrik ini akan dihitung untuk keseluruhan jumlah data, namun perlu dilakukan tanpa memindahkan data apa pun.

Sebagai gambaran pada bagian ini, kita asumsikan bahwa data pada setiap klien fusi terdiri dari sekumpulan masukan x_1, x_2, ...x_N dan keluaran y. Setiap klien fusi memiliki beberapa contoh data ini dan kami ingin menghitung berbagai metrik pada data ini di semua klien fusi. Mari kita asumsikan bahwa semua metrik bersifat numerik.

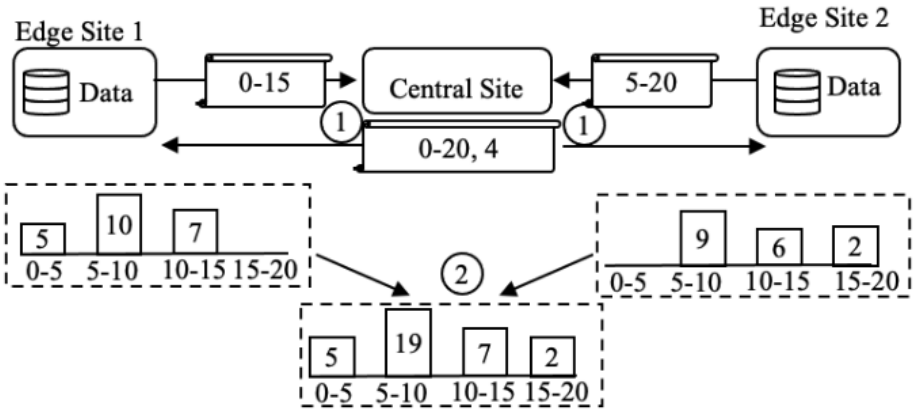
Beberapa metrik yang mudah dikumpulkan di seluruh klien fusi yang berbeda tanpa memindahkan data apa pun adalah atribut dari jenis data tertentu, seperti jumlah total, mean, distribusi/varian minimum, maksimum, dan standar dari setiap x_i di seluruh klien fusi yang berbeda. Penghitungan dapat dilakukan di seluruh klien fusi yang berbeda tanpa memindahkan datanya. Masing-masing klien fusi dapat mengirimkan atribut lokalnya ke server fusi, yang kemudian dapat menggabungkannya untuk mendapatkan perkiraan keseluruhan.

Proses agregasi untuk beberapa metrik mungkin mencakup pengirimannya bersama dengan jumlah total data. Ketika rata-rata, distribusi standar, atau varians perlu diagregasi, setiap klien fusi dapat mengirimkan metrik yang dihitung secara lokal beserta jumlah total titik data. Server fusi dapat membuat rata-rata metrik dengan bobot untuk setiap metrik klien fusi, yang ditentukan oleh jumlah total titik data yang menyertainya. Untuk metrik lainnya, seperti minimum atau maksimum, jumlah minimum yang dilaporkan oleh semua klien fusi adalah jumlah minimum agregat dan jumlah maksimum yang dilaporkan oleh semua klien fusi lainnya merupakan jumlah maksimum gabungan.

Distribusi gabungan atau statistik tentang data yang didistribusikan juga dapat dihitung tanpa memindahkan datanya. Mari kita asumsikan bahwa server fusi perlu menghitung keseluruhan distribusi x_1 yang berbeda di semua klien fusi. Ini memerlukan dua putaran komunikasi antara klien fusi dan server fusi. Pada putaran pertama, klien fusi akan mengirimkan rentang (yaitu maksimum dan minimum) x_1 ke server fusi. Server fusi akan menentukan rentang agregat, dan juga mengidentifikasi jumlah kelompok di mana data harus dibagi. Setiap klien fusi lokal kemudian dapat menghitung berapa banyak instance yang

dimilikinya dalam setiap rentang, dan mengirimkannya ke server fusion. Server fusi kemudian dapat mengagregasi bucket di seluruh klien fusi dan menemukan distribusi agregat.

Contoh dua klien fusi ditunjukkan pada Gambar 3.2. Setiap klien fusi memiliki beberapa data. Pada langkah pertama, klien fusi mengirimkan rentang datanya ke server fusi, yaitu masing-masing 0-15 dan 5-20. Server fusi menghitung rentang keseluruhan sebagai 0-20, dan menginstruksikan setiap klien fusi untuk membaginya menjadi 4 grup dengan ukuran yang sama, yaitu 5 unit. Pada langkah kedua, masing-masing klien fusi menghitung histogram distribusi pada grup ini, dan mengirimkannya ke server fusi. Server fusi dapat menghitung keseluruhan histogram dengan menjumlahkan entri dalam keranjang yang sesuai.



Gambar 3.2 Perhitungan distribusi statistik gabungan.

Proses serupa dapat digunakan di banyak klien fusi dan untuk menghitung jenis metrik lainnya termasuk persentil, median, atau metrik lainnya. Secara umum, komputasi apa pun yang dapat dipecah menjadi dua komponen, modul eksekusi klien fusi dan modul eksekusi server fusi, dapat diimplementasikan sehingga data tidak perlu berpindah-pindah. Masing-masing klien fusi memanggil modul klien fusi yang dapat dieksekusi untuk menghasilkan keluaran perantara. Keluaran perantara dapat dikirim ke server fusi yang dapat memproses informasi di seluruh keluaran perantara dari klien fusi yang berbeda, dan dapat menghasilkan keluaran akhir. Proses ini dapat diulangi untuk menghitung serangkaian fungsi yang kompleks tanpa memindahkan data apa pun.

Pendekatan ini merupakan pemanggilan fungsi area luas yang sering digunakan untuk komputasi paralel dalam suatu situs. Secara umum, pendekatan apa pun yang dapat diterapkan dengan paradigma pemrosesan data terpisah (misalnya pengurangan peta [33]) dapat diterapkan secara gabungan. Secara khusus, semua jenis metrik Indikator Kinerja Utama (KPI) yang digunakan oleh bisnis dapat dihitung secara gabungan menggunakan pendekatan ini.

3.2 ESTIMASI FUNGSI

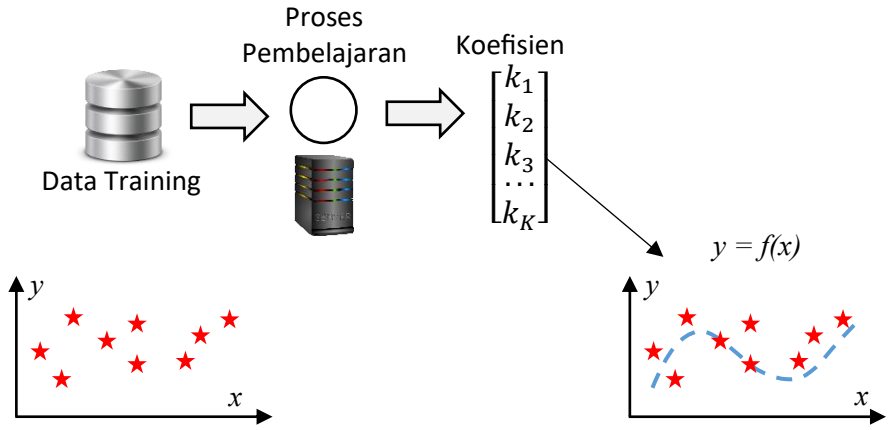
Memperkirakan fungsi yang sesuai dengan data yang tersedia di setiap klien fusi adalah jenis umum pembuatan model AI yang diperlukan untuk berbagai operasi bisnis. Pada bagian

ini, kita mengeksplorasi bagaimana fungsi tersebut dapat diperkirakan ketika data didistribusikan ke beberapa klien fusi dengan bantuan server fusi dalam konfigurasi yang diilustrasikan pada 3.1. Teknik yang mendasari estimasi fungsi akan serupa dengan rata-rata gabungan.

Mari kita asumsikan bahwa tugas model AI adalah mengambil beberapa variabel N x_1, x_2 . . . x_N , dan untuk menghasilkan nilai y. Dengan kata lain, kita berasumsi bahwa proses pembelajaran mesin mengambil data pelatihan masukan yang terdiri dari banyak catatan, masing-masing berisi variabel masukan dan variabel keluaran yang sesuai, dan menghasilkan representasi fungsi $y = f(x_1, x_2 \ldots x_N)$ yang merupakan estimasi terbaik dari pola yang terdapat pada data masukan. Ini akan menjadi contoh pembelajaran yang diawasi.

Estimasi fungsi dapat dipandang sebagai pendekatan mendasar yang mencirikan sejumlah besar masalah pembelajaran mesin. Beberapa pendekatan dalam AI, seperti pelatihan jaringan saraf, pelatihan pohon keputusan, pelatihan tabel keputusan, pembelajaran aturan simbolik, klasifikasi, deteksi anomali, dll., dapat dipandang sebagai cara alternatif untuk merepresentasikan suatu fungsi.

Meskipun terdapat banyak algoritma yang berbeda untuk estimasi fungsi, dan pilihan estimasi fungsi tertentu bergantung pada domain yang dipelajari, kita dapat memodelkan tugas umum estimasi fungsi sebagai pemetaan kumpulan data pelatihan yang tersedia ke sekumpulan angka K. Nilai dan semantik angka K ini bergantung pada algoritma yang digunakan untuk estimasi fungsi.



Gambar 3.3 Pembuatan model AI sebagai estimasi fungsi.

Pemetaan himpunan data latih ke himpunan bilangan K dijelaskan secara visual pada Gambar 3.3. Data pelatihan diasumsikan hanya memiliki satu masukan x yang akan dipetakan ke keluaran y. Pembatasan satu ini semata-mata untuk ilustrasi karena grafik 2 dimensi mudah dijelaskan. Data pelatihan dapat divisualisasikan sesuai grafik yang ditunjukkan di sebelah kiri Gambar 3.3, di mana setiap titik data ditandai dengan bintang. Namun, setiap titik data bersifat independen, dan pola dalam data tidak ditangkap dengan cara apa pun. Setelah proses pelatihan, serangkaian K angka K_1 . . . K_K dihasilkan, yang merupakan cara untuk merepresentasikan estimasi terbaik dari hubungan fungsional antara x dan y. Fungsi tersebut ditampilkan sebagai kurva putus-putus pada grafik di sebelah kanan gambar.

Ada banyak algoritma berbeda yang dapat digunakan untuk menghitung angka K yang mewakili fungsi tersebut. Sebagai contoh, regresi linier adalah algoritma yang sangat umum digunakan dalam pendekatan pembelajaran model AI. Tugas regresi linier adalah menemukan hubungan linier yang paling sesuai di antara variabel-variabel masukan yang dapat memprediksi variabel keluaran. Dalam representasi ini, tugas mempelajari suatu fungsi dengan input x_1, x_2 . . . x_N dan keluaran y menghasilkan representasi fungsi dengan $K = N + 1$, dan pembelajaran mesin menghitung $N + 1$ angka α_0, α_1 . . . α_N yang memberikan tebakan terbaik untuk hubungan antara keluaran y dan masukan melalui persamaan linier

$$y = \alpha_0 + \begin{matrix} i = N \\ i = 1 \end{matrix} \alpha_i x_i$$

Meskipun regresi linier hanya menangkap hubungan linier antara masukan dan keluaran, ada banyak algoritme pembelajaran mesin yang juga menangkap hubungan nonlinier. Meskipun ada banyak algoritme untuk menghitung hubungan non-linier, algoritme tersebut pada akhirnya menghasilkan pembuatan matriks $beta$ dengan ukuran Mx (N + 1) dengan M adalah pangkat tertinggi dari semua variabel masukan yang dipertimbangkan, dan N adalah bilangan masukannya. Estimasi fungsi terbaik diberikan oleh hubungan:

$$y = \begin{matrix} i = N \\ i = 1 \end{matrix} \beta_0, i x_i + \begin{matrix} i = N \\ i = 1 \end{matrix} \beta_1, j {x_i}^j$$

Ada banyak cara untuk menghitung hubungan non-linier antar fungsi yang berbeda, dan pendekatan yang umum adalah penggunaan fungsi kernel. Fungsi kernel mencoba memetakan hubungan non-linier di antara variabel masukan menjadi hubungan linier dalam ruang yang ditransformasikan, dan menyediakan metode yang efisien untuk menghitung hubungan non-linier. Tanpa membahas kompleksitasnya algoritma, mereka menghasilkan matriks $beta$ berukuran Mx (N + 1). Ini adalah cara ringkas untuk merepresentasikan angka Mx (N + 1). Dengan kata lain, K sama dengan Mx (N + 1).

Pendekatan lain untuk estimasi fungsi menggunakan serangkaian fungsi dasar. Fungsi yang dimodelkan diasumsikan merupakan kombinasi dari sekumpulan fungsi basis yang telah ditentukan sebelumnya. Fungsi basis adalah sekumpulan fungsi independen yang telah ditentukan sebelumnya yang kombinasinya mencakup semua fungsi yang diinginkan dalam domain tertentu. Fungsi estimasi akan dihitung sebagai kombinasi linier dari fungsi dasar. Fungsi dasar b_1 . . . b_K adalah fungsi yang ditentukan pada input x_1 . . . x_N dan kombinasi liniernya akan mencakup semua fungsi lain yang menarik untuk masalah pembelajaran mesin. Tujuannya adalah untuk mendapatkan koefisien K yaitu γ_1 . . . γ_K yang memberikan perkiraan terbaik untuk hubungan tersebut:

$$y = \begin{matrix} i = N \\ i = 1 \end{matrix} \gamma_i b_i (x_1 \ldots x_N)$$

Contoh umum dari pendekatan basis termasuk metode analisis wavelet dan model Campuran Gaussian.

Terlepas dari pendekatan yang digunakan untuk estimasi fungsi, hasil akhirnya adalah perhitungan angka K (atau parameter) yang diperlukan untuk memperkirakan fungsi yang tepat. Hampir semua algoritme pembelajaran mesin menghasilkan beberapa angka K yang mencirikan fungsi yang dipelajari untuk menangkap pola yang terdapat dalam data. Jika modelnya adalah jaringan saraf, parameter ini adalah bobot jaringan saraf. Jika modelnya adalah pohon keputusan, parameter ini memberikan jumlah node dan ambang batas percabangan pada node berbeda di pohon keputusan.

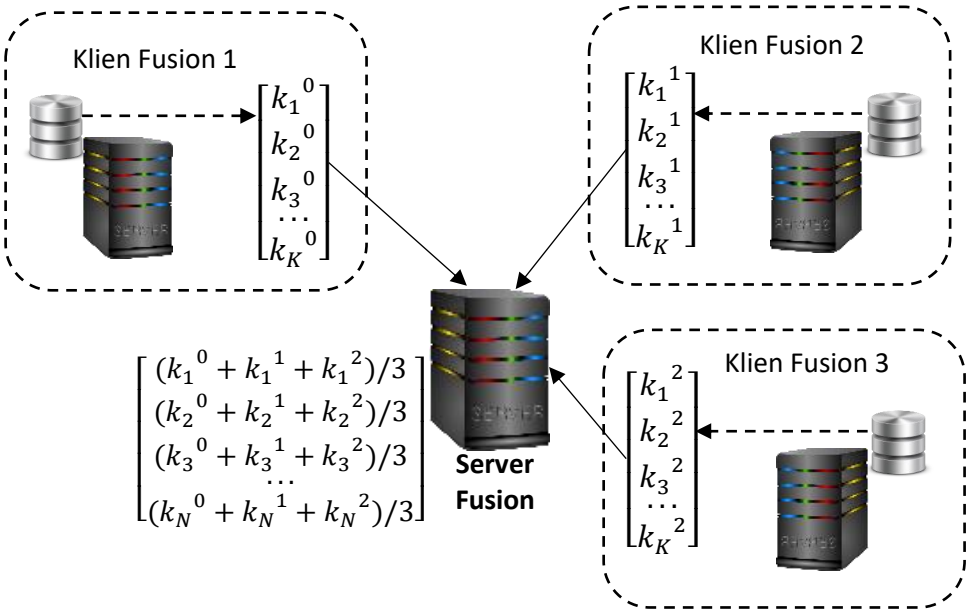
3.3 PEMBELAJARAN FEDERASI UNTUK ESTIMASI FUNGSI

Untuk menghadapi tantangan data terdistribusi, pendekatan pembelajaran gabungan akan memperkirakan fungsi secara independen di setiap klien fusi, dan kemudian menggabungkan semua fungsi di situs server fusi. Proses abstrak untuk fusion dapat dilihat pada Gambar 3.4 untuk tiga klien fusion. Setiap klien fusi memiliki beberapa data lokal. Data lokal klien fusi digunakan untuk menghitung angka K yang dapat memperkirakan model pada data lokal klien fusi.

Angka-angka ini dikirim ke server fusi, yang membuat rata-ratanya. Dalam komputasi perusahaan, hanya ada sejumlah kecil klien fusi yang bekerja dengan server fusi individual, dan alih-alih hanya membuat rata-rata model satu kali, masing-masing klien fusi menghitung angka K secara lokal dalam beberapa iterasi. Dalam setiap iterasi, subset acak dari data yang ada di klien fusi dipilih, dan angka K tersebut dihitung dan dikirim ke server fusi, yang membuat rata-rata nilai tersebut. Perulangan ini diulang sampai rata-rata telah dihitung berkali-kali.

Mari kita pahami alasan dasar mengapa rata-rata pada beberapa iterasi akan menghasilkan fungsi yang secara tepat menangkap pola dalam data di seluruh rangkaian klien fusi. Untuk tugas estimasi fungsi, asumsinya adalah terdapat fungsi ground truth yang dapat menangkap pola yang terdapat pada keseluruhan data. Subkumpulan data yang terdapat pada masing-masing klien fusi adalah bagian berbeda dari keseluruhan data. Ritme algoritma pembelajaran mesin di setiap klien fusi menganalisis sampel data yang sedikit berbeda, sehingga fungsi yang sedikit berbeda akan dipelajari. Namun, fungsi-fungsi ini merupakan variasi berbeda dari kebenaran dasar yang sama. Jika seseorang melakukan rata-rata pada ratusan variasi tersebut, kemungkinan besar bahwa rata-rata fungsi yang dihasilkan akan sama dengan kebenaran dasar.

Dalam lingkungan perusahaan, kami akan memiliki beberapa klien fusi, dan mempelajari fungsi dari semua data dan kemudian menghitung rata-ratanya bersama-sama mungkin tidak memberikan perkiraan yang baik tentang kebenaran dasar. Namun, jika kita dapat membuat subkumpulan data dari setiap klien fusi, membuat setiap subkumpulan memiliki ukuran yang masuk akal untuk mendapatkan perkiraan kebenaran dasar yang layak, dan kemudian membuat rata-rata perkiraan ini ratusan atau ribuan kali, kemungkinan besar kita akan mendapatkan ke kebenaran dasar.



Gambar 3.4 Pembelajaran gabungan untuk estimasi fungsi.

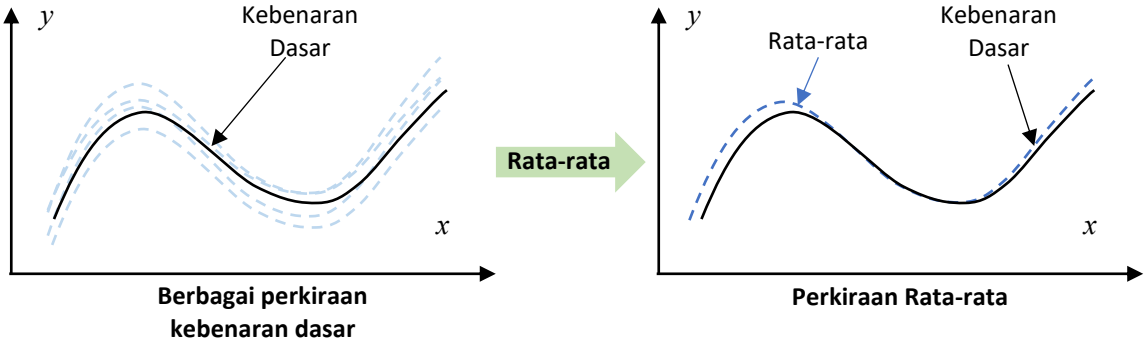
Alasan mengapa rata-rata estimasi fungsi dalam jumlah besar harus menghasilkan kebenaran dasar yang sebenarnya didasarkan pada pengamatan statistik yang disebut hukum bilangan besar. Hukum ini secara sederhana menyatakan bahwa jika seseorang melakukan percobaan dalam jumlah besar untuk mengukur nilai suatu variabel, maka rata-rata percobaan tersebut cenderung menjadi nilai yang diharapkan dari variabel tersebut. Hal ini berlaku untuk sebagian besar variabel dunia nyata yang mungkin ditemui dalam bisnis. Tugas pembelajaran mesin, yang pada intinya adalah mempelajari fungsi yang valid secara statistik, biasanya akan menghasilkan fungsi yang hukum bilangan besarnya mungkin berlaku. Namun perlu dicatat, seperti banyak hukum observasional, hukum ini lebih merupakan sebuah prinsip yang berlaku hampir sepanjang waktu, namun tidak selalu, yaitu, seseorang dapat menciptakan kasus-kasus patologis dimana hukum ini tidak berlaku.

Penerapan hukum bilangan besar pada tugas pembelajaran gabungan ditunjukkan pada Gambar 3.5. Di sisi kiri, beberapa perkiraan fungsi dibuat, dan perkiraan ini ditunjukkan dalam garis abu-abu berbeda yang ditunjukkan pada grafik. Garis gelap menunjukkan kebenaran dasar. Ketika sejumlah besar perkiraan dirata-ratakan, rata-rata yang dihasilkan akan cenderung mendekati kebenaran sebenarnya.

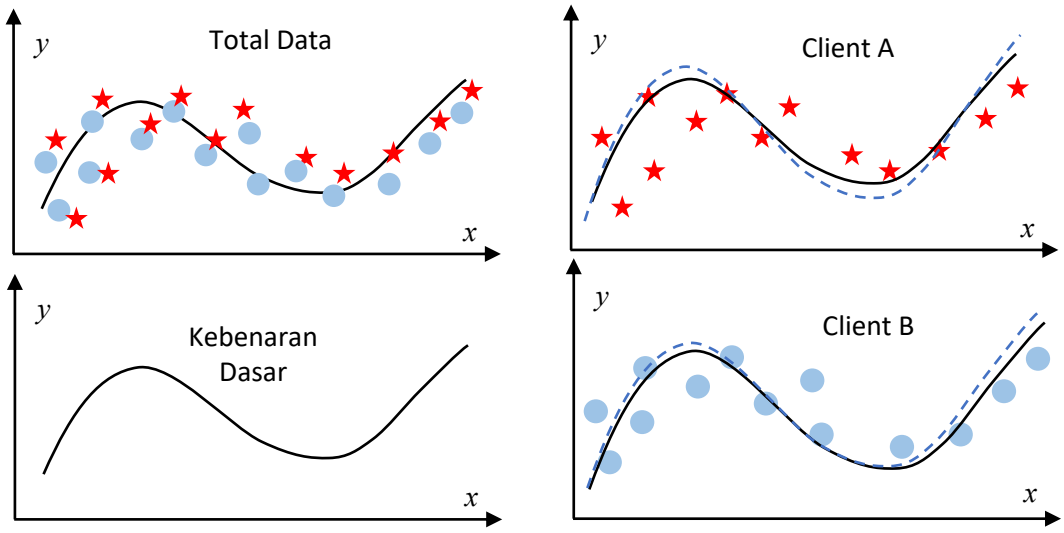
Contoh ilustrasi penggunaan proses rata-rata untuk dua klien fusi ditunjukkan pada Gambar 3.6. Gambar tersebut terdiri dari empat grafik, masing-masing memplot satu variabel masukan x dan variabel keluaran y. Grafik di kiri atas menunjukkan jumlah total data pelatihan yang tersedia, dengan bintang menandai data yang ada di klien fusi A, dan lingkaran menandai data yang ada di klien fusi B. Secara keseluruhan, bintang-bintang ini tandai kebenaran dasar yang ditunjukkan pada grafik kiri bawah. Grafik tersebut akan menjadi fungsi yang akan dipelajari jika semua data ada di satu lokasi.

Jika kedua klien fusi mempelajari fungsinya secara mandiri menggunakan algoritme pembelajaran mesin pilihan, mereka akan mempelajari fungsi yang agak berbeda. Fungsi yang

dipelajari masing-masing oleh klien A dan klien B ditunjukkan pada dua grafik di sebelah kanan. Garis abu-abu solid menunjukkan fungsi yang akan menjadi kebenaran dasar sebenarnya, sedangkan garis hitam putus-putus menunjukkan fungsi yang akan dipelajari sendiri oleh setiap klien fusi. Jika seseorang dapat menghitung rata-rata kedua fungsi tersebut, kemungkinan besar penyimpangannya akan hilang dan rata-ratanya akan menjadi sama dengan kebenaran dasar di sisi kiri bawah.



Gambar 3.5 Pendekatan rata-rata untuk pembelajaran gabungan.



Gambar 3.6 Pembelajaran gabungan untuk estimasi fungsi.

Rata-rata pada sejumlah kecil klien fusi (hanya 2 seperti yang ditunjukkan pada gambar) tidaklah cukup untuk menerapkan hukum bilangan besar dan meningkatkan kemungkinan penghapusan kesalahan. Namun, jika pemilihan acak subset data pada masing-masing klien fusi diambil sehingga fungsinya dipelajari beberapa ratus kali, proses rata-rata memiliki peluang yang sangat baik untuk menghilangkan kesalahan dan mendekati kebenaran dasar.

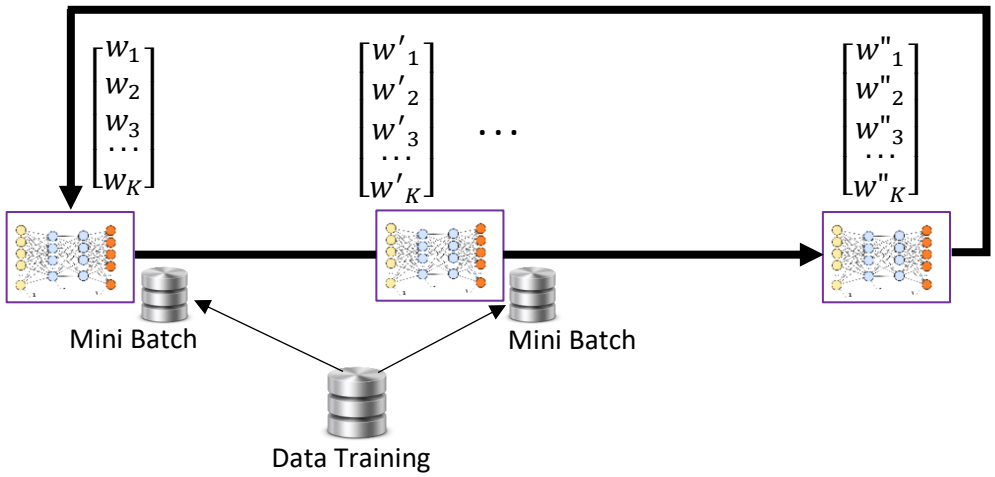
Rata-rata fungsi pada sejumlah besar sub-sampel mungkin terlalu memakan waktu sehingga tidak praktis dalam beberapa kasus. Namun, fungsi ini tidak bergantung pada jenis fungsi yang dipelajari dalam proses pembelajaran mesin. Mungkin ada beberapa optimasi

yang dapat dilakukan untuk mempercepat proses pembelajaran dan fusi dapat dilakukan lebih cepat. Salah satu pengoptimalan tersebut adalah mencoba meningkatkan parameter yang dipelajari secara bertahap, yang dapat diilustrasikan dengan baik dalam cara jaringan saraf dilatih.

3.4 PEMBELAJARAN FEDERASI UNTUK JARINGAN SYARAF TIRUAN

Jaringan saraf adalah pendekatan yang sangat populer untuk merepresentasikan fungsi umum yang diekstraksi dari sekumpulan data pelatihan. Parameter yang menjadi ciri jaringan saraf adalah bobot yang ditetapkan ke berbagai neuron dalam jaringan, dan pada dasarnya merupakan parameter K yang dijelaskan di Bagian 3.3. Daripada menggunakan subsampel acak dan merata-ratakan bobot yang dihasilkan, teknik yang lebih cepat yang dapat digunakan dalam proses pembelajaran adalah penyesuaian bobot model secara berulang.

Proses pelatihan model jaringan saraf ditunjukkan pada Gambar 3.7. Prosesnya dimulai dengan serangkaian bobot awal yang ditebak atau ditetapkan secara acak.

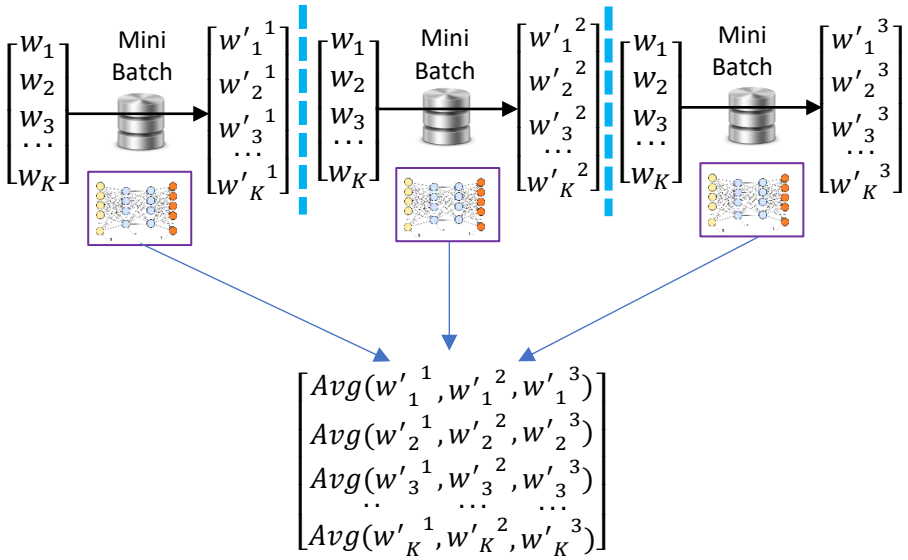


Gambar 3.7 Pelatihan berulang jaringan saraf.

Seluruh data pelatihan dibagi menjadi beberapa subset, yang disebut mini-batch. Bobot jaringan saraf digunakan untuk mengetahui seberapa berbeda prediksi jaringan yang menggunakan bobot tersebut dengan keluaran sebenarnya. Kemudian bobot disesuaikan agar kesalahan dapat dikurangi. Ada beberapa algoritme untuk penyesuaian bobot, namun membahas detail algoritme tersebut tidak penting pada tingkat penjelasan ini. Dengan bobot yang disesuaikan, proses ini diulangi untuk kumpulan data mini lainnya. Proses ini dapat diulangi dengan melakukan beberapa kali passing pada data, dan setiap pass akan mengurangi kesalahan antara prediksi dan nilai sebenarnya. Bobot yang dihitung pada akhir proses ini menentukan jaringan saraf terlatih yang akan digunakan untuk inferensi.

Variasi tersebut dapat dilihat sebagai penerapan pendekatan yang terdistribusi seperti dijelaskan pada Gambar 3.6. Keseluruhan data didistribusikan ke banyak klien fusi yang berbeda, misalnya. itu dibagi menjadi tiga klien fusi, seperti yang ditunjukkan pada Gambar 3.8. Setiap klien fusi akan melalui proses pelatihan bobot secara berulang, seperti yang

ditunjukkan pada Gambar 3.7. Namun, setelah masing-masing klien fusi menjalani satu pelatihan mini-batch secara independen, mereka akan mengirimkan bobot model ke server fusi dan bobot yang dihitung oleh semua klien fusi akan dirata-ratakan bersama. Masing-masing klien fusi kemudian akan memulai iterasi berikutnya dari mini-batch mereka dengan bobot rata-rata.



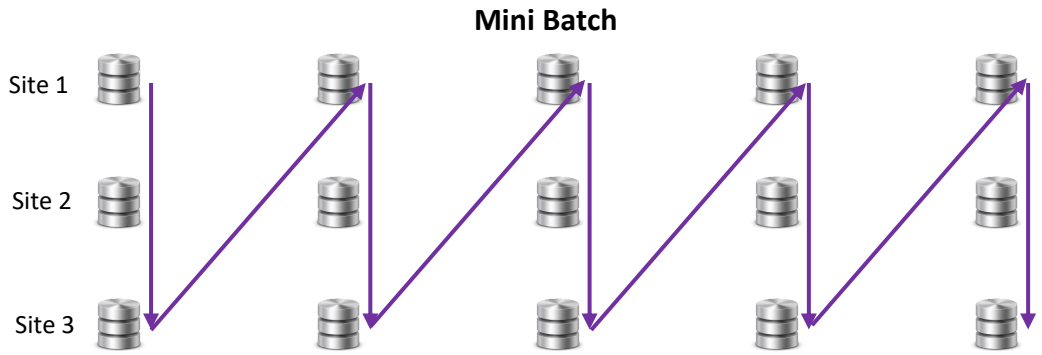
Gambar 3.8: Rata-rata gabungan untuk jaringan saraf.

Akibatnya, mini-batch yang seharusnya digunakan untuk satu klien fusi telah diganti dengan mini-batch yang dikumpulkan di seluruh klien fusi yang terlibat dalam proses pelatihan. Karena logika untuk melatih model ini sama dengan logika untuk melatih data di satu lokasi, proses fusi model pada akhirnya menghasilkan fungsi yang sama dengan yang dihasilkan oleh pendekatan terpusat. Selama semua mini-batch diproses dalam urutan tertentu, dan bergantung pada algoritme pembelajaran, semua data telah diperiksa satu kali atau lebih, proses pembelajaran kira-kira akan menghasilkan fungsi yang akan dipelajari dengan mengumpulkan semua data ke lokasi pusat.

Cara pemindaian kumpulan mini data yang berbeda pada klien fusi yang berbeda sesuai dengan algoritme yang dijelaskan pada Gambar 3.7 diilustrasikan pada Gambar 3.9. Jika data di setiap klien fusi dibagi menjadi beberapa mini-batch, mini-batch pertama dari masing-masing klien fusi diproses terlebih dahulu, kemudian algoritme pembelajaran gabungan memindai mini-batch kedua di setiap klien fusi dan melanjutkan ke tahap ini. cara sampai semua data dipindai. Data dapat dipindai beberapa kali jika diperlukan.

Ini bukan satu-satunya urutan di mana kami dapat memindai kumpulan mini untuk mencakup seluruh data di semua klien fusi. Daripada memindai array mini-batch dalam urutan vertikal, seseorang dapat memindai mini-batch dalam urutan horizontal. Urutan horizontal pemindaian mini-batch ditunjukkan pada Gambar 3.10. Dalam proses ini, semua batch mini

pada klien fusi tunggal dipindai terlebih dahulu, kemudian batch mini pada klien fusi kedua dipindai, dan seterusnya.



Gambar 3.9: Urutan pemindaian vertikal dari mini-batch.

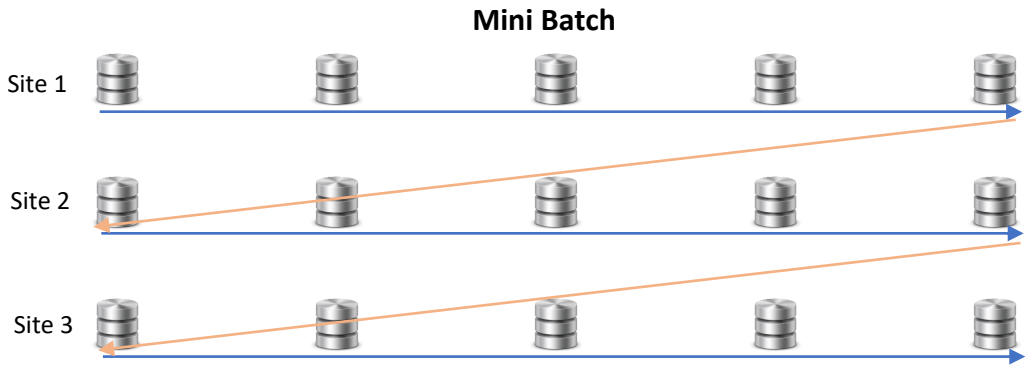
Dalam istilah praktis, pemindaian horizontal berarti bahwa seluruh jaringan saraf dilatih pada satu klien fusi, kemudian diteruskan ke klien fusi kedua untuk pelatihan, dan melewati semua klien fusi secara berurutan. Setelah melakukan beberapa lintasan melalui kumpulan semua klien fusi, lintasan pelatihan jaringan saraf menghasilkan perkiraan sistem yang sebanding dengan pelatihan terpusat. Pendekatan pemindaian horizontal kumpulan mini ini dapat menghasilkan pengurangan yang signifikan dalam frekuensi klien fusi menghubungi server fusi untuk menggabungkan model. Meskipun pemindaian vertikal asli dari mini-batch seperti yang ditunjukkan pada Gambar 3.9 mengharuskan setiap klien fusi untuk mengoordinasikan model mereka dengan server fusi pada setiap mini-batch, pemindaian horizontal memerlukan koordinasi hanya setelah semua data dipindai, yang menghasilkan pengurangan yang signifikan dalam jumlah pertukaran data antara klien fusi dan server fusi.

Pendekatan alternatif untuk pemindaian horizontal adalah meminta setiap klien fusi melatih semua model mereka secara paralel, sehingga setiap lapisan mini-batch horizontal dilintasi secara paralel. Ketika klien fusi telah menyelesaikan pelatihan, mereka dapat menghitung rata-rata bobot jaringan sarafnya. Proses ini harus diulangi beberapa kali agar hukum bilangan besar dapat berlaku, dan menjadikan keseluruhan jaringan saraf terlatih menjadi jaringan saraf yang dilatih secara terpusat.

Pemindaian horizontal dan vertikal bukan satu-satunya pilihan untuk melintasi kumpulan kecil data yang berbeda yang disimpan di masing-masing klien fusi. Klien fusi dapat memilih untuk berlatih dalam jumlah mini-batch yang berbeda, dan melakukan sinkronisasi pada waktu yang berbeda. Server fusi dapat mempertahankan bobot rata-rata jaringan saraf, dan membiarkan setiap klien fusi terhubung dengannya dan memperbarui hasil mini-batch saat ini pada interval yang berbeda. Rata-rata bobot dapat dilakukan dengan tingkat kepentingan berbeda yang diberikan kepada klien fusi berbeda, misalnya. jika klien fusi memiliki data dua kali lipat dari klien fusi lainnya, bobot modelnya dihitung dua kali saat dirata-ratakan terhadap bobot yang diberikan oleh semua klien fusi.

Pendekatan yang berbeda untuk melintasi mini-batch yang berbeda ini memunculkan jenis pembelajaran gabungan yang berbeda, yang dijelaskan dalam berbagai makalah seperti.

Terdapat perbedaan dalam konsumsi bandwidth jaringan, waktu konvergensi, dan beban pada server fusi yang ditentukan oleh algoritma yang berbeda, namun semuanya beroperasi dalam pendekatan keseluruhan yang dijelaskan di bagian ini.



Gambar 3.10: Urutan pemindaian horizontal dari mini-batch.

3.5 FEDERASI MODEL LAIN-LAIN

Rata-rata parameter dapat berfungsi di berbagai model AI jika hasilnya berupa sekumpulan angka yang masuk akal untuk membuat rata-rata parameter model yang dihasilkan. Namun, beberapa jenis model tidak direpresentasikan sedemikian rupa sehingga memungkinkan dilakukannya rata-rata sederhana.

Melihat kategori luas model AI yang dibahas di Bagian 1.5 Bab 1, kita telah membahas model fungsional dan jaringan saraf yang merupakan kandidat yang baik untuk melakukan rata-rata berulang di berbagai klien fusi. Beberapa model lainnya juga direpresentasikan sebagai sekumpulan parameter yang mewakili fungsi matematika. Dalam kasus tersebut, rata-rata gabungan akan bekerja dengan baik untuk menggabungkan jenis model ini bersama-sama. Secara khusus, Mesin Vektor Dukungan dapat dirata-ratakan bersama-sama dengan cara ini. Model AI yang direpresentasikan sebagai matriks, seperti Analisis Komponen Utama, juga dapat difederasi bersama menggunakan pendekatan rata-rata.

Beberapa model tidak mudah untuk dirata-ratakan jika mereka sudah terlatih sepenuhnya, misalnya pohon keputusan. Pohon keputusan memiliki beberapa titik percabangan dan, jika klien fusi melatih pohon keputusan mereka secara independen, pohon keputusan tidak dapat digabungkan secara langsung. Namun, algoritme pelatihan pohon keputusan yang ada membuat pohon keputusannya secara bertahap. Pada setiap tahap pelatihan pohon keputusan, pohon tersebut dibangun sebagian dan node baru dengan pengujian perlu dipilih. Pengujian tersebut memerlukan pemilihan salah satu variabel masukan $x_1 \ldots x_N$ yang akan digunakan untuk pengujian dan ambang batas untuk variabel. Pendekatan menumbuhkan pohon dari atas ke bawah akan memilih variabel dan ambang batas berdasarkan evaluasi nilai yang akan diperoleh oleh variabel dan ambang batas. Jika semua node memilih variabelnya, suara mayoritas dapat diambil di antara klien fusi yang berbeda untuk menentukan variabel berikutnya yang akan digunakan untuk mengembangkan pohon keputusan. Setelah variabel dipilih, ambang batas terbaik dapat disarankan oleh

masing-masing klien fusi dan rata-rata tertimbang untuk memilih ambang batas yang sesuai yang ditentukan oleh server fusi. Langkah individual dalam masing-masing titik keputusan untuk pertumbuhan tambahan ini akan mengikuti pendekatan gabungan untuk menghitung metrik seperti yang dijelaskan dalam Bagian 3.1.

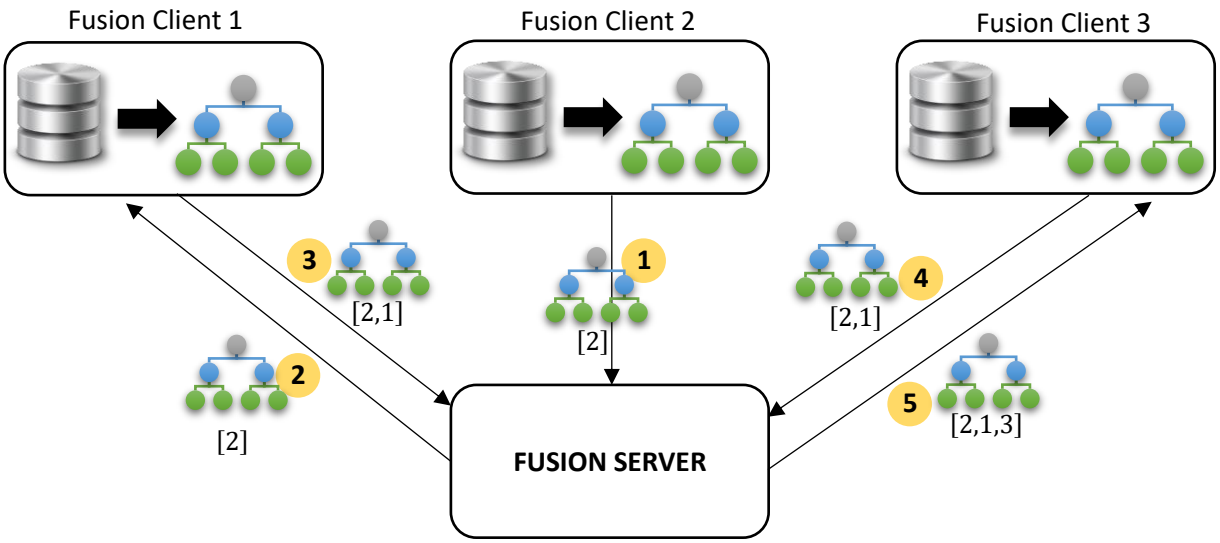
Beberapa model lain, seperti tabel keputusan dan aturan keputusan, mudah untuk difederasi. Semua entri dalam tabel keputusan dapat dimasukkan ke dalam satu tabel yang lebih besar dan dianalisis konflik antar area yang memiliki banyak entri yang bersesuaian. Ketidakkonsistenan kemudian dapat diidentifikasi menggunakan teknik yang dikenal dan dihilangkan dengan memilih berdasarkan prediksi mayoritas klien fusi, atau dengan inspeksi manusia.

Pembelajaran Inkremental Terdistribusi

Beberapa algoritme pelatihan untuk berbagai jenis model bersifat inkremental, dalam artian algoritme tersebut mengikuti paradigma untuk mempelajari suatu model dan kemudian menyempurnakan model tersebut menggunakan data tambahan. Pendekatan pelatihan pada mini-batch untuk meningkatkan bobot jaringan saraf adalah contoh proses tambahan yang meningkatkan jaringan saraf yang sudah ada. Sama seperti jaringan saraf yang dapat dilatih secara bertahap, jenis model lainnya juga dapat dilatih secara bertahap. Algoritma yang melakukan pembelajaran tambahan dikenal dengan berbagai estimasi fungsi yang mencoba mengoptimalkan beberapa atribut, kumpulan aturan, tabel keputusan, pohon keputusan dan mesin vektor pendukung.

Jika algoritme pelatihan bersifat inkremental, algoritme tersebut memberikan pendekatan alternatif untuk melakukan pembelajaran gabungan. Jika masing-masing klien fusi memiliki kemampuan untuk melatih model yang sama secara bertahap, server fusi dapat mengatur agar klien fusi melatih model secara berurutan. Urutan pelatihan acak di antara klien yang berbeda dapat ditentukan, model dilatih pada klien fusi pertama di situs yang kemudian mengunggah model ke server fusi. Server fusi mengirimkan model ke klien kedua secara berurutan, yang melatih model lebih lanjut menggunakan algoritma tambahan. Model dikirim ke klien ketiga dalam pesanan. Proses ini diulangi hingga model dilatih di semua situs. Proses ini dapat diulangi hingga beberapa lintasan telah dilakukan melalui semua lokasi. Mengulangi proses pelatihan melalui semua data berulang kali membantu beberapa model untuk meningkatkan dirinya sendiri.

Proses memanfaatkan klien untuk melatih model secara berurutan menggunakan pembelajaran tambahan ditunjukkan pada Gambar 3.11. Tiga klien fusi ditampilkan bersama dengan server fusi. Server fusi memilih urutan acak untuk menjalankan model melalui klien fusi dalam urutan. Klien fusi 2 akan menggunakan data lokalnya untuk melatih instance pertama model yang dikirim pada langkah 1 ke server fusi setelah pelatihan selesai. Server fusi kemudian mengirimkannya ke klien fusi 1 yang melatih model secara bertahap menggunakan datanya. Model hasil yang sekarang dilatih dari data dari klien fusi 2 dan 1 dikirim kembali ke server fusi. Server fusi kemudian akan mengirimkan model tersebut ke klien fusi 3, yang akan menggunakan algoritme pelatihan tambahannya untuk memperbarui model yang sekarang memiliki semua data dari tiga situs yang digunakan untuk melatihnya.



Gambar 3.11: Pembelajaran tambahan yang terdistribusi.

Salah satu kelemahan proses yang ditunjukkan pada Gambar 3.11 adalah hanya satu klien fusi yang secara aktif melatih model pada waktu tertentu. Hal ini tidak menggunakan jumlah paralelisme yang signifikan yang tersedia dalam sistem dan tidak menggunakan kapasitas komputasi yang tersedia pada klien fusi yang berbeda secara efisien. Untuk mengatasi keterbatasan tersebut, server fusi dapat mengatur proses pelatihan di klien sehingga semua N model dilatih secara paralel, setiap model dimulai dengan data dari salah satu N klien fusi. Urutan acak dapat ditentukan oleh server fusi sehingga masing-masing klien fusi melatih model secara paralel alih-alih mengeksekusinya secara berurutan, seperti yang ditunjukkan pada Gambar 3.11. Setelah urutan acak ditentukan, server fusi dapat membuat jadwal bergilir untuk menukar model yang dilatih dengan situs berbeda.

Dengan menggunakan pendekatan ini, misalkan server fusi menentukan urutan permutasi menjadi [2, 1, 3]. Ia dapat menggeser urutannya untuk membuat dua jadwal lagi yaitu [1, 3, 2] dan [3, 1, 2]. Pelatihan paralel model terjadi seperti yang ditunjukkan pada Tabel 3.1. Kami memberi label pada model A, B, dan C yang awalnya akan dilatih pada klien fusi 1, 2, dan 3 seperti yang ditunjukkan pada baris pertama tabel ini. Setelah langkah pelatihan ini selesai, model A dilatih di klien 3, model B di klien 1, dan model C di klien 2. Setelah selesainya langkah kedua ini, pada langkah ketiga, model A dilatih di klien 2, model B di klien 3 dan model C di klien 1. Pada setiap langkah, setiap model dilatih di salah satu klien fusi. Di akhir proses, ketiga model masing-masing telah dilatih pada ketiga sumber data dalam urutan berbeda. Siklus yang ditunjukkan pada tabel dapat diulang jika diperlukan.

Mengingat sifat stokastik dari pelatihan model AI, ketiga model tersebut akan memiliki beberapa perbedaan dan tidak memberikan jawaban yang persis sama untuk semua masukan. Namun, pada akhir proses pelatihan, model ini dapat disatukan menjadi sebuah ansambel dan digunakan oleh semua klien fusi selama fase inferensi dari siklus Learn→Infer→Act.

Tabel 3.1: Pelatihan model paralel dengan algoritma pelatihan tambahan.

Melangkah	Model A	Model B	Model C
1	Klien 1	Klien 2	Klien 3
2	Klien 3	Klien 1	Klien 2
3	Klien 2	Klien 3	Klien 1

3.6 ASUMSI DALAM PEMBELAJARAN FEDERASI YANG NAIF

Meskipun pendekatan berbeda untuk pembelajaran gabungan yang dijelaskan di Bagian 3.4 tampaknya bagus dari perspektif pelatihan model dari data yang didistribusikan ke banyak klien fusi yang berbeda, semua pendekatan ini bergantung pada beberapa asumsi umum. Asumsi-asumsi ini tidak harus dipenuhi dalam konteks hadiah di kehidupan nyata atau dapat menyebabkan beberapa tantangan dalam logistik dan pengoperasiannya. Pada bagian ini kita memeriksa beberapa asumsi tersebut.

Algoritme dalam bab ini berasumsi bahwa data pelatihan yang digunakan pada klien fusi yang berbeda konsisten di semua lokasi dan dapat digunakan untuk melatih model umum. Jika data tidak konsisten di seluruh lokasi, misalnya lokasi yang berbeda memiliki jenis masukan yang berbeda, atau masukan tidak diskalakan secara konsisten, maka parameter model tidak dapat dirata-ratakan.

Dalam praktiknya, faktor-faktor yang menghambat perpindahan data juga merupakan faktor yang menyebabkan sejumlah besar inkonsistensi di antara berbagai situs. Jika data tidak diperbolehkan berpindah ke situs klien fusi yang berbeda, data tersebut diatur oleh orang yang berbeda atau tunduk pada pedoman yang berbeda. Dalam beberapa kasus, data tersebut mungkin dikumpulkan secara independen. Akibatnya, data kemungkinan besar tidak konsisten di berbagai lokasi klien fusi.

Berbagai ketidakcocokan yang ada di antara kumpulan data klien fusi yang berbeda, masalah yang dapat ditimbulkannya, dan pendekatan untuk mengatasinya akan dieksplorasi lebih dalam di Bab 4.

Algoritme Naif berasumsi bahwa data yang ada tidak miring. Kemiringan data terjadi ketika klien fusi yang berbeda mengumpulkan data dengan karakteristik yang sangat berbeda. Sebagai contoh, salah satu klien fusi mungkin telah mengumpulkan banyak data yang menangkap perilaku pembelian masyarakat di lingkungan perkotaan. Klien fusi lain mungkin telah mengumpulkan data yang mungkin mewakili perilaku pembelian masyarakat di lingkungan pedesaan. Kita dapat mempelajari jaringan atau fungsi saraf untuk memprediksi karakteristik pembelian suatu populasi dengan merata-ratakan perilaku di kedua lingkungan, namun merata-ratakannya saja mungkin bukan cara yang tepat untuk menggabungkan kedua model tersebut. Ini adalah dua fungsi berbeda yang sedang dipelajari, dan asumsi yang mendasari rata-rata gabungan tidak valid dalam konteks ini.

Asumsi yang tersirat dalam pendekatan pembelajaran gabungan yang naif adalah bahwa semua klien fusi mempelajari fungsi yang sama. Jika data di semua klien fusi memiliki fungsi umum yang sama dengan yang sedang diestimasi, rata-rata parameter dalam urutan apa pun pada akhirnya akan menghasilkan fungsi yang sama. Ketimpangan data akan

menyebabkan pelanggaran terhadap asumsi ini. Fungsi yang dipelajari pada klien fusi yang berbeda mungkin berbeda jika mereka memelihara datanya secara berbeda, dan telah mengumpulkan data tentang jenis lingkungan yang berbeda. Tantangan yang disebabkan oleh ketimpangan data dan pendekatan untuk mengatasinya dijelaskan di Bab 5.

Asumsi tersirat dalam algoritme yang disajikan di bagian ini adalah bahwa klien fusi dan server fusi saling percaya untuk berperilaku dengan cara yang benar. Server fusi dan klien fusi semuanya berfungsi dengan baik dan semua model yang dipertukarkan dipelihara sesuai dengan kebijakan dan pedoman yang dapat diterima semua orang. Ketika pembelajaran gabungan terjadi di lingkungan yang melibatkan lebih dari satu organisasi, kepercayaan di antara organisasi yang berbeda mungkin tidak bersifat mutlak. Dalam kasus tersebut, mekanisme mungkin perlu untuk memastikan bahwa masalah kepercayaan yang terbatas dapat diatasi dan bahwa infrastruktur dan algoritma untuk pembelajaran gabungan dapat berfungsi dalam lingkungan ini. Skenario untuk mengatasi batasan kepercayaan yang berbeda dan pendekatan untuk mengatasinya akan dibahas lebih lanjut di Bab 6.

Asumsi lain dalam algoritma yang disajikan dalam bab ini adalah bahwa semua situs fusi aktif pada waktu yang sama dan dilatih dalam langkah kunci yang tersinkronisasi. Pemindaian vertikal mini-batch yang dijelaskan pada Gambar 3.9 hanya dapat dilakukan jika semua klien fusi aktif pada waktu yang sama. Namun, dalam banyak skenario dunia nyata, pendekatan model pelatihan yang lebih asinkron mungkin diperlukan. Model dari lokasi berbeda mungkin tersedia pada waktu berbeda.

Implikasi dari asumsi sinkronisasi yang dibuat dalam algoritme adalah bahwa model di setiap klien fusi memiliki arsitektur yang sama. Masing-masing klien fusi mempelajari beberapa parameter K tetapi parameter K ini harus sama di setiap klien fusi, dan memiliki semantik yang sama. Jika kita melatih jaringan saraf dengan beberapa L lapisan neuron dan beberapa N neuron di setiap lapisan, maka $K = L.N$. Namun, ini berarti bahwa setiap klien fusi harus melatih jaringan saraf dengan jumlah lapisan yang sama dan jumlah neuron yang sama di setiap lapisan. Artinya, semua klien fusi harus menyetujui arsitektur umum untuk model sebelum pelatihan.

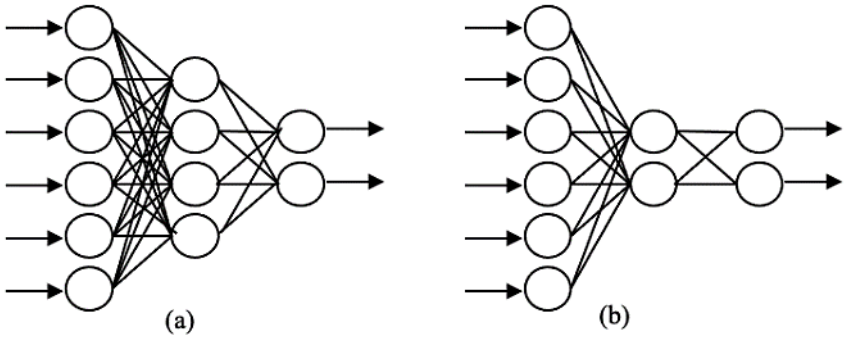
Untuk memahami alasan memerlukan arsitektur umum, mari kita periksa dua jaringan saraf berbeda yang dapat dilatih untuk melakukan tugas yang sama, misalnya. pelajari dari atribut permohonan pinjaman apakah pinjaman tersebut termasuk dalam salah satu dari empat kategori risiko. Risiko di sini adalah perkiraan kemungkinan peminjam tidak mampu membayar kembali pinjamannya. Banyak jenis jaringan saraf yang berbeda dapat dilatih untuk melakukan hal yang sama, dan dua contoh sederhana jaringan saraf yang dapat mengklasifikasikan atribut masukan dan keluaran suatu keputusan ke dalam salah satu dari empat kategori ditunjukkan pada Gambar 3.12. Perhatikan bahwa contoh sederhana ini hanya untuk ilustrasi, jaringan saraf dunia nyata akan memiliki lebih banyak lapisan dan lebih banyak atribut dan neuron per lapisan.

Pada jaringan saraf (a) yang ditunjukkan pada gambar, terdapat 7 node masukan, lapisan tengah sebanyak 4 node, dan lapisan keluaran sebanyak 2 node. Pada jaringan (b), jumlah node masukan dan node keluaran sama, namun tingkat menengah hanya mempunyai

2 node. Dengan asumsi bobot dikaitkan dengan masing-masing tautan masukan (atau neuron) untuk setiap node, jaringan saraf (a) akan memiliki bobot 4*7 + 4*2 = 36, sedangkan jaringan saraf (b) akan memiliki 2*7 + 2*2 atau 18 beban. Oleh karena itu, jaringan saraf (a) akan ditandai dengan 36 angka, sedangkan jaringan saraf (b) akan ditandai dengan 18 angka. Namun, kedua jaringan ini dapat dilatih untuk melakukan tugas yang sama persis, memetakan masukan dengan 7 atribut ke keluaran yang mengambil empat kemungkinan nilai.

Jika selama proses pembelajaran gabungan antara 2 klien fusi, salah satu klien memutuskan untuk menggunakan jaringan saraf (a) untuk melatih datanya, dan klien fusi lainnya memutuskan untuk menggunakan jaringan saraf (b) untuk melatih datanya, maka klien tersebut dua bobot jaringan saraf tidak dapat digabungkan bersama. Untuk melakukan rata-rata parameter dan menggabungkan model, jaringan di kedua klien fusi harus memiliki arsitektur yang sama.

Dalam konteks pembelajaran gabungan konsumen, di mana model dilatih oleh contoh berbeda dari aplikasi seluler yang sama, asumsi seperti itu bukannya tidak masuk akal. Setiap ponsel akan menggunakan datanya untuk melatih model menggunakan arsitektur yang ditentukan oleh pengembang aplikasi seluler. Namun, dalam konteks pembelajaran gabungan perusahaan, data disimpan di beberapa lokasi yang mungkin berada di bawah yurisdiksi administratif atau peraturan yang berbeda. Hal ini membuat tugas koordinasi untuk memastikan bahwa setiap klien fusi memiliki arsitektur model yang sama menjadi relatif lebih kompleks. Jika perusahaan memiliki bagian yang diperoleh melalui akuisisi, perusahaan yang diakuisisi mungkin menggunakan jenis model yang berbeda untuk mengambil keputusan. Model yang telah dilatih sebelumnya tersebut tidak dapat digunakan untuk pembelajaran gabungan secara langsung, dan bagaimanapun juga, semua bagian perusahaan harus menyetujui arsitektur umum yang sama.



Gambar 3.12 Dua jaringan saraf alternatif untuk tugas yang sama.

Kesepakatan mengenai model umum ini bisa sangat sulit tergantung pada struktur perusahaan. Asumsi kedua yang dibuat dalam algoritme adalah bahwa semua klien fusi harus berlatih pada waktu yang sama. Pendekatan pembelajaran gabungan mengharuskan bobot model dipelajari dan dilatih di semua klien fusi pada saat yang bersamaan. Hal ini mengharuskan klien fusi memulai pelatihan pada atau sekitar waktu yang sama, menukar bobot model secara berkala, dan melanjutkan bagian pelatihan secara tersinkronisasi. Beberapa masalah sinkronisasi dapat diatasi dengan kedatangan klien fusi dan memperbarui

bobot model agregat yang disimpan di server. Mengingat fakta bahwa berbagai silo perusahaan yang berisi data terdistribusi mungkin dikelola secara independen, koordinasi seperti itu akan menjadi tantangan. Dalam beberapa kasus, kita dapat berasumsi bahwa pemicu dari server fusi dapat memulai proses pelatihan. Pemicu ini dapat dibuat ketika klien fusi saling percaya, atau setidaknya klien fusi yang memicu. Namun, dalam kondisi seperti konsorsium atau koalisi militer, proses pemicuan yang terkoordinasi seperti itu mungkin tidak dapat dilakukan.

Bukanlah hal yang tidak beralasan jika bagian organisasi yang semi-independen atau independen melatih model mereka pada waktu yang berbeda. Sekalipun terdapat kesesuaian wajib atau sukarela dengan arsitektur model umum, hal ini berarti model dilatih pada waktu yang berbeda. Menggabungkan model yang dilatih secara independen dengan membuat rata-rata satu kali tidak akan berhasil karena hukum bilangan besar tidak dapat diterapkan. Hal ini membuat tugas pelatihan pada waktu yang berbeda menjadi sangat menantang bagi sistem. Isu-isu ini dieksplorasi secara lebih rinci di Bab 7.

Partisi data muncul ketika data tentang individu yang sama dikumpulkan untuk tujuan berbeda, sehingga entitas yang dipantau di satu situs memiliki fitur dan atribut yang sangat berbeda dibandingkan fitur dan atribut yang ada pada entitas tersebut di situs lain. Jika kita melihat berbagai perusahaan yang aktif di suatu wilayah, mereka menyediakan layanan kepada populasi yang sama. Namun, mengingat sifat bisnisnya, mereka mengumpulkan berbagai jenis informasi tentang orang-orang di wilayah layanan mereka.

Data yang dipartisi mematahkan asumsi bahwa model di klien fusi yang berbeda adalah sama. Ketika atribut data yang berbeda ada, model yang berbeda akan dibuat. Model-model ini tidak dapat dirata-ratakan secara membabi buta, namun masuk akal untuk menggunakan model-model yang berbeda secara bersamaan, jika memungkinkan. Pendekatan yang berbeda dapat digunakan untuk menggabungkan wawasan dari model yang berbeda, termasuk pendekatan yang dapat mengubahnya menjadi model yang konsisten dan pendekatan yang hanya menggunakan model yang tidak konsisten selama fase inferensi. Pendekatan-pendekatan ini dibahas secara lebih rinci di Bab 8.

3.7 RINGKASAN

Dalam bab ini, kita telah meninjau pendekatan dasar untuk pembelajaran gabungan menggunakan rata-rata berulang. Pendekatan ini berhasil ketika klien fusi yang berbeda mencoba membangun model bersama-sama, memiliki data yang didistribusikan secara relatif seragam, saling percaya, dan dapat melakukan sinkronisasi satu sama lain untuk melatih model. Dalam lingkungan dunia nyata, banyak asumsi yang tidak terpenuhi. Dalam beberapa bab berikutnya, kita akan melihat beberapa pendekatan untuk memodifikasi dan mengadaptasi pendekatan naif sehingga tantangan dalam pembelajaran gabungan perusahaan dapat diatasi secara memadai.

BAB 4
MENGATASI MASALAH KETIDAKCOCOKAN DATA DI AI FEDERASI

Salah satu tantangan besar bagi AI gabungan adalah kenyataan bahwa kumpulan data yang disimpan di lokasi fisik yang berbeda cenderung memiliki format yang berbeda dan tidak konsisten, memiliki kualitas yang berbeda-beda, dan menggunakan istilah yang berbeda untuk konsep yang sama. Untuk melatih model menggunakan pembelajaran gabungan, atau untuk menggunakan model menggunakan inferensi gabungan, semua data di semua klien fusi harus dikonversi ke format tunggal yang konsisten. Dalam bab ini, kita melihat beberapa teknik yang dapat digunakan untuk mencapai tujuan ini. Kecuali jika ketidakcocokan data pada klien fusi yang berbeda ini diselesaikan, AI gabungan tidak dapat digunakan secara bermakna baik dalam fase pembelajaran atau inferensi dari Siklus Pelajari→Infer→Bertindak.

Secara umum, data yang digunakan untuk pembelajaran mesin di klien fusi mana pun dapat berupa data berfitur atau mentah (yaitu tanpa fitur). Data mentah biasanya terdiri dari teks tidak terstruktur, gambar, video, suara, ucapan, pembacaan sensor, dll., yang biasanya merupakan representasi digital dari sinyal berkelanjutan atau dokumen bahasa alami. Data unggulan biasanya berupa informasi yang terdiri dari beberapa fitur yang digambarkan dengan jelas, misalnya. data pinjaman bank, catatan manajemen kesehatan, transaksi penjualan ritel, dll. Data unggulan biasanya direpresentasikan dalam format tabel seperti spreadsheet atau database relasional, atau dalam format teks terstruktur. Beberapa contoh format teks terstruktur adalah notasi objek javascript atau JSON dan bahasa markup yang diperluas atau XML.

Selama pembelajaran mesin, langkah awal adalah konversi data mentah atau tanpa fitur menjadi serangkaian fitur, yang selanjutnya digunakan untuk melatih model AI.Dalam beberapa kasus, ekstraksi fitur dapat dilakukan secara eksplisit, misalnya. lihat untuk beberapa teknik mengekstraksi fitur dari gambar. Dalam kasus lain, khususnya saat menggunakan jaringan saraf dalam, beberapa lapisan awal jaringan saraf mengekstrak fitur secara otomatis sebagai bagian dari proses pelatihan. Dalam kasus tersebut, jaringan saraf itu sendiri dapat dipandang terdiri dari dua sub-jaringan, satu melakukan ekstraksi fitur dan yang lainnya mengubah fitur menjadi keluaran yang diinginkan.

Terkadang, data pelatihan berisi keluaran yang dihasilkan dari kombinasi fitur. Kumpulan data pelatihan tersebut disebut berlabel, karena berisi label yang tepat yang terkait dengan setiap kumpulan data dalam masukan pelatihan. Beberapa pendekatan pembelajaran hanya bekerja dengan data berlabel sementara pendekatan pembelajaran lainnya tidak memerlukan data untuk diberi label.

Ketidakcocokan data di seluruh klien fusi dapat terjadi dalam berbagai bentuk. Berikut ini adalah beberapa masalah ketidakcocokan data yang umum:

- ***Perbedaan format***: Data pada klien fusi yang berbeda mungkin memiliki format yang berbeda. Untuk menciptakan model gabungan, perbedaan-perbedaan dalam format ini perlu direkonsiliasi. Jika datanya mentah atau tanpa fitur, perbedaan ini dapat

terlihat ketika data direpresentasikan dalam format yang berbeda. Saat data ditampilkan, beberapa fitur mungkin hilang dalam data di beberapa klien fusi. Fitur juga dapat diberi nama berbeda pada klien fusi yang berbeda, atau dicatat dengan cara berbeda. Sebagai contoh, fitur bernama Nama dapat dicatat mengikuti konvensi 'Nama Depan Nama Belakang' di satu klien fusi, sementara fitur tersebut dapat dicatat sebagai 'Nama Belakang, Nama Depan' di klien fusi lainnya. Perbedaan-perbedaan ini perlu direkonsiliasi sebelum proses pembelajaran mesin dapat dimulai.

- ***Perbedaan Nilai:*** Saat data masukan ditampilkan, nilai yang diambil fitur berbeda pada klien fusi berbeda belum tentu sama. Satu klien fusi mungkin memiliki data yang menilai risiko pinjaman dengan kode warna seperti kuning, biru, dan hijau, sementara klien fusi lainnya mungkin memiliki risiko yang dinilai secara numerik sebagai 1, 2, dan 3. Salah satu masalah khusus mengenai perbedaan nilai yang umum terjadi pada data unggulan dan data mentah muncul ketika data diberi label, dan nilai yang ditetapkan pada label berbeda pada klien fusi yang berbeda. Misalnya, jika data gambar telah dikumpulkan, lokasi manufaktur di Perancis mungkin telah memberi label pada gambar produk manufaktur yang berbeda dalam bahasa Prancis, sedangkan pabrik manufaktur di Amerika pada perusahaan yang sama mungkin memiliki label dalam bahasa Inggris. Skema pelabelan yang konsisten perlu dikembangkan di seluruh klien fusi untuk menciptakan model yang bermakna.
- ***Perbedaan Kualitas:*** Klien fusi yang berbeda mungkin memiliki data dengan kualitas yang berbeda. Untuk membangun model yang baik, data dari klien fusi yang mungkin memiliki kualitas lebih buruk dari rata-rata perlu dihilangkan. Akan berguna jika hanya menggunakan data yang memiliki tingkat kualitas minimum untuk membangun model. Untuk menangani aspek ini, penting untuk memiliki definisi kualitas data, dan pendekatan untuk menghitung kualitas data.
- ***Partisi data:*** Data yang tersedia di lokasi berbeda mungkin dipartisi dengan cara berbeda. Partisi data dapat dijelaskan dengan baik dalam bentuk data unggulan dengan label. Anggaplah data unggulan disimpan dalam tabel yang kolomnya adalah nama fitur dengan kolom terakhir sebagai keluarannya, dan barisnya diurutkan menurut labelnya. Partisi tersebut kemudian dapat didefinisikan sebagai partisi horizontal atau vertikal. Dalam partisi data vertikal, beberapa fitur mungkin hilang dari data di beberapa klien fusi, yaitu beberapa kolom representasi tabel hilang. Dalam partisi data secara horizontal, beberapa label mungkin hilang dari data pada klien fusi yang berbeda, yaitu blok baris dengan label tertentu hilang dari data pada beberapa klien fusi. Kedua jenis partisi ini menimbulkan tantangan dalam penggabungan model AI yang dilatih pada klien fusi yang berbeda.

Di bagian berbeda bab ini, kami mengeksplorasi pendekatan untuk menyelesaikan perbedaan ini guna menciptakan model yang dapat digabungkan satu sama lain di server fusi. Masalah Pemartisian masing-masing dibahas di Bab 5 dan 8.

4.1 MENGONVERSI KE FORMAT INPUT UMUM

Pendekatan untuk mengonversi ke format umum berbeda-beda bergantung pada jenis data yang tersedia untuk pelatihan. Jika tipe datanya mentah, konversi adalah tugas mengubah format data, misalnya. mengkonversi gambar dari format JPEG ke format TIFF. Jika tipe data diunggulkan, konversi mungkin memerlukan perubahan nama fitur, atau penyesuaian nilai fitur. Selain itu, jika ada normalisasi yang harus dilakukan pada fitur, normalisasi ini harus dilakukan secara konsisten di semua klien fusi.

4.1.1 Tipe Data Mentah

Mari kita pertimbangkan contoh ketika klien fusi yang berbeda memiliki data mentah (yaitu tanpa fitur eksplisit), dan data dipertahankan dalam format berbeda pada klien fusi yang berbeda. Meskipun datanya mungkin bertipe sama, ada banyak format untuk menyimpan setiap tipe data.

Beberapa tipe data umum dan format penyimpanannya pada klien fusi yang berbeda ditunjukkan pada Tabel 4.1. Mengingat banyaknya pilihan untuk tipe data yang sama, tidak masuk akal untuk mengharapkan bahwa beberapa klien fusi mungkin memilih format yang berbeda.

Tabel 4.1: Format umum untuk beberapa tipe data mentah.

Tipe data	Format
Gambar-gambar	jpeg, tiff, eps, png, svg, bitmap
video	film quicktime (.mov), video windows media (.wmv), mpeg 3(.mp3), video flash (.flv)
Kedengarannya	wav, mp3, webm
File Teks	format teks kaya, microsoft word, pdf, teks biasa, dokumen terbuka (.odt)

Tabel 4.1 hanya mencantumkan beberapa tipe tipe data mentah dan beberapa format umum untuk mewakili masing-masing tipe data tersebut. Masih banyak lagi contoh untuk masing-masing tipe data mentah, mis. banyak format gambar lain selain enam yang tercantum dalam tabel ditemukan di berbagai situs. Namun, masalah dalam menciptakan pendekatan AI gabungan tetap sama, klien fusi yang berbeda mungkin memiliki data mentah dalam berbagai format berbeda. Namun, pembuatan model gabungan yang umum mengharuskan data berada dalam format tunggal di semua klien fusi.

Pendekatan untuk memilih format adalah dengan memilih format yang memiliki jumlah data maksimum yang tersedia, dan tidak menggunakan data dalam format lain. Pendekatan ini menghindari tugas konversi format. Namun, hal ini mungkin akan mengecualikan sejumlah besar data yang tersedia. Dalam pembelajaran mesin, data sangatlah berharga, dan memperluas kumpulan data pelatihan yang tersedia adalah alasan paling penting yang mendorong pembelajaran gabungan. Tidak menggunakan data yang tersedia bukanlah pilihan yang baik jika tujuannya adalah membangun model AI yang baik.

Konversi data dalam format lain ke format umum yang dipilih dapat dilakukan melalui perangkat lunak konversi format. Banyak paket perangkat lunak konversi format yang tersedia, beberapa dalam bentuk sumber terbuka, beberapa sebagai perangkat lunak

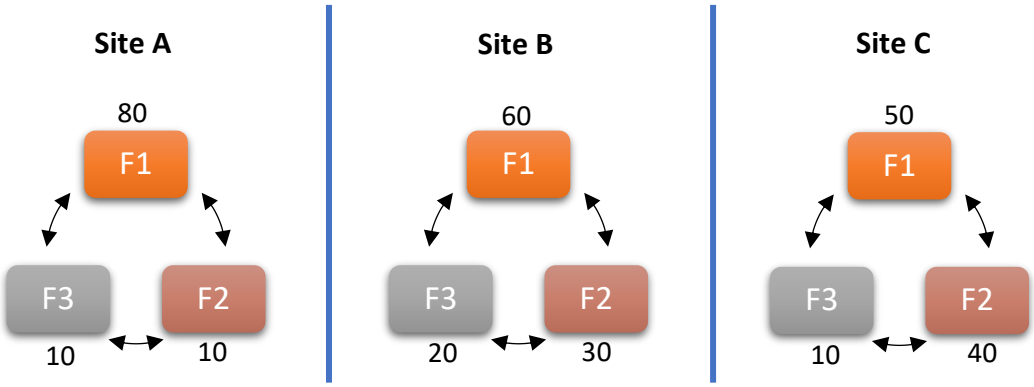
berpemilik, dan beberapa sebagai layanan berbasis Internet. Ketersediaan kemampuan konversi perangkat lunak untuk klien fusi yang berbeda dapat digunakan untuk memilih format umum yang tepat untuk digunakan dalam pembuatan model.

Untuk menentukan format yang tepat, setiap klien fusi dapat membuat grafik konversi. Grafik konversi adalah grafik yang simpulnya mewakili salah satu format data yang tersedia, dan tepi terarah menghubungkan dua simpul jika kemampuan konversi format searah panah tersedia di klien fusi. Setiap klien fusi akan memiliki grafik konversi berbeda yang ditentukan oleh ketersediaan paket perangkat lunak dan layanan yang tersedia untuk melakukan konversi ke klien fusi berbeda. Grafik konversi ini, bersama dengan jumlah total data yang tersedia di setiap format secara lokal, dilaporkan oleh setiap klien fusion ke server fusion. Dengan grafik konversi semua klien fusi, server fusi dapat mengidentifikasi format yang akan memberikan model terbaik di seluruh klien fusi.

Jika biaya konversi data dapat diabaikan, format yang tepat untuk dipilih adalah format yang menghasilkan jumlah data maksimum yang tersedia untuk membuat model. Untuk setiap grafik konversi, metrik dapat dihitung berdasarkan jumlah total data yang tersedia di setiap situs fusi berdasarkan grafik konversi. Contoh ilustratif ditunjukkan pada Gambar 4.1. Ini mengasumsikan bahwa ada tiga klien fusi dan masing-masing memiliki data dalam tiga format berbeda. Grafik konversi yang menunjukkan format mana yang dapat dikonversi satu sama lain ditunjukkan pada Gambar. Setiap situs juga dapat memberikan jumlah data asli yang tersedia dalam setiap format di situs tersebut. Pada gambar tersebut, kita dapat mengasumsikan bahwa jumlah tersebut diberikan dalam ribuan sampel yang tersedia.

Setelah grafik konversi untuk setiap klien fusi tersedia di server fusi, grafik konversi dapat menentukan berapa banyak data yang dapat tersedia dengan mengonversi ke setiap format yang mungkin tersedia. Untuk kumpulan grafik konversi tertentu yang ditunjukkan pada Gambar 4.1, jumlah data yang dapat diproses pada setiap klien fusi dihitung pada Tabel 4.2. Setiap baris dalam tabel menunjukkan jumlah sampel data di setiap klien fusi yang dapat dikonversi ke format di kolom pertama. Angka-angka di kolom Asli menunjukkan jumlah yang tersedia di klien fusi dalam format tersebut. Angka-angka di kolom Convertible menunjukkan jumlah data yang dapat dikonversi ke format yang ditentukan di setiap klien fusi. Baris kedua pada kolom Convertible menunjukkan format data yang dikonversi, dengan konvensi F1+F2+F3 yang menghitung data yang dapat dikonversi dari format tertentu.

Saat memeriksa jumlah data asli yang tersedia di masing-masing format, kita dapat melihat bahwa format F1 adalah yang paling umum dengan 190 juta sampel di seluruh klien fusi, dibandingkan dengan F2 (80 juta sampel) atau F3 (40 juta sampel). Namun, karena beragamnya kemampuan klien fusi yang berbeda untuk mengkonversi ke format data yang berbeda, lebih baik memilih format F3 untuk konversi. Pemilihan F3 memungkinkan 300 juta sampel digunakan untuk data pelatihan, dibandingkan dengan format F1 yang paling umum, yang pemilihannya hanya memungkinkan 280 juta sampel digunakan untuk pelatihan. Konsep grafik konversi memungkinkan penentuan format dimana sebagian besar sampel dapat dikonversi.



Gambar 4.1 Contoh grafik konversi.

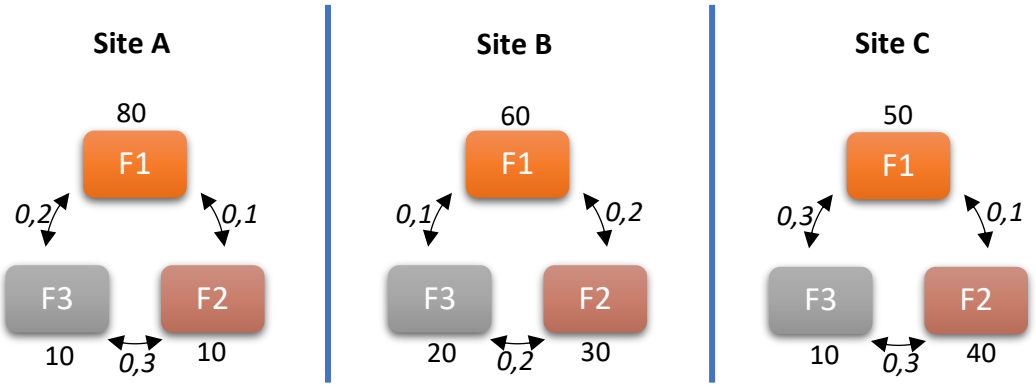
Tabel 4.2: Contoh pemilihan format menggunakan grafik konversi.

		Asli		
Format	Klien A	Klien B	Klien C	Total
F1	80	60	50	190
F2	10	30	40	80
F3	10	20	10	40

		Mobil atap terbuka		
Format	**Klien A**	**Klien B**	**Klien C**	**Total**
F1	100 (80+10+10)	90 (60+30+0)	90 (50+40 +0)	280
F2	10 (80+10+10)	30 (60+30+0)	40 (0+40+0)	80
F3	100 (80+10+10)	100 (60+30+20)	100 (50+40+10))	300

Pembahasan sejauh ini berdasarkan Tabel 4.2 mengasumsikan bahwa tidak ada biaya untuk mengkonversi data dari format apa pun ke format lainnya. Namun, dalam banyak kasus mungkin ada biaya yang terkait dengan konversi, khususnya jika layanan berbasis web digunakan. Biaya konversi apa pun dapat direpresentasikan sebagai biaya melintasi tautan grafik konversi. Hal ini akan memungkinkan nilai tertimbang dari biaya yang diperlukan dalam konversi data dan utilitas yang disediakan oleh setiap format data tambahan. Matriks tertimbang ini kemudian dapat digunakan untuk membandingkan trade-off biaya-utilitas untuk masing-masing format, dan format yang tepat kemudian dapat dipilih.

Mari kita asumsikan bahwa kegunaan setiap titik data tambahan adalah 0,1 unit, sedangkan biaya untuk mengubah item ke dalam format berbeda di berbagai klien fusi adalah seperti yang ditunjukkan pada Gambar 4.2. Dalam hal ini, biaya yang terkait dengan penggunaan masing-masing format yang berbeda dan kegunaan penggunaan format tertentu adalah seperti yang ditunjukkan pada Tabel 4.3.



Gambar 4.2 Grafik konversi dengan biaya.

Tabel 4.3: Contoh pemilihan format menggunakan biaya.

Format	Biaya Klien A	Biaya Klien B	Biaya Klien C	Biaya Bersih	Kegunaan
F1	3 (0,1*10+0,2*10)	6 (0,2*30)	4 (0)	13	28
F2	11 (0,1*80+0,3*10)	12 (0,2*60)	0 (0)	23	23
F3	7 (0,2*80+0,3*10)	12 (0,1*60+0,2*30)	27 (0,3*50+0,3*40)	46	30

Jika kita berasumsi bahwa nilai bersih adalah selisih antara biaya dan utilitas, kita dapat melihat bahwa format F1 adalah format terbaik untuk digunakan. Tergantung pada konteks bisnis tertentu, fungsi yang berbeda untuk menggabungkan biaya dan utilitas akan lebih tepat. Namun, analisis pemilihan format berdasarkan grafik konversi dapat digunakan dengan cara yang sama.

4.1.2 Data Unggulan

Berbeda dengan data mentah, data unggulan terdiri dari banyak fitur berbeda dan label keluaran. Fitur-fitur ini dapat diambil dari data mentah, misalnya. sampel audio dapat diubah menjadi fitur seperti Mel-Frequency Cepstral Coefficients (MFCC), yang merupakan sekumpulan fitur yang biasa digunakan dalam aplikasi pemrosesan ucapan. Secara umum, kumpulan data unggulan dapat dilihat sebagai representasi data dalam tabel, dimana salah satu kolomnya adalah kolom keluaran dan kolom lainnya adalah kolom masukan.

Ketika data unggulan didistribusikan ke banyak klien fusi, setiap klien berada di lokasi berbeda, fitur tersebut mungkin diberi nama berbeda di klien fusi berbeda, yaitu mereka dapat menggunakan nama berbeda untuk kolom yang sama dalam tabel datanya. Untuk keperluan bagian ini, kami akan berasumsi bahwa semua kolom ada di semua klien fusi. Namun, kolom tersebut mungkin diberi nama berbeda. Selain itu, beberapa kolom mungkin memiliki format berbeda di lokasi fusi berbeda. Sebelum klien fusi yang berbeda dapat mulai melatih model, masing-masing klien perlu mengonversi data ke format umum, yang fitur dan label keluarannya identik di seluruh klien fusi.

Pada bagian ini, pendekatan untuk menangani perbedaan penamaan fitur, dan representasi fitur yang berbeda akan dibahas. Pendekatan untuk menangani fitur yang hilang

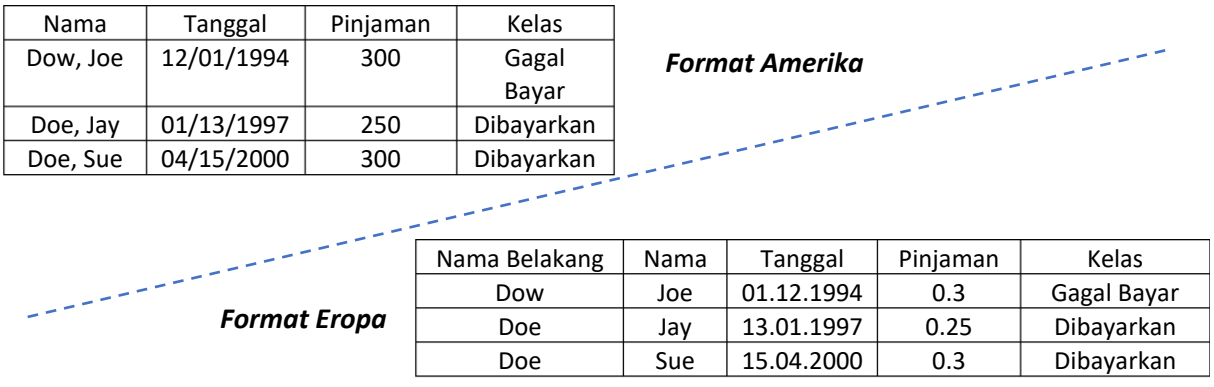
dibahas di Bab 8. Ilustrasi ketidakcocokan data ditunjukkan pada Gambar 4.3 untuk kasus sederhana, yaitu sebuah bank yang mencatat pinjaman mana yang telah dilunasi seluruhnya dan pinjaman mana yang gagal bayar di beberapa negara berbeda. Data di semua negara memiliki fitur yang sama, yaitu nama, tanggal lahir, jumlah pinjaman dan kelas pinjaman (baik yang sudah dibayar atau yang gagal bayar). Klien fusion di semua negara akan terlibat dalam latihan pembelajaran gabungan untuk menciptakan model umum. Namun, klien fusi memiliki akses ke data yang disimpan dalam dua format berbeda, format Amerika dan format Eropa.

Klien fusion mengikuti format Amerika menyimpan data di kolom nama sebagai Nama Depan, Nama Kedua, kolom tanggal menggunakan format bulan/hari/tahun, dan pinjaman dihitung jumlahnya dalam ribuan dolar. Untuk klien fusi yang mengikuti format Eropa, nama diwakili dalam dua kolom, Nama Belakang dan Nama (yang menunjukkan nama diri atau nama depan), tanggal menggunakan format dd.mm.yyyy (konvensi Eropa), dan pinjaman menghitung jumlah dalam jutaan dolar.

Dua tabel yang ditunjukkan pada Gambar 4.3 berisi data yang sama seperti yang muncul pada dua klien fusi yang berbeda. Kedua tabel tersebut terlihat sangat berbeda meskipun keduanya mewakili data yang sama persis. Kecuali jika data diubah ke format yang konsisten, akan sulit membuat model yang bermakna dari kombinasi data dari kedua sumber.

Konversi Berbasis Aturan ke Format Kanonis Proses konversi perlu diotomatisasi di setiap klien fusi. Pendekatan untuk melakukan penerjemahan secara otomatis adalah dengan menentukan seperangkat aturan untuk menerjemahkan dan mengkonversi entri di setiap kolom pada setiap klien fusi ke format umum. Ilmuwan data yang memulai proses pembelajaran model perlu mendefinisikan seperangkat aturan penerjemahan untuk setiap format asli yang berbeda dari format umum.

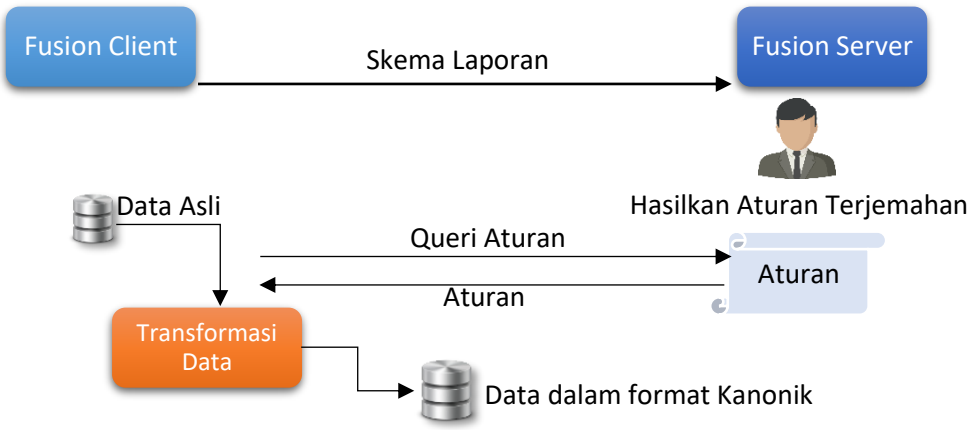
Untuk contoh khusus yang diilustrasikan pada Gambar 4.3, mari kita asumsikan bahwa ada dua klien fusi dengan data dalam format Eropa dan tiga klien fusi dengan data dalam format Amerika. Ilmuwan data telah memilih untuk menggunakan format kanonik yang berbeda dari format Amerika dan Eropa.



Gambar 4.3: Contoh ketidakcocokan data unggulan.

Untuk memungkinkan penggunaan data yang disimpan dalam format Eropa untuk mengkonversi data ke dalam format yang dipilih, seperangkat aturan penerjemahan dari

format Eropa ke format Kanonik akan ditentukan. Seperangkat aturan lain untuk mengubah format Amerika ke format Canonical juga diperlukan. Aturan-aturan ini akan disimpan di server fusi, dan setiap klien fusi perlu mengambil aturan terjemahan untuk formatnya dari server sebagai langkah awal.



Gambar 4.4 Diagram interaksi untuk memperoleh aturan penerjemahan.

Prosesnya dapat diimplementasikan menggunakan diagram interaksi yang ditunjukkan pada Gambar 4.4. Dalam diagram interaksi, urutan pesan yang mengalir antara entitas yang berbeda ditampilkan. Awalnya, masing-masing klien fusi akan melaporkan skema mereka (misalnya nama kolom lokal) ke server fusi. Ilmuwan data di server fusi akan memeriksa skema yang dilaporkan dari semua situs, menentukan jumlah format yang berbeda, dan menentukan kebijakan terjemahan untuk setiap format. Setelah kebijakan terjemahan ditentukan untuk setiap format, dan setiap klien fusi dikaitkan dengan format yang digunakannya, klien fusi dapat menanyakan server fusi untuk mengambil aturan terjemahan yang berkaitan dengannya, dan menerapkan aturan tersebut untuk mengonversi data mereka menjadi format umum.

Mari kita asumsikan bahwa ilmuwan data telah mendefinisikan format kanonik yang terdiri dari kolom Nama Belakang, Nama Depan, Tanggal, Jumlah pinjaman dalam ribuan, dan Kelas. Ini memeriksa lima klien fusi untuk menentukan bahwa ada dua format berbeda yang digunakan. Untuk klien fusi yang menggunakan format pertama (format Amerika), aturan penerjemahan akan menyatakan bahwa bidang Nama Belakang adalah bagian dari bidang Nama sebelum pemisah koma, dan bidang Nama Depan adalah bagian dari bidang Nama setelah pemisah koma. Bidang Tanggal, Pinjaman, dan Kelas tetap tidak berubah. Untuk klien fusi yang menggunakan format Eropa, aturan terkait akan menyatakan bahwa bidang Nama Belakang tidak berubah, bidang Nama Depan adalah bidang Nama, bidang Tanggal diperoleh dengan mengubah pemisah pada data asli dari titik menjadi garis miring dan membalikkan karakter bidang bulan dan tanggal, dan bidang Pinjaman diperoleh dengan mengalikan bidang Pinjaman asli dengan seribu.

Pengelolaan dan definisi aturan penerjemahan ini dapat dilakukan dengan berbagai cara. Kami memilih salah satu cara seperti penggunaan sistem manajemen kebijakan. Dalam

sistem ini, kebijakan didefinisikan sebagai kumpulan kondisi dan tindakan, dimana tindakan tersebut dapat melakukan tugas tertentu seperti mengubah kolom data dengan menjalankan beberapa fungsi terjemahan. Hal ini dapat diungkapkan dalam bahasa alami dan diterjemahkan ke dalam representasi yang dapat dibaca komputer, yang dapat menjadi representasi terstruktur dari pasangan tindakan kondisi dalam bahasa seperti CIM-SPL. Representasi yang dapat dibaca komputer ini digunakan oleh klien fusi untuk menerjemahkan data lokal mereka ke dalam format kanonik.

Normalisasi yang Konsisten Normalisasi adalah proses mengubah semua fitur numerik menjadi rentang umum sehingga tidak ada fitur yang mendominasi fitur lainnya melalui representasi volumenya. Ini digunakan saat melatih model untuk memastikan tidak ada fitur yang dianggap terlalu penting saat menganalisis data untuk melatih model.

Sebagai contoh perlunya normalisasi, mari kita pertimbangkan tugas menentukan apakah suatu pinjaman bank berisiko berdasarkan usia peminjam, tingkat pendapatan tahunan mereka dan jumlah pinjaman yang diminta. Usia peminjam biasanya berkisar antara 20 dan 100, sedangkan pendapatan tahunan dan jumlah pinjaman berkisar beberapa ribu. Oleh karena itu, hal terakhir ini kemungkinan besar akan membebani parameter formula atau model AI apa pun yang digunakan untuk memprediksinya. Lebih jauh lagi, jika pendapatan bulanan digunakan sebagai pengganti pendapatan tahunan, kepentingan relatif dari parameter-parameter tersebut akan berubah lagi. Karena angka yang besar cenderung mempengaruhi parameter model secara tidak proporsional, menjaga parameter yang berbeda dalam rentang yang berbeda akan menyebabkan distorsi pada model.

Untuk mendapatkan kepentingan relatif yang lebih baik di antara berbagai fitur, praktik yang disarankan adalah menskalakan semua fitur numerik ke skala umum, biasanya nilai antara 0 dan 1. Pendekatan berbeda untuk penskalaan ini tersedia dan didukung di berbagai perangkat lunak pembelajaran mesin paket.

Untuk fitur yang mengambil nilai diskrit tanpa urutan, umumnya disebut sebagai fitur kategorikal, langkah normalisasi biasanya mengharuskan fitur tersebut dikonversi ke sekumpulan nilai biner, dengan jumlah nilai berbeda dari fitur tersebut menentukan berapa banyak nilai biner tersebut. diperkenalkan. Konvensi umum mencakup satu variabel bernilai biner untuk setiap nilai fitur yang berbeda, atau satu variabel kurang dari angka ini. Konversi variabel kategori ke nilai biner memungkinkan model yang lebih akurat.

Meskipun normalisasi relatif mudah jika semua data berada di satu lokasi, hal ini dapat menjadi masalah jika data disimpan di lokasi berbeda. Untuk fitur numerik, pendekatan penskalaan didasarkan pada properti statistik data, seperti mean, max, atau mean, bergantung pada jenis pendekatan penskalaan. Atribut ini perlu dihitung di seluruh klien fusi dengan datanya. Tanpa koordinasi seperti itu, klien fusi yang berbeda dapat menormalkan fitur secara berbeda. Misalnya, jika konvensinya adalah menskalakan data secara seragam antara 0 dan 1 berdasarkan nilai minimum dan maksimum fitur, nilai-nilai ini kemungkinan besar akan berbeda di setiap klien fusi. Kurangnya koordinasi dalam penskalaan berarti bahwa fitur yang sama dipetakan secara berbeda oleh klien fusi yang berbeda, sehingga menghasilkan model yang tidak sesuai.

Jenis penskalaan lainnya adalah dengan memodifikasi nilai parameter sehingga semuanya diskalakan ke distribusi dengan rata-rata nol dan deviasi standar 1. Semua klien fusi harus menyetujui jenis penskalaan yang akan digunakan, dan menghitung parameternya melakukan penskalaan secara konsisten pada semuanya.

Jenis interaksi antara klien fusi dan server fusi yang digunakan untuk mencapai format kanonik umum yang dijelaskan sebelumnya dapat digunakan kembali untuk mendapatkan skema normalisasi yang konsisten. Setiap klien fusi melaporkan statistik data lokalnya (misalnya hitungan, min, rata-rata, maks, dll.) ke server fusi. Server fusi dapat menentukan nilai yang sesuai untuk semua data yang didistribusikan ke seluruh klien fusi, mengikuti pendekatan yang dijelaskan di Bagian 3.1 Bab 3. Aturan normalisasi kemudian dapat ditentukan, yang mencakup pemilihan algoritma untuk normalisasi, bersama dengan dengan statistik agregat data yang relevan.

Perhatikan bahwa server fusi dapat memilih skema normalisasi apa pun yang konsisten di semua klien fusi, dan skema tersebut tidak perlu didasarkan pada properti data agregat. Misalnya, akan baik-baik saja bagi server fusi untuk menentukan aturan normalisasi berdasarkan statistik yang dimilikinya untuk klien fusi pertama yang menghubunginya. Setiap klien fusi berikutnya yang menghubungi server fusi akan diberikan aturan normalisasi berdasarkan statistik klien fusi pertama Sebagai contoh, mari kita pertimbangkan klien fusi yang menskalakan semua nilai secara linear antara nilai minimum dan maksimum suatu fitur. Terdapat empat klien fusi dengan data yang didistribusikan seperti yang ditunjukkan pada Tabel 4.4. Dalam tabel, hanya nilai untuk satu fitur numerik yang ditampilkan, namun konsep tersebut dapat dengan mudah diperluas ke beberapa fitur.

Tabel 4.4: Contoh normalisasi data.

Klien Fusi	Jumlah Titik Data	Nilai Minimum	Nilai maksimum
Klien A	3000	6	28
Klien B	5000	1	30
Klien C	9000	3	46
klien D	1000	7	20

Dengan contoh ini, tanpa koordinasi apa pun, rumus penskalaan klien fusi A untuk fitur yang ditunjukkan pada Tabel 4.4 adalah $(x - 6) / 22$, dengan x adalah nilai fitur untuk 3.000 titik data mana pun. Rumus penskalaan untuk klien fusi B adalah $(x - 1)/22$, untuk klien fusi C adalah $(x - 3)/43$, dan untuk klien fusi D adalah $(x - 7)/20$. Meskipun setiap klien fusi akan menskalakan fiturnya ke nilai antara 0 dan 1, nilai fitur asli yang sama akan diskalakan secara berbeda pada klien fusi yang berbeda. Hal ini tidak akan menghasilkan model AI yang benar. Sebelum pembelajaran model dimulai, setiap klien fusi harus mengirimkan statistik agregat, jumlah poin, nilai minimum, maksimum, atau rata-rata fitur ke server fusi. Server fusi dapat menghitung nilai minimum di antara semua klien fusi, serta maksimum, dan mengirimkannya kembali ke masing-masing klien fusi. Kemudian, setiap klien fusi dapat menskalakan sesuai dengan minimum atau maksimum global. Dalam hal ini, minimum global adalah 1 dan

maksimum global adalah 46, sehingga rumus penskalaan umum di setiap klien fusi akan menjadi $(x - 1)/46$.

Perhatikan bahwa pendekatan alternatif di mana klien fusi pertama yang dihubungi (misalnya klien fusi A) server fusi memberikan nilainya dan server fusi menginstruksikan semua klien lain untuk menggunakan rumus penskalaan $(x - 6)/28$ berdasarkan klien fusi pertama yang dihubungi dia. Itu juga akan menjadi fungsi penskalaan yang dapat diterima di semua klien, dengan satu-satunya masalah adalah bahwa fungsi tersebut mungkin menskalakan beberapa nilai menjadi negatif (kurang dari 0) dan beberapa di atas 1, jika klien fusi memiliki data yang tidak termasuk dalam rentang penskalaan yang disediakan oleh server fusi. Namun, karena semua klien fusi menggunakan rumus penskalaan yang sama, mereka tetap dapat melatih model yang menggabungkan wawasan secara akurat di seluruh klien fusi.

Untuk nilai kategoris, server fusi perlu menentukan kumpulan nilai global yang dapat diambil oleh fitur kategoris ini. Beberapa nilai mungkin tidak ada dalam data di beberapa klien fusi. Untuk mendefinisikan pemetaan ini, skema penamaan yang konsisten untuk setiap fitur biner yang dihasilkan perlu dibuat. Jika setiap klien fusi melaporkan semua nilai unik yang dimilikinya untuk setiap variabel kategori, server fusi dapat menggabungkan keseluruhan kumpulan nilai unik, dan menentukan kumpulan nilai unik untuk digunakan semua orang. Dengan nilai kategoris, akan berguna jika semua klien fusi melaporkan nilai uniknya sebelum keseluruhan kumpulan dibuat.

Alternatifnya, server fusi dapat memutuskan untuk menggunakan sejumlah variabel biner tetap untuk setiap kategori, dan menetapkan nilai-nilai ini ke klien fusi yang berbeda saat mereka menghubungi server fusi secara berurutan. Salah satu dari variabel-variabel ini harus dicadangkan sebagai kategori umum jika jumlah kategori tetap yang telah ditentukan terlalu kecil, dan nilai-nilai baru ditemukan kemudian. Contoh koordinasi antar variabel kategori ditunjukkan pada Tabel 4.5. Ini menunjukkan empat klien fusi dengan fitur tertentu mengambil tiga nilai berbeda di masing-masing klien fusi. Namun, nilai pada masing-masing klien fusi sedikit berbeda. Misalkan nilai fitur adalah nama warna, proses standar pengkodean variabel kategori sebagai pengkodean One-Hot dapat diikuti.

Tabel 4.5: Contoh normalisasi data kategorikal.

Klien Fusi	Nilai-Nilai yang Berbeda
Klien A	Biru, Merah, Emas
Klien B	Merah, Biru, Kuning
Klien C	Emas, Hijau, Kuning
klien D	Merah, Biru, Hijau

. Oleh karena itu, tanpa koordinasi, setiap situs akan memilih array biner yang panjangnya tiga unit dan satu kemungkinan pengkodean akan seperti yang ditunjukkan pada Tabel 4.6. Seperti terlihat pada tabel, nilai yang sama dapat diberi kode berbeda pada klien fusi yang berbeda. Selain itu, pengkodean yang sama pada klien fusi yang berbeda mungkin mewakili nilai yang berbeda. Meskipun terdapat korespondensi satu-ke-satu antara nilai

kategorikal dan representasi yang dikodekan pada klien fusi mana pun, korespondensi ini hilang saat memeriksa data di seluruh klien fusi. Menggabungkan data dari masing-masing klien fusi dengan pengkodean yang tidak konsisten ini akan menghasilkan model yang aneh dan tidak konsisten.

Tabel 4.6: Pengkodean data kategorikal.

Klien Fusi A	
Kode	*Nilai*
[1, 0, 0]	Biru
[0, 1, 0]	Merah
[0, 0, 1]	Emas

Klien Fusi B	
Kode	*Nilai*
[1, 0, 0]	Merah
[0, 1, 0]	Biru
[0, 0, 1]	Kuning

Klien Fusi C	
Kode	*Nilai*
[1, 0, 0]	Emas
[0, 1, 0]	Hijau
[0, 0, 1]	Kuning

Klien Fusi D	
Kode	*Nilai*
[1, 0, 0]	Merah
[0, 1, 0]	Biru
[0, 0, 1]	Hijau

Dalam pengkodean One-Hot, fitur kategorikal direpresentasikan sebagai vektor biner yang panjangnya merupakan jumlah nilai unik yang diambil fitur tersebut. Setiap nilai direpresentasikan sebagai array satu dan nol. Setiap array mempunyai tepat satu kemunculan angka 1, yaitu semua entri kecuali satu adalah nol. Posisi dalam array yang bernilai 1 berbeda untuk setiap nilai unik fitur. Dalam contoh yang ditunjukkan pada Tabel 4.5, fitur tersebut mengambil tiga nilai berbeda pada masing-masing klien fusi

Jika server fusi digunakan untuk koordinasi nilai kategorikal, maka server fusi dapat menentukan bahwa ada 5 nilai berbeda: Biru, Merah, Emas, Kuning, dan Hijau. Ini akan memungkinkan server fusi untuk merekomendasikan pengkodean One-Hot yang panjangnya 5 bit untuk nilai fitur. Hasil akhirnya adalah pemetaan yang konsisten di antara klien fusi yang berbeda. Pengkodean bersih yang dihasilkan adalah seperti yang ditunjukkan pada Tabel 4.7. Sinkronisasi untuk normalisasi tidak memerlukan pertukaran data mentah. Sebaliknya, hanya informasi ringkasan tentang data yang ada di setiap klien fusi yang perlu dipertukarkan. Proses ini memastikan data pelatihan telah diproses sebelumnya dan disiapkan secara konsisten di seluruh klien fusi.

Tabel 4.7: Pengkodean data kategorikal dengan koordinasi.

Kode	**Nilai**
[1, 0, 0, 0, 0]	Biru
[0, 1, 0, 0, 0]	Merah
[0, 0, 1, 0, 0]	Emas
[0, 0, 0, 1, 0]	Kuning
[0, 0, 0, 0, 1]	Hijau

4.2 MENYELESAIKAN KONFLIK NILAI

Konflik nilai adalah masalah yang muncul pada fitur kategoris, termasuk label keluaran yang bersifat kategoris dalam banyak situasi. Jika klien fusion A mengacu pada kualitas pinjamannya sebagai hijau, kuning, dan merah sedangkan klien fusion B mengacu pada kualitas pinjamannya sebagai vert, jaune, dan rouge (nama Perancis untuk warna yang sama), kita mengalami konflik nilai. Untuk masukan unggulan, konflik nilai dapat muncul di fitur mana pun (yaitu kolom mana pun dalam representasi tabel). Untuk masukan mentah, konflik nilai dapat muncul pada label keluaran.

Untuk mengilustrasikan masalah ini, mari kita pertimbangkan masalah klasik klasifikasi gambar. Jika gambar dipertahankan pada klien fusi yang berbeda secara independen atau di bawah konvensi yang berbeda, kemungkinan besar gambar tersebut diberi label berbeda. Sebagai contoh, mari kita perhatikan tiga klien fusi yang ditunjukkan pada Gambar 4.5. Jenis gambar yang sama terdaftar sebagai Mobil di klien fusi A, Mobil di klien fusi B, dan Kendaraan di klien fusi C. Demikian pula, apa yang diberi label sebagai bunga di klien fusi A disebut Tanaman di klien fusi B dan Bunga di klien fusi C. Setiap klien fusi hanya menggunakan dua label untuk mendeskripsikan gambar yang diambilnya, namun gambar tersebut diberi nama dan dikategorikan dengan sangat berbeda.

Gambar 4.5: Contoh yang mengilustrasikan konflik nilai.

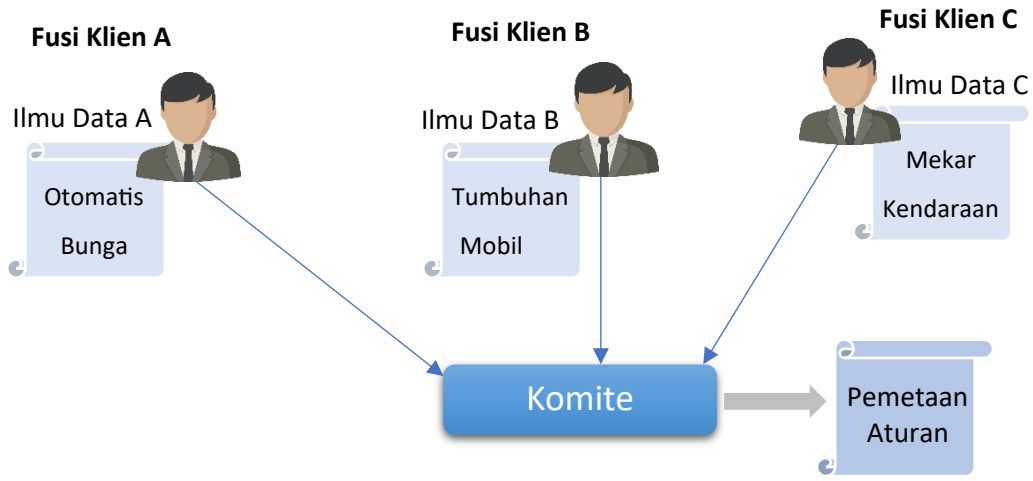
Jika label yang berbeda pada klien fusi yang berbeda tidak direkonsiliasi, model yang dilatih dengan label yang tidak konsisten akan memiliki kualitas yang buruk. Jenis data yang sama akan didefinisikan dengan label berbeda dan model akan mencoba mempelajari perbedaan di antara dua kelompok yang pada dasarnya sama, misalnya. dua gambar pertama pada klien fusi A dan klien fusi B sama, tetapi diberi label berbeda. Tantangannya, tentu saja, adalah untuk merekonsiliasi konflik di antara semua label ini tanpa mengharuskan data dipindahkan dari klien fusi. Dalam beberapa subbagian berikutnya, kita melihat pendekatan untuk mengatasi konflik nilai dan menyelesaikannya.

4.2.1 Pendekatan Komite terhadap Rekonsiliasi

Pendekatan pertama dan yang lebih disukai adalah dengan menerapkan rekonsiliasi manual terhadap label-label yang berbeda. Pendekatan rekonsiliasi manual adalah dengan menentukan sekumpulan label terjemahan untuk masing-masing klien fusi yang menentukan

bagaimana label pada setiap klien harus diubah. Kita dapat membayangkan sebuah komite yang terdiri dari satu orang yang mewakili setiap klien fusi yang duduk bersama dalam konferensi web dan membuat keputusan konsensus tentang label umum yang seharusnya dan bagaimana label di setiap situs harus dipetakan ke label yang konsisten. Proses ini diilustrasikan secara kiasan pada Gambar 4.6. Untuk contoh yang ditunjukkan pada Gambar 4.5, kita dapat memutuskan bahwa label akhir adalah Mobil dan Bunga, dan aturan pemetaan ulang label dapat didefinisikan sebagai berikut:

- Jika klien fusi adalah A dan labelnya adalah 'Otomatis', ubah label menjadi 'Mobil'
- Jika klien fusi adalah B dan labelnya adalah 'Tanaman', ubah label menjadi 'Bunga'
- Jika klien fusi adalah C dan labelnya adalah 'Kendaraan', ubah label menjadi 'Mobil'
- Jika klien fusi adalah C dan labelnya adalah 'Mekar', ubah label menjadi 'Bunga'



Gambar 4.6: Rekonsiliasi konflik nilai secara manual.

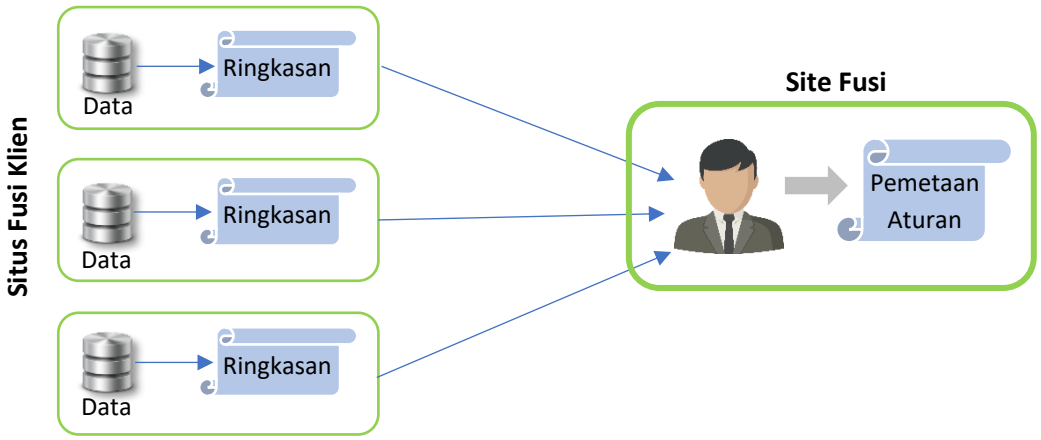
Aturan-aturan ini dapat didefinisikan secara terpusat berdasarkan konsensus manual, disimpan di server fusi, dan kemudian diambil oleh setiap klien fusi sebelum pelatihan model dilakukan untuk merekonsiliasi semua label. Jika data ditampilkan, nilai yang berbeda dari setiap kolom dapat dijalankan melalui proses serupa untuk mendapatkan serangkaian nilai yang konsisten untuk setiap item dalam kolom.

4.2.2 Pendekatan Rangkuman dalam Rekonsiliasi

Karena tidak selalu memungkinkan untuk memiliki pakar untuk setiap klien fusi untuk proses pembelajaran gabungan, kita mungkin ingin mempertimbangkan pendekatan alternatif jika hanya ada satu pakar yang tersedia dan beroperasi di server fusi. Pakar manusia ini dapat melihat ringkasan hasil dari setiap klien fusi, dan menentukan aturan penerjemahan untuk setiap klien fusi. Prosesnya akan dimulai dengan perangkat lunak di setiap situs yang mengumpulkan kumpulan semua label untuk setiap fitur dan mengirimkannya ke server fusi. Pakar manusia kemudian akan melihat kumpulan label gabungan di seluruh klien fusi, dan menentukan aturan pelabelan ulang yang harus digunakan oleh setiap klien fusi. Proses ini hanya memerlukan satu orang ahli, di server fusi, yang perlu mengeksplorasi data yang diringkas dan menentukan kebijakan penerjemahan yang sesuai.

Setelah kebijakan ditentukan, klien fusi dapat mengambil kebijakan pelabelan ulang dari server fusi, dan menerapkannya untuk mendapatkan tampilan data yang diubah yang akan digunakan. Selain tugas menentukan kebijakan, seluruh proses dapat diotomatisasi. Hasilnya, pendekatan yang ditunjukkan pada Gambar 4.7 lebih praktis dan efektif dibandingkan pendekatan yang ditunjukkan pada Gambar 4.6.

Proses rekonsiliasi manual yang ditunjukkan pada Gambar 4.7 akan bekerja dengan baik jika terdapat sejumlah kecil label yang terdefinisi dengan baik untuk setiap entri yang berbeda, atau sekumpulan nilai yang terdefinisi dengan baik untuk setiap kolom dari kumpulan data unggulan. Namun, dalam praktiknya, seseorang mungkin menghadapi situasi ketika terdapat banyak label, dan rekonsiliasi manual tidak dapat dilakukan. Alat yang dapat membantu pakar di server fusi dalam tugas merekonsiliasi kebijakan akan berguna untuk membantu proses peringkasan. Kami menjelaskan beberapa alat ini di bagian selanjutnya.



Gambar 4.7: Pendekatan peringkasan untuk rekonsiliasi label.

4.2.3 Matriks Kebingungan Lintas Situs

Matriks kebingungan lintas situs dapat menjadi alat yang berguna untuk mengidentifikasi label yang berbeda pada klien fusi yang berbeda. Hal ini menggunakan konsep matriks konfusi, yang sering digunakan dalam pendekatan klasifikasi pembelajaran mesin untuk menganalisis dan memahami kinerja berbagai algoritma. Matriks konfusi model AI menunjukkan persentase prediksi model yang cocok dengan label data sebenarnya. Setiap baris matriks mewakili jumlah kelas prediksi untuk suatu titik data, sedangkan setiap kolom mewakili nilai sebenarnya dari titik data tersebut. Pada beberapa makalah, urutan baris dan kolom diubah, namun hal tersebut tidak membuat perbedaan material pada penggunaan matriks konfusi. Untuk keperluan bab ini, kita akan membahas matriks konfusi di mana setiap entri dalam sel matriks konfusi mewakili persentase kejadian aktual dalam data pengujian yang diprediksi dengan benar oleh model. Dalam hal ini, kolom-kolom matriks konfusi akan berjumlah 1,0.

Praktik umum dalam pembelajaran mesin adalah membagi data yang tersedia menjadi set pelatihan dan set pengujian, melatih model pada set pelatihan, dan menggunakan matriks konfusi untuk mengevaluasi model yang dihasilkan pada set pengujian. Biasanya, jika data

pelatihan yang digunakan untuk melatih model memiliki label yang sama dengan data pengujian, label pada baris dan kolom matriks konfusi akan sama.

Jika seseorang melatih model AI sepenuhnya menggunakan data yang tersedia untuk klien fusi A, seseorang akan mendapatkan model yang mengubah gambar apa pun menjadi salah satu dari dua kelas, Otomatis atau Bunga. Ketika diterapkan pada data di klien yang sama, A, mis. dengan membaginya menjadi kumpulan data pelatihan dan pengujian, hal ini dapat menghasilkan matriks kebingungan seperti yang ditunjukkan pada Tabel 4.8. Dari gambar sebenarnya di kelas 'Otomatis', model memprediksi 95% benar dan 5% salah. Dari gambar sebenarnya di kelas 'Bunga', 97% diklasifikasikan dengan benar dan 3% salah. Secara umum, jika data pelatihan dan pengujian tidak identik, jarang ada model yang 100% akurat. Namun, jika keakuratannya cukup baik untuk memberikan keuntungan bisnis, beberapa kesalahan dapat diterima.

Tabel 4.8: Contoh matriks kebingungan.

	Sebenarnya	
Diprediksi	Mobil	Bunga
Mobil	95%	3%
Bunga	5%	97%

Tabel 4.9: Matriks kebingungan lintas situs untuk model yang dilatih di Klien A dan diuji di Klien B.

	Sebenarnya	
Diprediksi	Mobil	Tanaman
Mobil	96%	2%
Bunga	4%	98%

Untuk memeriksa ketidakkonsistenan label di seluruh klien fusi yang berbeda, kita dapat menggunakan matriks konfusi, kecuali dengan menggunakannya di beberapa lokasi. Masing-masing klien fusi akan melatih model mereka sendiri dengan murni menggunakan data lokal, dan mengirimkannya ke klien fusi lainnya. Server fusi dapat digunakan sebagai titik pusat untuk distribusi model ke setiap klien fusi. Setelah menerima model dari klien fusi lain, masing-masing klien fusi akan menjalankan model pada datanya dan membuat matriks konfusi. Perhatikan bahwa, dalam kasus khusus ini, model akan memprediksi label tempat model dilatih, yaitu label yang ada di klien fusi yang menyediakan data pelatihan. Namun, label sebenarnya adalah label yang ada pada klien fusi yang menjalankan model. Dalam matriks konfusi lintas situs ini, label pada baris dan kolom matriks konfusi akan berbeda. Perbandingan matriks konfusi akan mengungkapkan bagaimana label yang berbeda di seluruh klien fusi dipetakan satu sama lain.

Dalam matriks kebingungan lintas situs yang ditunjukkan pada Tabel 4.9, model dilatih menggunakan data di klien fusi A, seperti yang ditunjukkan pada Gambar 4.5 dan diuji pada data yang tersedia di klien B. Hasilnya, label sebenarnya, ditampilkan sebagai judul kolom

pada tabel terdapat label yang digunakan pada fusion client B yaitu Mobil dan Pabrik. Label baris, bagaimanapun, adalah nilai yang akan diprediksi oleh model, yang merupakan label yang digunakan pada klien fusi A, yaitu Otomatis dan Bunga. Perbandingan entri dalam dua matriks konfusi menunjukkan bahwa label Otomatis berhubungan dengan Mobil dan label Bunga berhubungan dengan Tanaman.

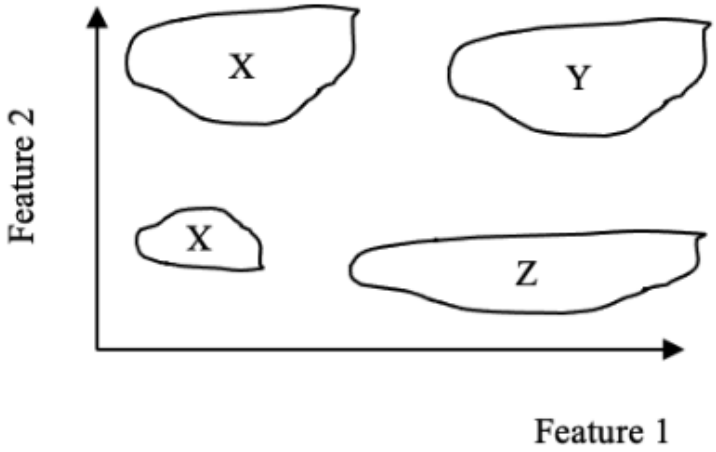
Alat yang menghitung matriks kebingungan lintas situs di seluruh klien fusi dapat membantu ilmuwan data yang ditunjukkan pada Gambar 4.7 dengan tugas mengidentifikasi kebijakan pemetaan label yang tepat. Untuk contoh spesifik dari tiga klien fusi yang sedang kita diskusikan, pendekatan ini dapat digunakan untuk mengidentifikasi korespondensi antar label pada klien fusi yang berbeda.

4.2.4 Analisis Fitur Ruang

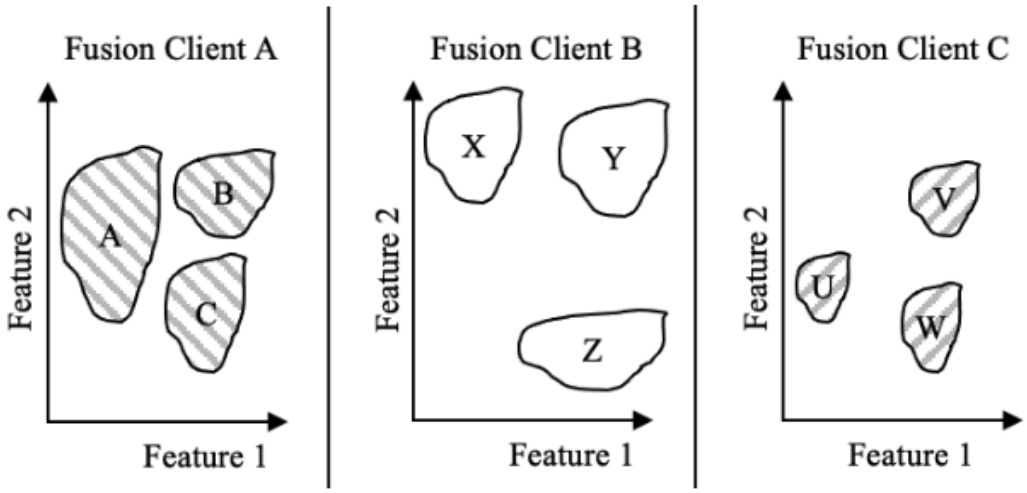
Meskipun matriks kebingungan lintas situs memberikan pendekatan yang baik untuk menentukan pemetaan label satu sama lain ketika label pada klien fusi yang berbeda memiliki korespondensi satu-satu, hal ini tidak akan bekerja dengan efektif ketika label-label tersebut tumpang tindih satu sama lain. Ada kemungkinan bahwa satu klien fusi mungkin menggunakan dua label untuk beberapa jenis data dan klien fusi lainnya mungkin hanya menggunakan satu jenis data. Untuk kasus gambar yang spesifik, mari kita pertimbangkan situasi di mana tiga klien fusi mempunyai gambar mesin pemotong rumput atau mobil yang perlu dibedakan. Klien fusi pertama mungkin menggunakan gambar mobil yang diberi label dengan nama penanda mobil atau mesin pemotong rumput, sedangkan klien fusi kedua mungkin menggunakan gambar yang hanya memberi label semuanya sebagai kendaraan atau mesin pemotong rumput. Demikian pula, klien fusi ketiga mungkin menggunakan label yang membedakan nama pabrikan mesin pemotong rumput, namun tetap menyatukan semua mobil sebagai satu kelas. Jika sebuah perusahaan memproduksi mobil dan mesin pemotong rumput, gambar pada klien fusi pertama yang diberi label dengan nama pabrikan harus dipisahkan menjadi mesin pemotong rumput atau mobil. Memisahkan label-label ini menggunakan matriks konfusi lintas situs mungkin tidak berfungsi dengan baik karena sejumlah besar gambar dalam label yang pertama akan dipetakan ke kedua label dalam penyelesaian masalah.

Jika data diunggulkan, pemeriksaan ruang fitur dan tumpang tindih antar label dalam ruang fitur dapat memberikan cara untuk mengatasi situasi tersebut. Himpunan semua fitur yang digabungkan menentukan ruang fitur, yaitu ruang multidimensi yang mencakup semua titik data yang menyediakan kemungkinan kumpulan masukan. Setiap label mencakup beberapa wilayah dalam ruang fitur ini, ditentukan oleh titik-titik berbeda yang sesuai dengan label keluaran tertentu. Untuk menjelaskan pendekatan ini, mari kita perhatikan kumpulan data yang ditunjukkan pada Gambar 4.8. Untuk memudahkan ilustrasi, hanya dua fitur yang ditampilkan. Kumpulan data pelatihan akan terdiri dari titik-titik format <nilai fitur 1, nilai fitur 2, label keluaran>. Pada Gambar 4.8, kita asumsikan ada tiga label yang diberi tanda X, Y, dan Z. Jika kita menelusuri kurva terdekat yang melingkupi titik-titik yang sesuai dengan masing-masing label, kurva tersebut akan menelusuri wilayah berbeda dalam ruang fitur. Sebuah label dapat dipetakan ke lebih dari satu wilayah di ruang fitur, seperti yang ditunjukkan oleh label

X di ruang fitur. Wilayah ini dapat diidentifikasi menggunakan berbagai algoritma pengelompokan.



Gambar 4.8 Ilustrasi ruang fitur.

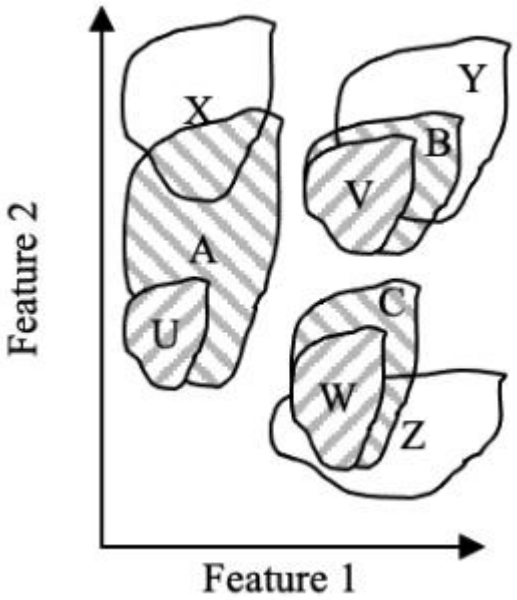


Gambar 4.9 Contoh ruang fitur dengan tiga klien fusi.

Dengan asumsi bahwa ruang fitur didefinisikan secara konsisten di seluruh klien fusi, tumpang tindih di antara wilayah berbeda di klien fusi berbeda dapat digunakan untuk menentukan korespondensi antar label. Contoh dengan tiga klien fusi dengan masing-masing tiga label ditunjukkan pada Gambar 4.8. Ketiga label tersebut mengukir wilayah yang ditandai dengan A, B, C pada klien fusi A, sebagai X, Y, Z pada klien fusi B, dan sebagai U, V, W pada klien fusi C. Wilayah pada setiap klien fusi ditandai dengan warna berbeda untuk menonjolkan perbedaannya.

Jika wilayah yang diberi label pada klien fusi berbeda saling bertumpukan, kita akan mendapatkan diagram seperti Gambar 4.10. Ketiga wilayah tersebut tumpang tindih satu sama lain dengan cara yang berbeda. Wilayah V, B, dan Y tampaknya saling tumpang tindih dengan cukup baik, dan dapat dianggap sebagai satu label tunggal. Wilayah C, W dan Z juga saling tumpang tindih, dan dapat dipandang sebagai satu label. Namun, A tumpang tindih

dengan U dan X, namun U dan X tidak tumpang tindih. Artinya mungkin A harus dipecah menjadi dua wilayah, yang satu sama dengan W dan yang lainnya sama dengan U.



Gambar 4.10: Superposisi ruang fitur tiga klien fusi.

Secara konseptual, jika sketsa ringkasan masing-masing wilayah menggunakan serangkaian fitur yang konsisten sebagai dimensi dikirim ke server fusi, maka server fusi dapat memeriksa tumpang tindih di antara wilayah setiap klien fusi, dan menentukan cara mendefinisikan kumpulan kanonik. label dengan mengelompokkan wilayah ke dalam kelompok yang berbeda.

Mari kita periksa bagaimana konsep tersebut dapat dicapai dalam praktik. Karena klien fusion tidak akan mengirimkan data mentah ke server fusion, mereka harus menyetujui ringkasan data mereka secara konsisten. Langkah pertama adalah agar semua klien fusi menggunakan serangkaian fitur yang konsisten, yang dapat dilakukan dengan menggunakan teknik yang dijelaskan di Bagian 4.1.2. Jika semua fitur telah diskalakan secara konsisten, serangkaian algoritme pengelompokan dapat dijalankan pada kumpulan fitur untuk mengidentifikasi wilayah dalam ruang fitur yang sesuai dengan masing-masing kelas yang diberi label.

Cluster yang teridentifikasi dapat direpresentasikan secara kompak, misalnya. sebagai pusat massa dan jari-jari cluster yang sesuai dengan masing-masing label. Ringkasan data yang ringkas dapat dikirim ke server fusi, yang dapat memeriksa tumpang tindih di antara label yang berbeda. Tumpang tindih ini dapat dilakukan dengan menjalankan algoritma pengelompokan putaran lain yang dapat dijalankan di server fusi dengan langkah-langkah berikut (i) memilih radius terkecil di antara semua cluster yang dilaporkan (ii) mewakili setiap cluster dengan radius lebih besar sebagai beberapa cluster, setiap cluster diwakili oleh sebuah centroid dengan ukuran radius yang dipilih untuk mencakup seluruh cluster asli dan (iii) menjalankan algoritma clustering di antara semua centroid untuk menemukan grup yang tumpang tindih.

Proses keseluruhannya akan sama dengan pendekatan yang ditunjukkan pada Gambar 4.9 dan 4.10. Tergantung pada algoritma pengelompokan yang dipilih, bentuk yang dihasilkan untuk setiap wilayah mungkin memiliki bentuk teratur dan bukan bentuk tidak beraturan yang digunakan dalam ilustrasi.

Beberapa algoritma pengelompokan bekerja lebih baik jika langkah lain, seperti analisis komponen utama, dilakukan sebelum proses pengelompokan. Jika algoritma seperti itu digunakan, seseorang harus menggunakan mekanisme yang konsisten untuk konversi ke komponen utama di seluruh klien fusi. Dalam kasus ini, penggunaan algoritme pembelajaran gabungan yang naif, seperti fusi fungsi di seluruh klien fusi, 3.3 dapat digunakan untuk menghasilkan pendekatan analisis komponen utama yang konsisten di seluruh klien.

4.3 MENGHILANGKAN DATA BERKUALITAS BURUK DAN BERNILAI RENDAH

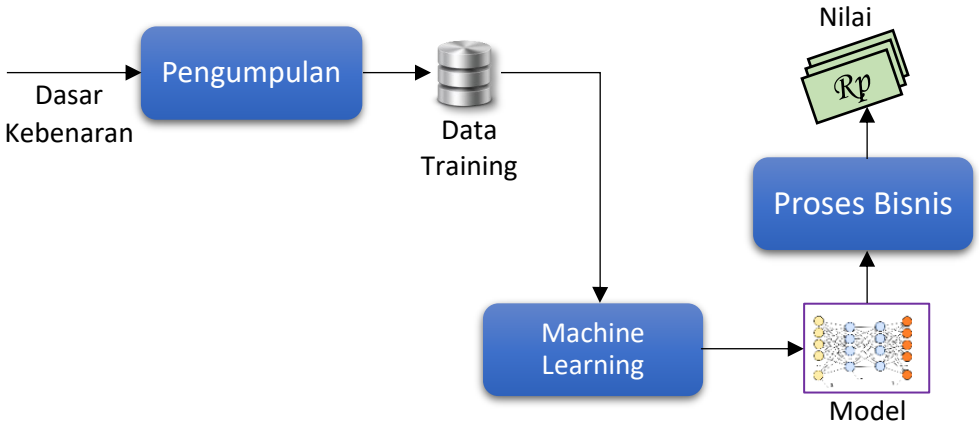
Salah satu tantangan dalam menggunakan data dari beberapa klien fusi adalah proses pengumpulan data dan prosedur pengelolaan data di semua klien fusi tidak harus konsisten. Hal ini menghasilkan data yang memiliki kualitas bervariasi di seluruh klien. Memasukkan data dengan kualitas buruk dapat menghasilkan model AI yang buruk. Oleh karena itu, salah satu tugas pra-pemrosesan adalah memahami klien yang memiliki kualitas data buruk, dan menghilangkan kejadian serupa.

Meskipun ada banyak definisi berbeda tentang kualitas data yang tersedia, dan banyak definisi kualitas bergantung pada jenis data tertentu, misalnya data kualitas data. pada data gambar atau citra satelit, definisi yang paling relevan untuk pembelajaran mesin dalam konteks bisnis diartikulasikan dalam dengan fokus pada fusi informasi sensor. Ini mendefinisikan dua istilah, Kualitas Informasi (QoI) dan Nilai Informasi (VoI), bersama dengan model informasi untuk menggambarkan atribut yang diperlukan untuk keduanya. QoI didefinisikan sebagai ukuran intrinsik terhadap informasi yang dapat diukur tanpa mengacu pada penggunaannya, yaitu tugas pengambilan keputusan yang memanfaatkan informasi tersebut. VoI diartikulasikan sebagai ukuran manfaat yang bergantung pada penggunaan informasi. QoI telah disempurnakan dan diterapkan lebih lanjut untuk berbagai domain aplikasi, termasuk jaringan komunikasi, jaringan sensor nirkabel, dan penginderaan massa seluler. VoI telah didefinisikan lebih lanjut dan diterapkan untuk Internet of Things, pengambilan data dan pembelajaran mesin untuk operasi koalisi.

Kualitas adalah properti intrinsik dari data yang dikumpulkan, dan kualitas dapat diperkirakan dengan memeriksa properti data yang tidak bergantung pada proses bisnis di mana data tersebut digunakan. Nilai data bergantung pada proses bisnis, yaitu bagaimana data tersebut akan digunakan. Nilainya juga bersifat inkremental, yaitu nilai suatu data bergantung pada data tambahan apa yang sudah tersedia untuk model atau proses bisnis. Jika kumpulan data baru merupakan duplikasi dari data yang sudah tersedia, maka kumpulan data tersebut mungkin tidak memberikan banyak nilai tambah meskipun kualitasnya sangat tinggi.

Untuk menentukan kualitas dan nilai data, operasi dalam siklus Learn Infer Act dapat dimodelkan seperti proses yang ditunjukkan pada Gambar 4.11. Ada beberapa fungsi yang berkaitan dengan proses bisnis yang perlu menggunakan model AI untuk menjalankan atau

meningkatkan fungsinya. Kebenaran dasarnya adalah fungsi di dunia nyata yang coba dimodelkan oleh proses pelatihan. Untuk mempelajari fungsi ini, data pelatihan dikumpulkan. Data yang dikumpulkan mengandung representasi kebenaran dasar, namun kesalahan mungkin terjadi selama proses pengumpulan.



Gambar 4.11: Proses abstrak pembelajaran mesin untuk definisi kualitas.

Akibatnya, data pelatihan mungkin mengkodekan fungsi yang berbeda dari kebenaran dasarnya. Model itu sendiri merupakan representasi alternatif dari fungsi tersebut. Ketika model tersebut digunakan dalam proses bisnis, sejumlah nilai dihasilkan. Nilai data pelatihan atau model adalah nilai yang dihasilkan oleh proses bisnis. Nilai dari proses bisnis dapat berupa pendapatan tambahan yang dikumpulkan, biaya yang dihemat, atau metrik terkait lainnya yang perlu dipantau dan ditingkatkan.

Kualitas data atau kualitas model adalah kemampuannya menangkap kebenaran dasar dengan benar. Kualitas data pelatihan dapat diukur dari seberapa dekat data tersebut dengan kebenaran sebenarnya. Kualitas model juga dapat diperkirakan berdasarkan seberapa dekat model tersebut dengan kebenaran sebenarnya. Kebenaran dasar, data pelatihan, dan model merupakan representasi berbeda dari fungsi yang sama, dan pada prinsipnya dapat dibandingkan untuk memperkirakan seberapa berbedanya keduanya. Semakin kecil perbedaannya, semakin baik kualitas datanya. Tantangan dalam sebagian besar penerapan bisnis praktis adalah kebenaran mendasarnya mungkin tidak diketahui. Akibatnya, pengukuran kualitas data secara absolut tidak mungkin dilakukan, dan tindakan proksi harus diterapkan. Nilai informasi adalah kontribusi yang diberikan pada fungsi nilai proses bisnis melalui data yang dikumpulkan. Nilai ini dapat bergantung pada beberapa faktor di luar model, serta perbedaan yang diberikan model melebihi nilai yang belum tersedia.

Pembahasan rinci tentang berbagai pendekatan untuk memperkirakan kualitas dan nilai informasi dapat ditemukan dalam referensi seperti. Tanpa membahas aspek yang lebih teknis dalam mengukur kualitas dan nilai, kita akan berasumsi bahwa ada pendekatan yang baik untuk mengukur kualitas dan nilai informasi. Kami akan fokus pada bagaimana metrik yang terkait dengannya dapat digunakan untuk meningkatkan data yang tersedia untuk pembelajaran gabungan.

4.3.1 Pemilihan Data Berbasis Reputasi

Salah satu pendekatan untuk meningkatkan kualitas data adalah dengan menetapkan metrik kualitas dan/atau nilai pada rincian klien fusi. Dalam hal ini, seseorang dapat menentukan apakah akan menyertakan data yang disediakan oleh klien fusi mana pun berdasarkan seberapa dapat dipercaya klien fusi tersebut. Dalam beberapa kasus, terdapat kurangnya kepercayaan di antara pihak-pihak yang berbeda atau beberapa klien fusi mungkin dianggap lebih dapat dipercaya dibandingkan yang lain. Misalnya, dalam koalisi antar negara yang berbeda, beberapa negara yang memiliki aliansi militer terpisah mungkin lebih percaya satu sama lain dibandingkan negara-negara anggota yang tidak tergabung dalam aliansi untuk menjaga kualitas data yang baik. Contoh spesifiknya adalah ketika negara-negara NATO melakukan misi penjaga perdamaian di Afrika, koalisinya dapat terdiri dari negara-negara anggota NATO dan negara-negara Afrika, namun negara-negara NATO mungkin tidak menganggap data dari anggota aliansi Afrika dapat diandalkan. Contoh lainnya adalah berbagi informasi terkait kesehatan dan penyakit di antara negara-negara yang berbeda untuk tugas seperti memerangi pandemi, dimana hampir semua negara di dunia akan bekerja sama, namun negara-negara kaya dengan infrastruktur layanan kesehatan yang lebih maju mungkin mempertimbangkan data. dari negara-negara dengan sumber daya yang lebih sedikit dan infrastruktur layanan kesehatan yang lebih buruk karena kurang dapat diandalkan.

Jika ada pendekatan untuk menetapkan reputasi ke klien fusi berbeda yang menyediakan data untuk proses pembelajaran gabungan, data dari klien fusi dengan reputasi buruk dapat dikeluarkan dari proses federasi. Server fusi mungkin atau mungkin tidak menghapus klien fusi secara eksplisit, memilih untuk membiarkan klien tersebut berpartisipasi tetapi mengabaikan model yang disediakan oleh klien bereputasi rendah selama proses fusi model. Pendekatan diam memberikan model kepada klien bereputasi rendah yang dibuat dengan menggabungkan model dari klien bereputasi tinggi. Ada banyak teknik untuk menetapkan reputasi untuk klien yang berbeda dalam konteks komputasi terdistribusi, dan sistem komputasi reputasi apa pun yang sesuai dapat digunakan.

Mekanisme berbasis reputasi untuk mengabaikan data atau model dari klien yang dipilih akan bekerja dengan baik di lingkungan di mana terdapat banyak situs dengan data, dan seseorang dapat membuat metrik perbandingan untuk menetapkan reputasi. Hal ini dapat berhasil dalam situasi seperti pembelajaran gabungan konsumen (Lihat 2.3.1) di mana terdapat ribuan atau jutaan klien yang memiliki data. Namun, dalam konteks pembelajaran gabungan perusahaan (Lihat 2.3.2), jumlah lokasi mungkin tidak cukup untuk menghapus semua data dari salah satu klien.

Karena banyaknya rincian yang diperlukan untuk pemilihan per lokasi, pemilihan data berbasis reputasi mungkin bukan pendekatan yang tepat untuk Pembelajaran Gabungan Perusahaan, yang merupakan cakupan utama buku ini.

4.3.2 Pemilihan Data Berbasis Nilai

Dalam pemilihan data berbasis nilai, data untuk pembelajaran gabungan dipilih dari klien fusi berdasarkan nilai yang dikontribusikan setiap titik data tambahan pada tugas pembuatan model. Penilaian nilai dapat dibuat berdasarkan kontribusi yang diberikan oleh

masing-masing klien dalam tugas pembelajaran gabungan. Ketika tugas utama difokuskan pada pelatihan model AI, salah satu perkiraan yang baik mengenai nilai kumpulan data adalah jumlah data baru yang dikontribusikan oleh masing-masing klien fusi.

Dalam analisis pembelajaran gabungan untuk operasi koalisi, analisis teoretis tentang nilai data yang diperoleh dari mitra koalisi yang berbeda diperoleh dalam beberapa kondisi yang disederhanakan. Analisis ini didasarkan pada eksplorasi apakah ada gunanya memperoleh data pelatihan baru dari mitra berdasarkan data yang tersedia saat ini untuk melatih model. Setiap mitra memiliki beberapa data yang berisik dan beberapa data yang bagus dan bebas gangguan. Analisis menunjukkan bahwa lebih baik mendapatkan data tambahan dari mitra selama mitra tersebut menyediakan beberapa data baru yang bebas gangguan di luar data yang sudah tersedia. Hal ini didasarkan pada asumsi bahwa untuk masing-masing mitra, data yang berisik merupakan sebagian kecil dari keseluruhan data. Oleh karena itu, jika ada data yang belum tersedia secara lokal, sebaiknya informasi tersebut diperoleh dari mitra lain.

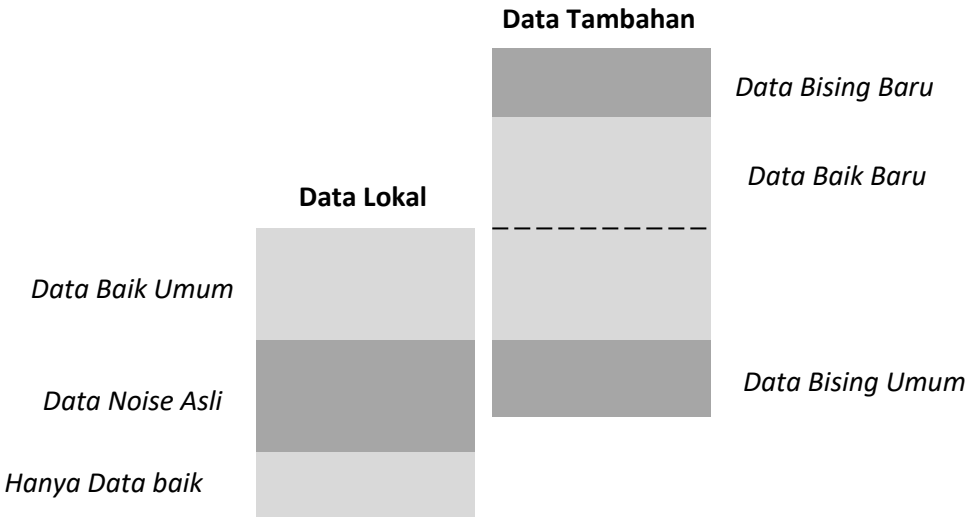
Situasinya ditunjukkan secara grafis pada Gambar 4.12. Diasumsikan bahwa terdapat beberapa data lokal di satu situs, dan situs tersebut memiliki pilihan untuk menerima data dari situs mitra. Wilayah yang diarsir menunjukkan bagian data yang bermasalah, baik secara lokal maupun di situs mitra. Meskipun data yang mengandung noise merupakan bagian yang relatif kecil, angka tersebut akan melebih-lebihkan jika dibandingkan dengan data bagus yang tidak memiliki shade. Sebagian dari data bagus yang tersedia di mitra dan sebagian dari data bermasalah yang tersedia di mitra adalah umum karena data tersebut ada secara lokal. Namun, ada beberapa bagian data yang bagus dan tidak disajikan secara lokal. Jika sejumlah besar data bagus diperoleh dari mitra, akan bermanfaat jika memperoleh data tersebut dari mitra.

Tantangan dalam kehidupan nyata adalah tugas menentukan seberapa banyak data yang umum dan seberapa banyak data dari mitra yang baru. Kita perlu mengambil keputusan ini tanpa bertukar data satu sama lain. Ide dasar untuk menilai kesamaan antara data yang ada di lokasi berbeda akan serupa dengan yang dijelaskan dalam analisis ruang fitur untuk menyelesaikan konflik nilai yang dijelaskan di Bagian 4.2.4. Label dapat diabaikan untuk tujuan menentukan nilai data baru, dan ukurannya akan menjadi area tambahan pada ruang fitur yang dicakup oleh data milik mitra baru.

Penilaian atas nilai yang dikontribusikan oleh mitra dapat diukur berdasarkan seluruh data yang tersedia dari mitra, atau berdasarkan subset data terpilih yang disediakan oleh mitra. Pemilihan subset dapat dilakukan berdasarkan label, misalnya seseorang dapat mengumpulkan data milik label tertentu dari mitranya hanya jika mitra tersebut memiliki beberapa data baru yang belum disediakan oleh pihak lain, pada dasarnya menerima data hanya jika jumlah poin milik label bertambah. Dalam hal ini, beberapa label dari mitra diterima sementara label lain dari mitra yang sama dapat ditolak.

Penilaian nilai juga dapat dilakukan pada partisi data secara acak untuk memilih titik data yang tidak memiliki tumpang tindih yang besar. Dalam hal ini, sebelum proses pelatihan dimulai, masing-masing klien akan membagi datanya menjadi beberapa partisi dengan ukuran

yang telah ditentukan, dan nilai yang ditawarkan oleh partisi data tersebut dibandingkan dengan data yang tersedia dari klien lain akan diberikan. Hal ini akan menghilangkan partisi data yang tidak memberikan nilai baik atau sudah tersedia dari mitra lain.



Gambar 4.12: Pemilihan data berdasarkan nilai dari mitra.

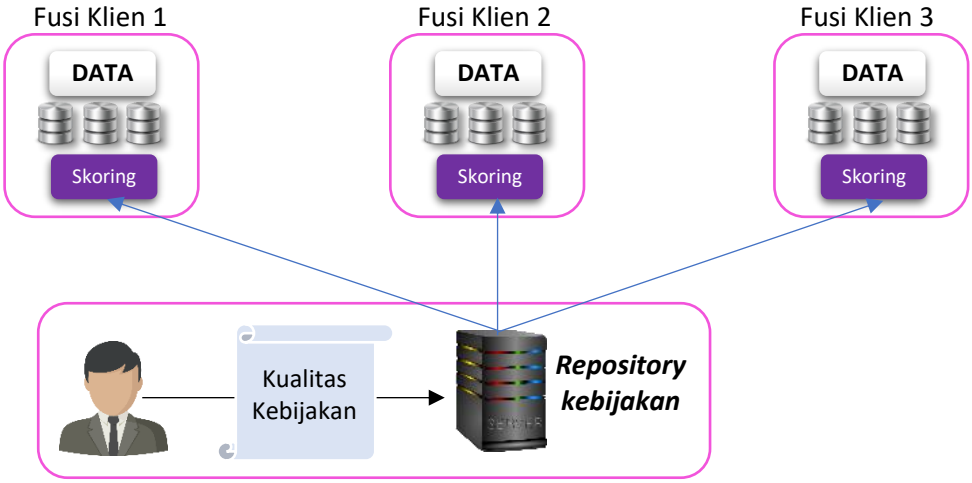
4.3.3 Peningkatan Kualitas Berbasis Kebijakan

Metode penilaian kualitas berbasis kebijakan dapat digunakan untuk memilih bagian data dari klien fusi terpilih yang tidak sesuai dengan persyaratan kualitas. Kebijakan yang menentukan metrik berbeda untuk kualitas data dapat ditentukan di server fusi. Kebijakan citegrueneberg 2019 arsitektur kualitas data berbasis kebijakan memiliki beberapa keunggulan dibandingkan cara lain untuk mendefinisikan kualitas data. Hal ini memungkinkan definisi yang fleksibel dan dapat dengan mudah disesuaikan untuk tugas pembuatan model AI tertentu, aplikasi proses bisnis tertentu, dan bahkan dapat menyediakan sarana otomatis untuk menyesuaikan kebijakan untuk klien fusi yang berbeda.

Kebijakan kualitas data memberikan skor kualitas pada setiap bagian data berdasarkan berbagai aturan. Aturan-aturan ini disusun untuk memperoleh skor kualitas pada berbagai kategori, seperti kelengkapan data, konsistensi data, validitas data, atau keunikan seluruh data. Skor kualitas akhir diperoleh dengan menggabungkan skor pada masing-masing kategori.

Perasaan intuitif untuk berbagai kategori dapat diperoleh dengan memetakan atribut ini ke data yang disimpan dalam representasi tabel, seperti spreadsheet atau database relasional. Setiap baris dalam tabel adalah catatan atau titik data. Kelengkapan akan menunjukkan bahwa nilai tidak hilang dari kolom tertentu dalam catatan. Konsistensi akan mencakup aturan-aturan yang memerlukan batasan pada nilai-nilai dalam suatu catatan, atau pada nilai-nilai di berbagai jenis catatan. Validitas akan memberikan batasan pada entri dalam setiap entri dalam tabel, dan aturan keunikan akan mengidentifikasi nilai-nilai yang harus unik di seluruh catatan data. Berdasarkan kepatuhan terhadap kebijakan kualitas data, skor kualitas data dapat ditetapkan di seluruh data yang tersedia di klien fusi, atau pada bagian

tertentu dari catatan data di klien. Demikian pula, kebijakan untuk menerima data berdasarkan kualitas dapat ditentukan.



Gambar 4.13: Peningkatan kualitas berbasis kebijakan.

Pendekatan berbasis kebijakan dapat digunakan untuk menentukan ambang batas kumpulan data yang dapat diterima dari berbagai klien yang terlibat dalam proses pembelajaran gabungan. Kebijakan ditentukan secara terpusat dan disimpan ke dalam repositori kebijakan di server fusi. Kebijakan ini memungkinkan skor kualitas ditentukan untuk setiap kumpulan data. Setiap klien fusi akan mengambil kebijakan yang relevan dengan kebutuhan mereka dari penyimpanan kebijakan, dan menggunakannya untuk menghitung skor kualitas untuk setiap kumpulan data. Kumpulan data yang melewati ambang batas kualitas yang dapat diterima akan menjadi satu-satunya kumpulan data yang akan digunakan setiap klien untuk melatih modelnya.

Arsitektur pendekatan berbasis kebijakan ini ditunjukkan pada Gambar 4.13. Pencetak skor data berbasis kebijakan melihat berbagai partisi data lokal dan memberikan skor kualitas kepada mereka berdasarkan kebijakan yang diambil dari lokasi pusat. Pencetak skor dapat menentukan subset data lokal yang seharusnya digunakan dalam tugas pembelajaran gabungan.

4.4 RINGKASAN

Dalam bab ini, kami telah mengkaji pendekatan yang dapat menangani masalah ketidakcocokan data yang muncul dalam konteks pembelajaran di seluruh kumpulan data yang didistribusikan ke klien fusi yang berbeda. Kami telah mengidentifikasi berbagai jenis ketidakcocokan yang mungkin terjadi di berbagai klien fusi. Hal ini mencakup perbedaan format, perbedaan cara penamaan konsep yang sama pada klien fusi yang berbeda, dan perbedaan kualitas dan nilai data yang tersedia pada klien fusi yang berbeda. Kami telah mengeksplorasi bagaimana seseorang dapat mengatasi ketidaksesuaian ini dengan menggunakan konstruksi seperti grafik konversi, matriks kebingungan lintas situs, definisi

aturan transformasi data yang terpusat, dan penerapan serangkaian kebijakan kualitas yang konsisten.

Penerapan pembelajaran gabungan dalam konteks realistis apa pun dengan perbedaan data akan memerlukan penggunaan konstruksi ini. Beberapa dari konstruksi ini sendiri bergantung pada menjalankan algoritme pembelajaran mesin tambahan, seperti pengelompokan sebagai sub-komponen. Penciptaan solusi apa pun akan memerlukan banyak komponen dasar untuk disusun menjadi alur kerja end-to-end yang disesuaikan untuk masalah bisnis, dan rangkaian algoritma manajemen data yang tepat untuk dipilih dan diterapkan sebelum tugas sebenarnya. pembelajaran federasi dimulai.

BAB 5
MENGATASI KEMIRINGAN DATA DALAM PEMBELAJARAN FEDERASI

Kemiringan data mengacu pada situasi di mana data pelatihan yang tersedia untuk pembelajaran pada klien fusi berbeda memiliki karakteristik yang sangat berbeda. Dalam bab ini, kita akan melihat pendekatan-pendekatan untuk mengatasi permasalahan yang disebabkan oleh ketimpangan tersebut. Untuk keperluan bab ini, kami berasumsi bahwa konsistensi data dan prosedur normalisasi data yang dijelaskan dalam Bab 4 telah diterapkan. Secara khusus, masalah yang berhubungan dengan format data, normalisasi yang konsisten di seluruh klien fusi, dan pengelolaan kualitas data telah ditangani. Sebagai hasil dari langkah-langkah ini, setiap klien fusi akan memiliki data dalam format yang sama, dan dinormalisasi secara konsisten. Data di semua klien fusi memiliki kualitas sebanding yang dapat diterima oleh semua orang. Kami juga akan berasumsi bahwa klien fusi memiliki kepercayaan implisit satu sama lain dan situs server data. Asumsi ini realistis untuk sebagian besar kondisi perusahaan. Namun demikian, di Bab 6, kita akan membahas beberapa pendekatan untuk menangani situasi ketika kepercayaan ini mungkin terbatas.

Jika kita beruntung dan data didistribusikan secara acak ke seluruh klien fusi, salah satu pendekatan yang dijelaskan di Bab 3 akan bekerja dengan baik. Di sebagian besar lingkungan nyata, distribusi data di seluruh klien fusi akan menjadi tidak seimbang. Oleh karena itu, kita harus memiliki pendekatan untuk membuat model bekerja dengan baik meskipun distribusi datanya miring. Salah satu jenis ketimpangan data mungkin terjadi karena klien fusi yang berbeda mungkin mengumpulkan data yang dikumpulkan berdasarkan asumsi dasar yang berbeda. Jenis data yang miring lainnya dapat terjadi karena data yang dikumpulkan di beberapa klien mungkin kehilangan beberapa kelas yang diperlukan. Salah satu kasus yang sangat menantang untuk pembelajaran mesin adalah ketika data dipartisi sepenuhnya di antara klien fusi yang berbeda.

Untuk menjelaskan partisi dalam data unggulan, mari kita pertimbangkan struktur data yang disimpan dalam format tabel. Data tersebut terdiri dari beberapa baris dan kolom. Dalam permasalahan seperti klasifikasi atau pemetaan, terdapat satu kolom khusus (kolom keluaran) yang berisi kelas atau label entri. Mari kita asumsikan bahwa versi data terpusat akan memiliki konten seperti yang ditunjukkan pada Tabel 5.1. Dalam tabel ini, 6 record ditampilkan untuk tujuan ilustrasi, dimana data tersebut termasuk dalam tiga kelas output dengan label L_0, L_1 dan L_3. Data terdiri dari tiga fitur F_1, F_2 dan F_3, dan indeks menunjukkan nomor berbeda untuk setiap catatan.

Data terpusat akan diperoleh jika semua data dari semua klien fusi dikumpulkan di lokasi pusat, catatan data yang berlebihan dihilangkan dan indeks tunggal dibuat untuk mengidentifikasi setiap catatan secara unik. Meskipun sentralisasi seperti itu tidak mungkin dilakukan di banyak lingkungan kehidupan nyata, yang merupakan alasan utama AI terfederasi, eksperimen pemikiran dalam mengumpulkan data akan berguna untuk tujuan ilustrasi. Ketika data terdapat pada klien fusi yang berbeda, kemungkinan besar terjadi

redundansi data di antara klien fusi yang berbeda, dan kecil kemungkinannya terdapat indeks yang konsisten di seluruh klien fusi. Namun kami akan menggunakan indeks untuk mengilustrasikan identitas unik hipotetis dari setiap catatan. Data pada masing-masing klien fusi yang berbeda akan menjadi subset dari data terpusat yang ditunjukkan pada Tabel 5.1.

Satu kemungkinan pemisahan data di antara tiga klien fusi yang berbeda dapat dilihat pada Tabel 5.2. Ini adalah partisi data yang relatif bagus. Masing-masing klien fusi memiliki data milik masing-masing dari tiga kelas keluaran berlabel. Setiap klien fusi memiliki semua fitur yang ada. Entri indeks tidak akan ada pada klien fusi mana pun dalam lingkungan nyata, atau setidaknya akan independen. Namun, data ini dianggap memiliki korelasi yang mudah dengan data terpusat pada Tabel 5.1.

Tabel 5.1: Contoh data yang dipartisi.

Indeks	F_1	F_2	F_3	Keluaran
1	A	X	0,3	L_1
2	B	Y	0,2	L_2
3	C	Z	0,3	L_2
4	A	X	0,4	L_3
5	B	X	0,8	L_1
6	B	Y	0,9	L_3

Tabel 5.2: Partisi yang bagus misalnya untuk data yang dipartisi.

Klien Fusi A				
Indeks	F_1	F_2	F_3	**Keluaran**
1	A	X	0,3	L_1
2	B	Y	0,2	L_2
4	A	X	0,4	L_3

Klien Fusi B				
Indeks	F_1	F_2	F_3	**Keluaran**
3	C	Z	0.3	L_2
4	A	X	0.4	L_3
5	B	X	0.8	L_1

Klien Fusi C				
Indeks	F_1	F_2	F_3	**Keluaran**
1	A	X	0.3	L_1
2	B	Y	0.2	L_2
6	B	Y	0.9	L_3

Tabel 5.3: Contoh partisi data.

Klien Fusi A				
Indeks	F_1	F_2	F_3	**Keluaran**
1	A	X	0.3	L_1

5	B	X	0.8	L_1

Klien Fusi B				
Indeks	F_1	F_2	F_3	**Keluaran**
2	B	Y	0.2	L_2
3	C	Z	0.3	L_2

Klien Fusi C				
Indeks	F_1	F_2	F_3	**Keluaran**
4	A	X	0.4	L_3
6	B	Y	0.9	L_3

Jika kumpulan data dipartisi seperti yang ditunjukkan pada Tabel 5.2, algoritma pembelajaran gabungan yang dijelaskan pada Bab 3 akan bekerja dengan cukup baik. Meskipun data dibagi menjadi beberapa klien fusi, data di setiap klien fusi memiliki beberapa tumpang tindih berpasangan dengan data yang ada di semua klien lainnya, setiap kelas keluaran ada di setiap klien fusi, dan kemungkinan besar mereka mempelajari fungsi yang sama.

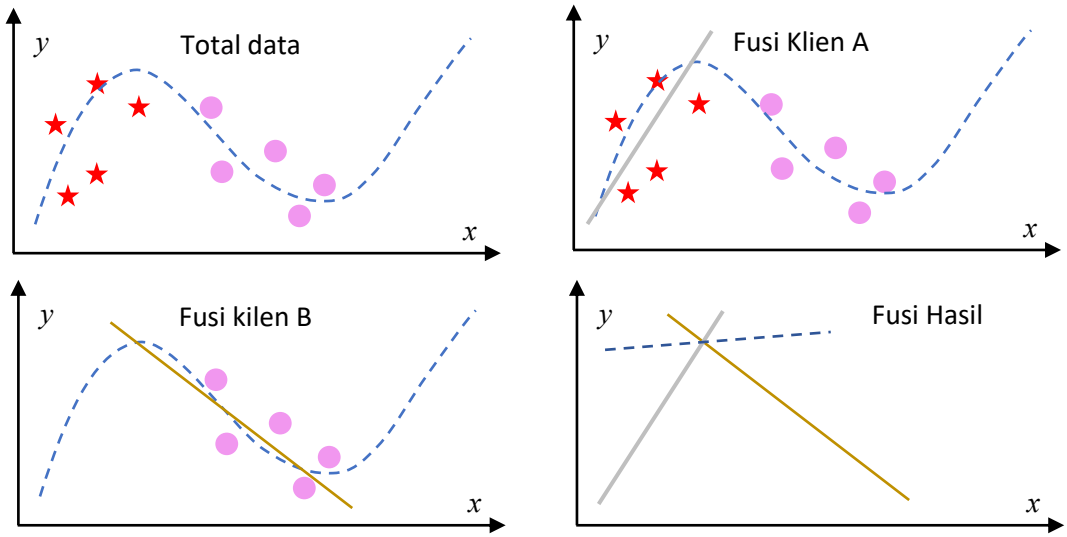
Dalam kehidupan nyata, sering terjadi skenario di mana pembagian data di seluruh klien fusi tidak selalu dirancang agar sesuai dengan algoritme yang digunakan untuk pembelajaran gabungan. Salah satu jenis situasi spesifik yang muncul adalah ketika data dibagi menjadi klien fusi yang berbeda sehingga data pada setiap klien fusi kehilangan beberapa kelasnya. Bentuk ekstrimnya ditunjukkan pada Tabel 5.3. Dalam distribusi data khusus di antara klien, masing-masing klien fusi memiliki data yang hanya dimiliki satu kelas. Ketika data dipartisi dengan cara ini, algoritma naif untuk federasi gagal bekerja dengan baik. Pemartisian tidak perlu terlalu ekstrim untuk mempengaruhi kinerja, ada pola partisi data lain yang dapat menimbulkan tantangan serupa bagi federasi, misalnya. beberapa kelas hanya ada di satu klien fusi, atau setiap klien fusi mungkin memiliki beberapa kelas tetapi beberapa klien fusi kehilangan sebagian besar kelas. Partisi data serupa juga dapat terjadi pada data tanpa fitur. Mungkin ada klien fusi yang kehilangan data milik beberapa label keluaran.

5.1 DAMPAK DATA YANG DIPARTISI DAN TIDAK SEIMBANG

Di bagian ini, kita akan memeriksa dampak data yang dipartisi atau tidak seimbang dalam beberapa situasi, dan membahas penyebab mendasar mengapa partisi data menyebabkan masalah dalam tugas pembelajaran gabungan.

5.1.1 Masalah Kemiringan Data dalam Estimasi Fungsi

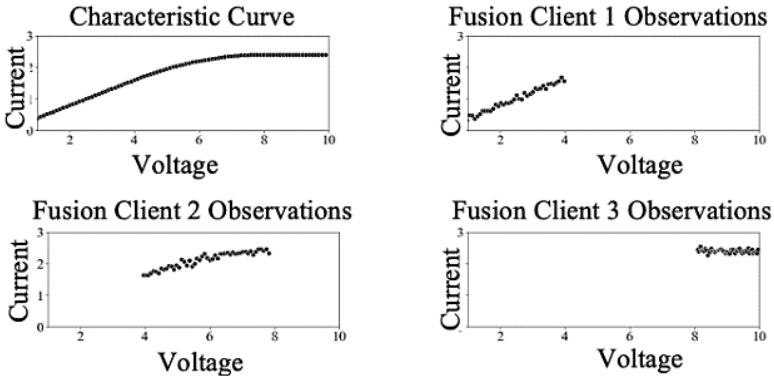
Salah satu cara untuk memahami data skew pada data terdistribusi adalah dengan mengeksplorasi masalah dari perspektif estimasi fungsi. Fungsi di dunia nyata sangatlah kompleks, dan klien fusi yang berbeda mungkin mengumpulkan titik data yang beroperasi dalam kondisi berbeda saat fungsi tersebut diukur. Ketika klien fusi yang berbeda mempelajari fungsi yang berbeda, algoritme naif tidak akan mampu menyusun atau merepresentasikan fungsi tersebut dengan benar.



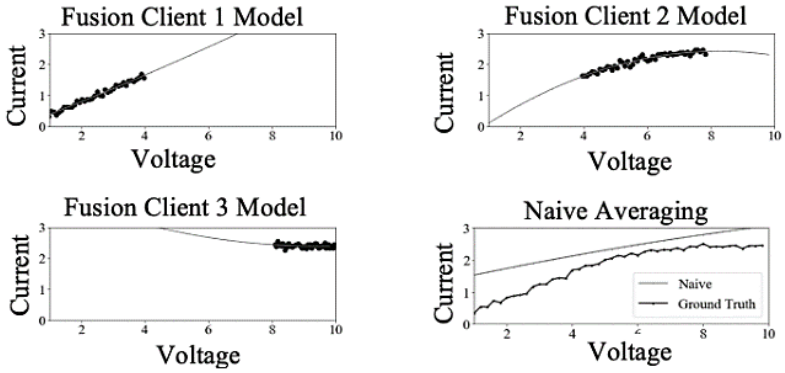
Gambar 5.1: Tampilan data yang berbeda.

Contoh di mana klien fusi yang berbeda mungkin mempelajari fungsi yang berbeda diilustrasikan pada Gambar 5.1. Gambar tersebut memiliki empat diagram, dan menunjukkan situasi yang mungkin muncul di dunia nyata ketika data dikumpulkan secara independen pada dua klien fusi yang berbeda. Semua data yang dikumpulkan di dua klien fusi beserta kebenaran dasarnya ditampilkan dalam diagram di kiri atas. Kebenaran dasar adalah hubungan sebenarnya yang harus dipertahankan antara masukan (x) dan keluaran (y) dan ditunjukkan dengan garis putus-putus. Untuk memudahkan ilustrasi, hanya satu masukan yang ditampilkan, namun pembahasannya dapat dengan mudah dilihat untuk menggeneralisasi masukan yang lebih kompleks dengan beberapa ciri. Bintang menandai titik data yang tersedia di salah satu klien fusi (klien A) dan lingkaran menandai titik data yang tersedia di satu klien fusi lainnya (klien B).

Ketika salah satu dari dua klien fusi memperkirakan fungsi tersebut cocok dengan datanya, fungsi yang dipelajari akan didasarkan pada kumpulan data yang tersedia di klien fusi tersebut. Pada data lokalnya sendiri, klien fusi A akan mempelajari fungsi yang meningkat secara linier antara x dan y, seperti yang ditunjukkan dengan garis abu-abu tebal di diagram kanan atas. Merupakan harapan alami bahwa estimasi fungsi memungkinkan sistem untuk memperkirakan hubungan antara input dan output. Demikian pula, klien fusi B, ketika memperkirakan suatu fungsi hanya menggunakan data lokalnya, akan mempelajari fungsi penurunan linier, seperti yang ditunjukkan dengan garis abu-abu putus-putus tebal di gambar kiri bawah. Jika teknik rata-rata gabungan digunakan pada data ini, hasil akhirnya adalah fungsi linier yang ditunjukkan dalam garis putus-putus hitam di gambar kanan bawah. Penjumlahan kedua fungsi tersebut tidak akan terlihat seperti kebenaran dasarnya.



Gambar 5.2: Ilustrasi kemiringan data untuk estimasi fungsi.



Gambar 5.3: Dampak ketimpangan data pada estimasi fungsi.

Masalah muncul dari fakta bahwa kedua klien fusi mengumpulkan data dalam kondisi berbeda. Hasilnya, setiap klien fusi memiliki data yang mewakili fungsi berbeda. Fungsi pada masing-masing klien fusi adalah hubungan yang valid antara masukan dan keluaran, namun setiap hubungan berlaku pada kondisi yang berbeda. Pada contoh yang ditampilkan, kondisinya adalah kisaran nilai masukan. Rata-rata sederhana dari fungsi-fungsi yang dipelajari (yaitu model), dengan asumsi bahwa semuanya mewakili hubungan yang sama, mungkin tidak selalu merupakan pendekatan yang tepat untuk dilakukan.

Mari kita perhatikan contoh dalam pemodelan terdistribusi, yaitu perilaku komponen elektronik. Situasinya ditunjukkan pada Gambar 5.2. Ada empat sub-gambar pada gambar, kiri atas menunjukkan kurva karakteristik komponen elektronik yang merupakan kebenaran dasar, dan tiga lainnya menunjukkan titik data yang dikumpulkan oleh klien fusi berbeda yang menguji komponen tersebut. Eksperimen di masing-masing klien fusi berfokus pada salah satu wilayah pengoperasian komponen elektronik. Akibatnya, kurva karakteristik yang diprediksi oleh masing-masing klien fusi akan sangat berbeda.

Hasil dari algoritma pembelajaran federasi naif yang dijelaskan pada Bab 3 pada masing-masing model ini dapat menjadi tidak tepat sasaran karena data di lokasi yang berbeda terkait dengan wilayah yang berbeda dan karakteristik yang berbeda. Alasannya serupa dengan yang diilustrasikan pada Gambar 5.1. Hasil penyesuaian fungsi pada tiga kumpulan data yang berbeda dan menggunakan pembelajaran gabungan yang naif ditunjukkan pada

Gambar 5.3. Gambar tersebut menunjukkan empat subdiagram. Klien fusi 1 telah mengumpulkan data untuk ujung bawah kurva karakteristik, dan memperkirakan hubungan antara masukan (Arus) dan keluaran (Tegangan) yang merupakan hubungan peningkatan linier.

Klien Fusion 2 telah mengumpulkan data dari tengah rentang operasi komponen dan telah mempelajari hubungan yang paling dekat dengan kebenaran dasar di antara masing-masing dari tiga klien. Klien Fusion 3 telah mempelajari hubungan yang menunjukkan penurunan hubungan antara arus dan tegangan. Dengan kemungkinan pengecualian pada klien 2, tidak ada klien yang memiliki perkiraan yang masuk akal mengenai kebenaran dasar. Ketika hasil dari semua klien dirata-ratakan, seseorang mendapatkan hasil yang tidak sesuai dengan kebenaran dasarnya. Untuk menggabungkan fungsi estimasi dengan benar, kita perlu memasukkan kesadaran bahwa klien fusi yang berbeda mempelajari fungsi yang dapat diterapkan untuk rentang masukan yang berbeda ke dalam tugas menggabungkan model.

5.1.2 Masalah Partisi Label dalam Klasifikasi

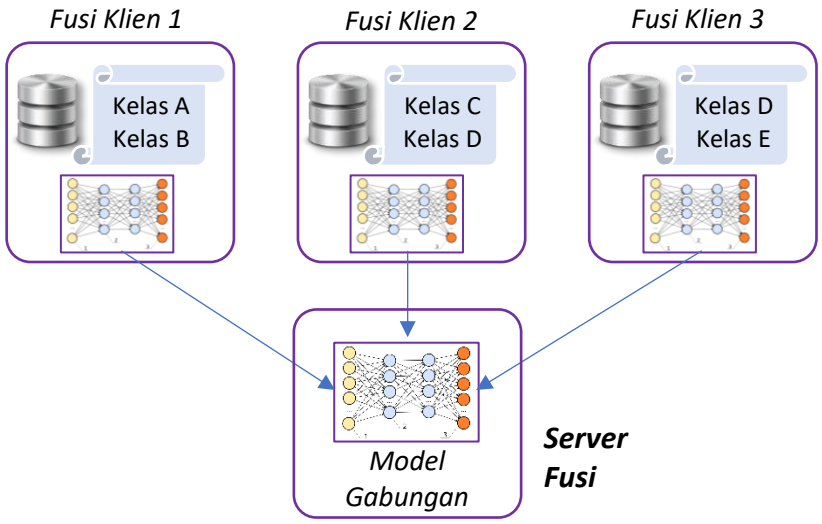
Klasifikasi adalah tugas yang sangat umum dalam fungsi yang mendukung AI, yang terutama berkaitan dengan tugas memetakan masukan apa pun ke dalam satu atau lebih kelas. Ada banyak contoh proses bisnis yang berguna berdasarkan klasifikasi, yang mencakup hal berikut, dan masih banyak lagi:

1. mengklasifikasikan pemohon kartu kredit ke dalam tingkat risiko yang berbeda, misalnya. 5 kelas penolakan pasti, risiko tinggi, risiko sedang, risiko rendah, dan penerimaan pasti
2. mengklasifikasikan pelamar kerja ke dalam kategori yang berbeda, pasti menerima, mengkaji lebih lanjut, dan pasti menolak
3. mengklasifikasikan gambar produk di jalur perakitan ke dalam berbagai kategori cacat, kualitas baik, atau memerlukan pemeriksaan lebih lanjut
4. mengklasifikasikan permintaan yang tiba di situs web ke dalam kategori berbeda yaitu pelanggan bernilai tinggi yang cenderung melakukan pembelian, pelanggan dengan prioritas rendah yang hanya menelusuri, atau netral mengklasifikasikan gambar MRI pasien untuk mendiagnosis berbagai kelas penyakit.

Mengingat penerapan klasifikasi yang luas dalam banyak kasus penggunaan bisnis, beberapa algoritma untuk klasifikasi yang bekerja dengan baik ketika data dipusatkan telah dikembangkan. Klasifikasi dapat dilakukan dengan cara yang diawasi atau tidak diawasi. Dalam klasifikasi yang diawasi, tersedia beberapa contoh kumpulan data pelatihan dengan label, dan model AI mempelajari cara membedakan kelas-kelas yang berbeda berdasarkan data pelatihan. Dalam klasifikasi tanpa pengawasan, data tidak diberi label, namun kelompok gambar serupa dapat ditentukan dan diberi label. Tantangan tambahan dengan data yang miring berlaku untuk klasifikasi yang diawasi dan tidak diawasi, namun akan lebih mudah dijelaskan dalam kaitannya dengan masalah klasifikasi yang diawasi.

Biasanya, jika tugas memerlukan data untuk diklasifikasi ke dalam sejumlah N kelas, dan setiap klien fusi memiliki beberapa instance dari masing-masing kelas, algoritme pembelajaran gabungan naif yang dijelaskan di Bab 3 akan berfungsi dengan baik. Namun,

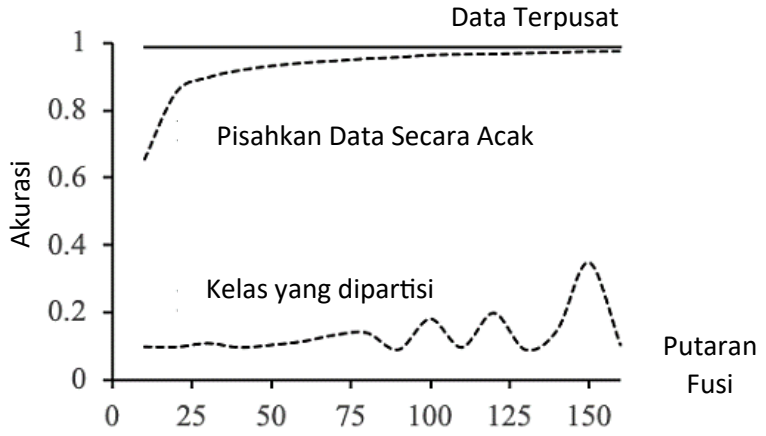
ketika kelas hilang dari sekumpulan klien fusi, algoritme pembelajaran gabungan yang naif menjadi kurang mahir. Situasi spesifik diilustrasikan pada Gambar 5.4, yang menunjukkan tiga klien fusi dengan data dan labelnya. Klien fusi pertama memiliki data yang termasuk dalam dua kelas, A dan B, klien fusi kedua memiliki data yang termasuk dalam dua kelas lain, C dan D, dan klien fusi ketiga memiliki data yang termasuk dalam dua kelas lainnya, D dan E. Setiap klien fusi melatih model AI pada datanya sendiri, dan kemudian mengirimkannya ke server fusi untuk digabungkan sesuai konfigurasi pada Gambar 3.1.



Gambar 5.4 Ilustrasi kecondongan data dalam klasifikasi.

Masalah dalam situasi seperti yang ditunjukkan pada Gambar 5.4 adalah setiap klien fusi melatih model sehingga data apa pun dipetakan ke dalam salah satu label yang tersedia di data lokal. Model AI untuk klasifikasi belajar memetakan masukan apa pun ke salah satu kelas tempat mereka dilatih. Mereka biasanya tidak memiliki kemampuan untuk mendeteksi fakta bahwa suatu input data mungkin termasuk dalam kelas baru.

Dalam kasus khusus ini, model yang dilatih pada klien fusi 1 akan memetakan masukan apa pun ke label A dan B, klien fusi 2 akan melatih modelnya sehingga masukan apa pun diklasifikasikan ke dalam label C dan D, dan klien fusi 3 akan mengklasifikasikan masukan apa pun ke dalam label label D dan E. Satu-satunya kelas yang datanya ada di lebih dari satu klien fusi adalah kelas D. Selain input yang dipetakan ke kelas D, tidak ada dua model yang akan memprediksi kelas yang sama untuk input apa pun. Dengan probabilitas yang tinggi, input apa pun yang termasuk dalam kelas A akan diklasifikasikan sebagai A oleh model yang dilatih oleh klien fusi 1, namun input tersebut akan diklasifikasikan ke dalam salah satu kelas C atau D oleh model di klien fusi 2, dan itu akan diklasifikasikan ke dalam D atau E berdasarkan model yang dilatih di klien fusi 3. Ketiga model tersebut telah mempelajari fungsi yang sangat berbeda, dan membuat rata-ratanya dengan cara yang bermakna akan sangat sulit.



Gambar 5.5 Dampak ketimpangan data terhadap klasifikasi.

Perilaku model yang hanya memprediksi apa yang telah dilatihnya merupakan hal yang umum terjadi di banyak jenis model AI, termasuk jaringan saraf. Bahkan ketika arsitektur jaringan saraf sama, yaitu mereka menggunakan jumlah level neuron yang sama, dan jumlah neuron yang sama di setiap level yang saling terhubung dalam jaringan yang identik, rata-rata parameter yang naif di seluruh klien fusi yang berbeda dalam sebuah skenario seperti yang ditunjukkan pada Gambar 5.4 kemungkinan besar tidak akan menghasilkan model yang baik. Alasannya adalah setiap klien fusi melatih jaringan sarafnya untuk menyesuaikan bobot guna mempelajari kelas fungsi yang berbeda, yang memetakan masukan ke kelas yang berbeda. Rata-rata bobot pada berbagai fungsi akan menghasilkan sesuatu yang aneh karena tidak ada alasan mendasar untuk merata-ratakan bobot tersebut agar menghasilkan sesuatu yang bermakna. Alasan mendasar untuk membuat rata-rata bobot jaringan saraf dan membuatnya berfungsi adalah bahwa sistem mempelajari fungsi yang sama di semua klien fusi, dan pandangan berbeda dari fungsi yang berbeda dapat dirata-ratakan bersama-sama untuk mendapatkan hasil yang bermakna.

Diskusi tentang dampak hilangnya kelas di seluruh klien fusi yang berbeda untuk pembelajaran gabungan di antara lembaga federal yang berbeda [70] dapat berguna untuk menggambarkan tantangan dalam pembelajaran gabungan ketika kelas dipartisi. Situasi spesifik yang dipertimbangkan adalah masalah klasifikasi di mana data dibagi menjadi sepuluh kelas unik, yang dibagi menjadi sepuluh klien fusi berbeda (lembaga federal dalam kasus makalah asli). Ketika masing-masing klien fusi memiliki beberapa data milik masing-masing dari sepuluh kelas, algoritma pembelajaran gabungan yang naif bekerja dengan sangat baik. Namun, ketika data dipecah sehingga setiap kelas hanya hadir di salah satu klien fusi, performa model yang dihasilkan menjadi sangat buruk.

Tantangan yang ditimbulkan oleh kelas yang dipartisi ditunjukkan pada Gambar 5.5. Efektivitas klasifikasi ke dalam sepuluh kelas diukur dengan menggunakan tiga skenario berbeda. Dalam setiap skenario, efektivitas model diukur dengan membuat model pada kumpulan data pelatihan, dan mengevaluasinya pada kumpulan data pengujian yang berbeda. Kumpulan data pelatihan identik di seluruh skenario. Dalam skenario pertama, semua data disimpan di lokasi terpusat, yaitu data dari masing-masing sepuluh klien fusi yang

berpartisipasi dikumpulkan di lokasi pusat dan model dilatih mengenai hal tersebut. Dalam skenario kedua, data pelatihan tidak dipindahkan ke lokasi pusat, tetapi diasumsikan bahwa setiap klien fusi memiliki partisi data pelatihan secara acak, yaitu setiap klien fusi memiliki beberapa instance milik masing-masing dari sepuluh kelas. Algoritme rata-rata gabungan, sebuah contoh dari algoritma pembelajaran gabungan naif yang dibahas di Bab 3, digunakan dan federasi dilakukan dalam beberapa putaran rata-rata pada berbagai contoh model. Dalam skenario ketiga, algoritma yang sama digunakan, namun data pelatihan didistribusikan sehingga masing-masing dari sepuluh klien fusi memiliki data yang hanya dimiliki oleh satu kelas.

Dampak dari partisi ini terhadap kinerja algoritme pembelajaran gabungan yang naif sangatlah buruk. Model gabungan tidak mampu berkinerja baik meskipun dilakukan beberapa putaran rata-rata gabungan. Akibatnya, pendekatan penggabungan model menggunakan rata-rata gabungan tidak berhasil. Akan ada data yang dipartisi di hampir setiap lingkungan bisnis kehidupan nyata. Oleh karena itu, penting untuk mengeksplorasi pendekatan yang dapat menangani dampak dari partisi data yang berbeda. Dalam dua bagian selanjutnya dari bab ini, kita akan melihat beberapa pendekatan ini.

5.2 PERTUKARAN DATA TERBATAS

Motivasi untuk tidak berbagi data dan menggunakan pembelajaran gabungan bervariasi tergantung pada konteks aplikasi, dan dalam beberapa kasus, pertukaran data apa pun mungkin tidak diizinkan karena masalah keamanan atau privasi. Dalam kasus lain, pertukaran data tidak dilarang, namun relatif mahal dan memakan waktu. Dalam kasus terakhir, pertukaran sejumlah data di antara lokasi fusi yang berbeda diperbolehkan, asalkan masih dalam ambang batas yang diizinkan karena pertimbangan biaya atau waktu transfer. Dalam kasus di mana pertukaran data diperbolehkan, mungkin terdapat mekanisme yang sederhana namun efektif untuk mengatasi tantangan ketimpangan data. Jika tantangan muncul karena kelas yang ada tidak mencukupi, atau karena data yang termasuk dalam rentang ruang masukan yang berbeda (seperti dalam estimasi fungsi), seseorang mungkin dapat bertukar beberapa data dan mengizinkan masing-masing klien fusi untuk memiliki beberapa contoh dari data yang ada.

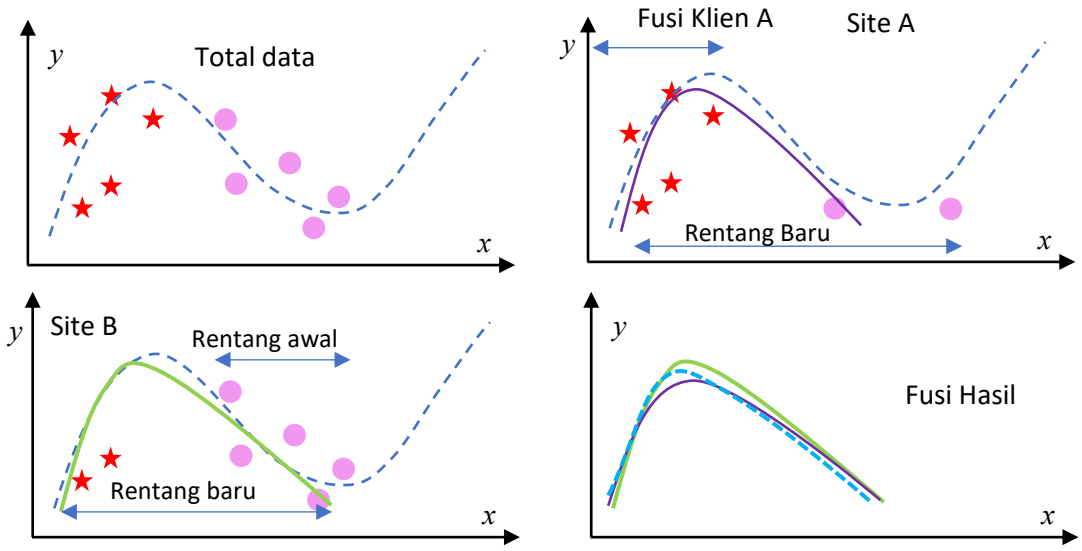
Untuk kasus estimasi fungsi, masing-masing klien fusi dapat menentukan rentang masukan untuk pengumpulan data mereka. Jika rentang masukan berbeda, masing-masing klien fusi dapat menukar sejumlah data terbatas sehingga semuanya memiliki rentang masukan yang sama. Pertukaran data terbatas ini perlu dilakukan sebagai langkah pra pemrosesan sebelum proses pembelajaran gabungan yang sebenarnya terjadi. Pertukaran data yang terbatas dapat menghasilkan peningkatan dalam tugas pembelajaran gabungan dengan meningkatkan rentang yang berlaku di mana fungsi tersebut dapat diterapkan.

Untuk melakukan pertukaran data terbatas, setiap klien fusi dapat mengirimkan rentang input yang telah dikumpulkan datanya ke server fusi. Salah satu cara untuk mengirimkan rentang ini adalah sebagai nilai atas dan bawah pada semua fitur masukan. Ukuran statistik lainnya pada fitur masukan, mis. persentil ke-5 dan persentil ke-95, juga dapat

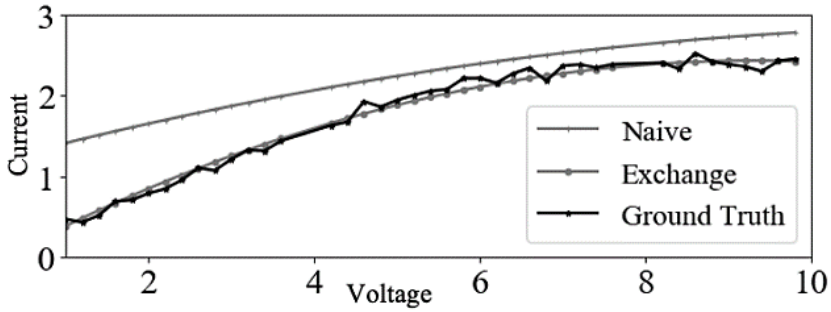
digunakan. Server fusi menggunakan rentang untuk menentukan apakah ada klien fusi yang kehilangan data dalam rentang ruang input tertentu. Jika semua klien fusi telah mengumpulkan data pada rentang nilai input yang sama atau sangat mirip, tidak diperlukan tindakan lebih lanjut. Namun, jika klien fusi mana pun memiliki kesenjangan dalam jangkauan pengumpulan datanya, server fusi dapat mengidentifikasi kesenjangan tersebut. Ia dapat meminta klien fusi lain untuk memberikan beberapa sampel data dalam kesenjangan tersebut kepada rekan mereka yang memiliki kesenjangan tersebut. Hal ini memastikan bahwa semua klien fusi memiliki rentang data yang dikumpulkan sama, menghitung estimasi fungsinya pada rentang yang sama, dan setiap klien fusi pada akhirnya menghitung perkiraan berbeda untuk fungsi yang sama.

Kita dapat mengilustrasikan penyesuaian rentang tersebut dengan mengeksplorasi apa yang dimungkinkan oleh pertukaran data dalam situasi hipotetis yang diilustrasikan pada Gambar 5.6. Ini adalah pengulangan contoh hipotetis yang ditunjukkan pada Gambar 5.2. Namun, sebagai bagian dari pertukaran data terbatas, beberapa poin dipertukarkan di antara klien fusi yang berbeda. Poin yang dipertukarkan ditampilkan dalam warna abu-abu di data masing-masing untuk klien fusi A dan B. Dengan pertukaran ini, rentang fungsi yang dipelajari berubah untuk kedua klien fusi. Rentang asli ditunjukkan dalam warna hitam dan rentang diperluas ditunjukkan dalam panah abu-abu pada gambar data untuk dua klien fusi. Karena rentangnya sekarang kira-kira sama untuk kedua klien fusi, mereka mempelajari perkiraan berbeda dari fungsi yang sama, dan proses rata-rata gabungan membawa mereka ke fungsi yang perkiraannya jauh lebih dekat dengan kebenaran dasar.

Kita juga dapat menguji dampak dari pertukaran data yang terbatas ini dalam contoh model komponen elektronik yang dibahas sebelumnya dalam bab ini (lihat Gambar 5.3 di Bagian 5.1.1). Dalam pengulangan percobaan dengan tiga klien fusi berbeda dengan tiga perspektif berbeda pada data komponen, setiap klien fusi bertukar informasi dan dengan demikian memperluas jangkauan fungsi yang datanya tersedia di setiap klien fusi. Hasil bersih dari pertukaran ini adalah estimasi fungsi yang jauh lebih baik, seperti yang ditunjukkan pada Gambar 5.7.



Gambar 5.6 Dampak pertukaran data terhadap estimasi fungsi.



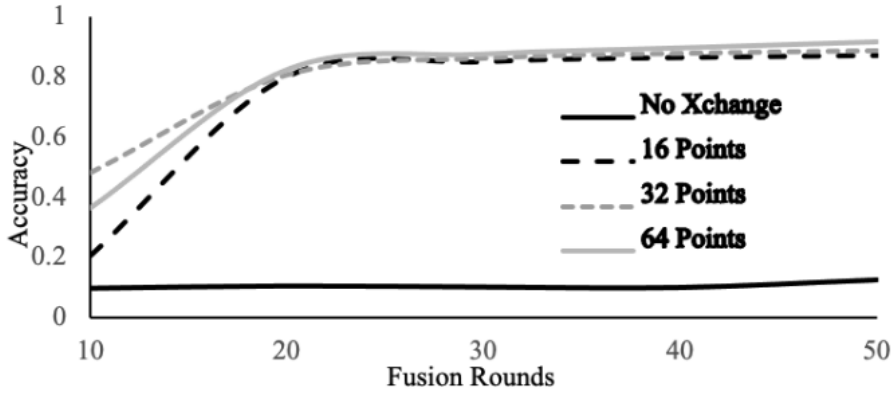
Gambar 5.7 Dampak pertukaran data pada model komponen elektronik.

Pertukaran data yang terbatas juga berfungsi baik dalam kasus data yang dipartisi untuk klasifikasi. Dalam kasus ini, situs fusi dapat mengirimkan statistik tentang jumlah titik data yang mereka miliki sesuai dengan setiap label/kelas ke server fusi. Server fusi dapat menentukan apakah ada klien fusi yang kehilangan beberapa kelas, dan meminta klien fusi lain yang memiliki data untuk kelas tersebut untuk menyediakan beberapa kelas. Hal ini memastikan bahwa setiap klien fusi merupakan model pelatihan yang memprediksi semua kelas sebagai keluaran, dan proses fusi model dapat diterapkan. Alternatifnya, setiap klien fusi dapat melaporkan beberapa titik data yang dimilikinya berdasarkan kelas ke server fusi, yang kemudian menggabungkan semua poin dan mengirimkannya ke setiap klien fusi. Setelah menerima data ini, setiap klien fusi dapat menambahkan data lokalnya untuk model pelatihan.

Dampak pertukaran data terhadap efektivitas model klasifikasi dengan data yang dipartisi ditunjukkan pada Gambar 5.8. Garis hitam pekat menunjukkan dampak tanpa pertukaran apapun, sama dengan hasil yang ditunjukkan pada Gambar 5.5. Performa model AI yang dilatih dengan data yang dipartisi kurang baik. Tiga baris lainnya menunjukkan hasil pertukaran beberapa instance dari setiap jenis kelas. Hanya ada sedikit titik data yang dipertukarkan dibandingkan dengan titik data yang ada. Setiap klien fusi memiliki sekitar 6.000 instance dari setiap kelas, namun jumlah data yang dipertukarkan adalah 16, 32, atau 64

instance per kelas. Bahkan dengan sedikit pertukaran kelas, metrik akurasi model meningkat secara signifikan dengan sangat sedikit putaran fusi rata-rata gabungan.

Hasil dari pertukaran data yang terbatas memberikan pendekatan untuk meningkatkan kualitas model pembelajaran gabungan dalam situasi ketika data dipartisi. Dalam situasi di mana data nyata tidak dapat dipertukarkan, memberikan beberapa sampel sintetis dari label yang hilang akan berdampak serupa pada kualitas model gabungan yang dibuat.

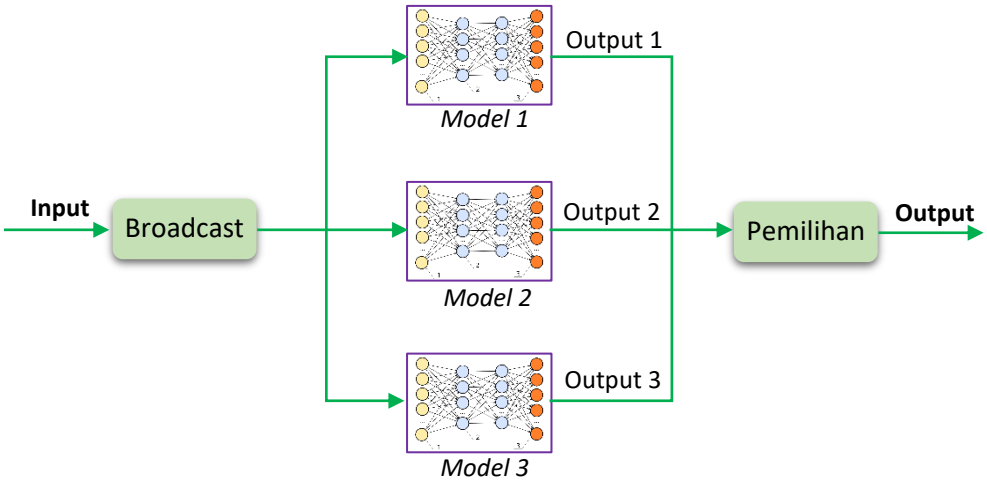


Gambar 5.8 Dampak pertukaran data terhadap klasifikasi.

5.3 ANSAMBEL BERBASIS KEBIJAKAN

Alternatif lain untuk menangani masalah partisi data adalah penggunaan ensembel. Ansambel adalah cara menggunakan beberapa model untuk memecahkan masalah yang sama. Asumsi dalam ansambel adalah bahwa semua model dilatih untuk melakukan tugas yang sama, yaitu menggunakan masukan yang sama persis dan menghasilkan keluaran yang sama persis. Selain itu, ansambel diasumsikan independen.

Dalam pendekatan ansambel, masukan yang sama dilewatkan melalui beberapa model, yaitu masukan disiarkan ke semua model, dan keluaran yang dihasilkan oleh masing-masing model diperiksa dan salah satu keluaran yang mungkin dipilih, misalnya dengan suara terbanyak untuk menentukan hasil akhir. Pendekatan ansambel dapat bekerja lebih baik daripada model individual mana pun, dan lebih mampu menangani variasi masukan, dengan asumsi bahwa masing-masing model dalam ansambel tidak bergantung pada model lainnya. Sebuah ansambel dengan tiga model ditunjukkan pada Gambar 5.9, beserta komponen untuk menyiarkan masukan dan menggabungkan keluaran model.

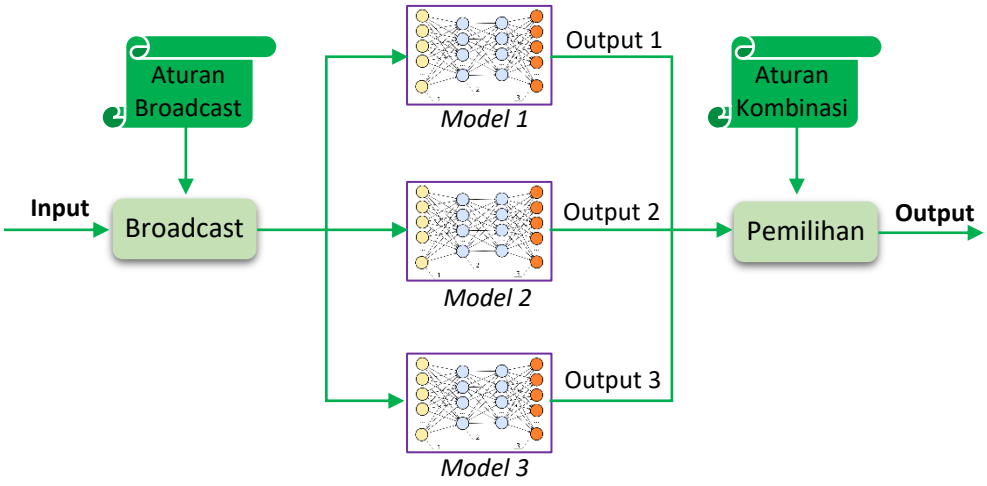


Gambar 5.9 Kumpulan model.

Ansambel memberikan cara alternatif untuk menggabungkan model dari klien fusi berbeda yang terlibat dalam sesi pembelajaran gabungan. Daripada mencoba menggabungkan model yang disediakan oleh masing-masing klien fusi yang berbeda, server fusi membuat ansambel model yang berbeda. Salah satu keuntungan dari ansambel adalah bahwa masing-masing model tidak perlu mengikuti arsitektur yang sama. Pendekatan ansambel dapat digunakan untuk menggabungkan keluaran dari berbagai jenis model, yang mungkin memiliki arsitektur berbeda dan struktur model internal berbeda.

Karena identitas masing-masing model dipertahankan secara terpisah, metadata mengenai pelatihan setiap model dapat dipertahankan untuk setiap model yang membentuk ansambel. Informasi ini kemudian dapat digunakan untuk secara bijaksana menggabungkan berbagai prediksi yang dibuat oleh masing-masing model yang membentuk ansambel. Salah satu pendekatan tersebut adalah penggunaan kebijakan.

Kebijakan adalah kumpulan aturan, masing-masing aturan ditentukan oleh pasangan kondisi-tindakan. Setiap aturan menentukan tindakan yang harus diambil ketika serangkaian kondisi yang ditentukan dalam aturan terpenuhi. Kebijakan dapat didefinisikan secara langsung sebagai seperangkat aturan, atau didefinisikan sebagai serangkaian konstruksi yang lebih tinggi yang kemudian disempurnakan menjadi serangkaian aturan. Dengan mendefinisikan kebijakan untuk mengambil tindakan yang berbeda, pengoperasian sistem apa pun dapat dimodifikasi agar berbeda dalam kondisi yang berbeda. Manajemen berbasis kebijakan telah digunakan untuk menyederhanakan pengelolaan sistem dan jaringan di banyak lingkungan berbeda, mulai dari pengelolaan jaringan hingga pengelolaan sistem dan aplikasi caching. Masuk akal jika kebijakan juga dapat digunakan untuk mengontrol operasional ansambel.



Gambar 5.10 Kumpulan berbasis kebijakan.

Ansambel berbasis kebijakan beroperasi seperti sebuah ansambel tetapi dengan dua komponen tambahan, yang keduanya dikendalikan oleh kebijakan, seperti yang ditunjukkan pada Gambar 5.10. Komponen pertama berkaitan dengan pemilihan model dalam ansambel. Di bawah kendali kebijakan, seseorang tidak perlu menggunakan semua ansambel dalam kebijakan, namun subset dapat dipilih berdasarkan serangkaian kondisi. Dalam kondisi tertentu, semua model dalam ansambel dapat digunakan. Dalam kondisi lain, subkumpulan model yang lebih kecil dapat digunakan dalam ansambel. Dalam kondisi lain, hanya satu model yang akan digunakan. Kondisi pemilihan subset ansambel yang akan digunakan ditentukan melalui kebijakan penyiaran yang ditunjukkan pada Gambar 5.10.

Komponen kedua yang dikendalikan oleh kebijakan berkaitan dengan kombinasi keluaran dari masing-masing model yang dipilih dalam ansambel. Daripada hanya menggunakan bobot mayoritas, atau rata-rata tertimbang, kombinasi output dapat dilakukan dengan lebih fleksibel dengan menggunakan serangkaian kebijakan kombinasi. Misalkan kombinasi dilakukan dengan menetapkan serangkaian bobot pada keluaran model yang berbeda. Salah satu kegunaan kebijakan kombinasi adalah mengubah bobot relatif model berdasarkan serangkaian kondisi.

Kebijakan kombinasi dan kebijakan penyiaran didefinisikan menggunakan serangkaian kondisi dan tindakan yang sesuai. Untuk kebijakan penyiaran, tindakan yang dilakukan hanyalah pemilihan model berbeda yang tersedia dalam ansambel. Dengan kata lain, jika ansambel terdiri dari N model, tindakan pemilihan dapat dilihat sebagai vektor yang berisi N entri Boolean, yang masing-masing menunjukkan apakah model terkait dipilih atau tidak. Kondisi untuk kebijakan penyiaran dapat ditentukan dengan menggunakan nilai masukan pada model, karakteristik model yang akan digunakan, atau informasi tentang data pelatihan yang digunakan untuk model tersebut.

Beberapa contoh kebijakan penyiaran untuk ansambel yang terdiri dari satu masukan x, satu keluaran y, dan menggunakan tiga model A, B, dan C adalah sebagai berikut:

- Jika x kurang dari 0,25, gunakan model A dan B saja
- Jika x antara 0,25 dan 0,5, gunakan model B dan C saja

- Jika x antara 0,25 dan 0,75, gunakan ketiga model A dan B saja
- Jika x lebih dari 0,75, gunakan keluaran model A saja

Secara umum, masukan akan terdiri dari fitur-fitur yang berbeda, dan kebijakannya akan lebih kompleks karena menggambarkan kombinasi nilai fitur yang berbeda-beda.

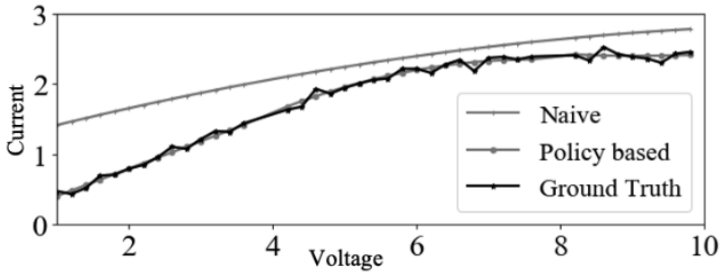
Setiap kombinasi kebijakan pasti mempunyai kondisi dan tindakan. Kondisinya bisa berupa ekspresi logis yang ditentukan atas nilai masukan ke ansambel, jenis model, keluaran model apa pun, atau atribut apa pun dari metadata tentang model tersebut. Tindakan tersebut akan menjadi bobot yang ditetapkan pada keluaran setiap model dalam ansambel. Untuk ansambel yang menerima satu masukan x, memprediksi satu keluaran y, dan terdiri dari tiga model A, B, dan C, kebijakan kombinasi mungkin terlihat seperti berikut:

- Jika ada model yang memperkirakan y kurang dari 0,25, gunakan hanya rata-rata tertimbang model yang memperkirakan y kurang dari 0,5
- Jika x antara 0,25 dan 0,5, gunakan bobot 0,25, 0,5, dan 0,25 untuk menggabungkan model A, B, dan C secara berturut-turut
- Jika x antara 0,25 dan 0,75, gunakan keluaran model A dan B saja
- Jika x kurang dari 0,25 atau lebih dari 0,75, gunakan bobot 0,5, 0,25, dan 0,25 untuk menggabungkan model A, B, dan C.

Pembahasan lebih rinci mengenai ansambel berbasis kebijakan dan penggunaannya dapat ditemukan di dan. Mereka telah terbukti merupakan peningkatan dari pendekatan lain dalam membuat model untuk berbagai model AI.

Mari kita periksa dampak ansambel berbasis kebijakan terhadap kinerja model estimasi fungsi komponen elektronik yang dijelaskan sebelumnya. Setiap model dicirikan oleh batas atas dan bawah masukan (Tegangan) data pelatihan yang digunakan untuk membuat model. Kebijakan penyiaran dalam hal ini dapat menyatakan bahwa hanya model yang dilatih berdasarkan data yang mencakup nilai masukan yang boleh digunakan dalam ansambel. Jika tidak ada model yang digunakan, sebaiknya digunakan dua model yang rentangnya paling dekat dengan nilai masukan dan keluarannya dirata-ratakan. Ini memberikan mekanisme alternatif untuk menggabungkan model keluaran dari tiga klien fusi.

Hasil penggabungan ketiga model dengan data ditunjukkan pada Gambar 5.2 dan Gambar 5.3 ditunjukkan pada Gambar 5.11. Ketika hasil dari ansambel berbasis kebijakan dibandingkan dengan pendekatan naif dalam menggabungkan output, terlihat bahwa ansambel berbasis kebijakan menghasilkan model yang lebih akurat. Hasilnya tidak terlalu mengejutkan, karena kebijakan memungkinkan preferensi diberikan kepada model-model yang lebih sesuai dengan kondisi yang berbeda-beda. Kebijakan untuk menggabungkan model dalam suatu ansambel tidak perlu didefinisikan secara manual. Mereka dapat dihasilkan secara otomatis dengan memeriksa properti data yang digunakan untuk melatih model.



Gambar 5.11 Dampak ansambel berbasis kebijakan terhadap estimasi fungsi.

Ansambel berbasis kebijakan juga memberikan solusi yang baik untuk beberapa masalah sinkronisasi data yang dibahas di Bab 7.

5.4 RINGKASAN

Dalam bab ini, kita melihat tantangan yang muncul ketika data tidak seimbang. Ketika data tidak seimbang, klien fusi yang berbeda mungkin mempelajari fungsi berbeda yang tidak boleh dirata-ratakan secara naif. Kemiringan data dapat diakibatkan oleh situasi di mana klien fusi yang berbeda mengamati informasi berdasarkan asumsi atau lingkungan operasi yang berbeda. Hal ini juga dapat terjadi ketika beberapa kelas keluaran hilang dari beberapa klien fusi.

Salah satu pendekatan yang bekerja dengan baik untuk mengatasi kesenjangan data adalah pertukaran data yang terbatas jika diizinkan dalam konteks bisnis. Pertukaran data memperluas cakupan fungsi yang dipelajari sehingga semua klien fusi mempelajari fungsi yang sama. Pendekatan lain untuk mengatasi data skew adalah dengan menggabungkan model menggunakan konsep ansambel. Kombinasi tersebut dapat digabungkan dengan kebijakan untuk menciptakan model keseluruhan yang dapat bekerja lebih baik dalam kondisi yang berbeda-beda. Kebijakan untuk ansambel dapat dibuat sehingga mereka memilih subkumpulan model yang lebih terlatih untuk lingkungan tempat tugas inferensi dilakukan.

BAB 6
MENGATASI MASALAH KEPERCAYAAN DALAM PEMBELAJARAN FEDERASI

Meskipun sebagian besar skenario untuk pembelajaran gabungan di lingkungan perusahaan dibuat sedemikian rupa sehingga klien fusi dan server fusi dapat saling percaya, ada beberapa skenario di mana kepercayaan di antara berbagai pihak tidak bersifat mutlak. Dalam kasus ini, sistem pembelajaran gabungan perlu mempertimbangkan keterbatasan yang mungkin timbul karena terbatasnya kepercayaan di antara para pihak. Dalam bab ini, kita akan melihat beberapa permasalahan kepercayaan dan pendekatan untuk mengatasinya.

Ketika kepercayaan terbatas, situs mungkin tidak mau membagikan parameter model secara mentah dengan mitra lain, dan mekanisme akan diperlukan agar model dapat dibangun tanpa harus membagikan parameter model secara jelas kepada pihak yang tidak dipercaya. Kebutuhan untuk tidak membagikan parameter model secara jelas dapat menjadi penghalang dalam beberapa konteks.

Salah satu cara untuk memahami situasi ini dan mengatasinya adalah dengan menggunakan konsep zona kepercayaan. Zona kepercayaan terdiri dari sistem yang saling percaya dan bersedia berbagi data dan informasi satu sama lain tanpa batasan. Contoh zona kepercayaan adalah server dalam firewall perusahaan. Karena dilindungi oleh firewall, sistem di dalam perusahaan lebih percaya satu sama lain dibandingkan sistem yang berada di luar firewall. Di perusahaan besar, di pusat data yang menampung banyak aplikasi, atau di lokasi yang dihosting di cloud, mungkin terdapat beberapa zona kepercayaan yang dilindungi oleh serangkaian beberapa firewall atau perangkat keamanan lainnya.

Biasanya, dalam sisa diskusi tentang pembelajaran gabungan dalam buku ini, kami berasumsi bahwa semua sistem termasuk dalam zona kepercayaan yang sama, yaitu klien fusi yang berbeda dan server fusi semuanya termasuk dalam zona kepercayaan yang sama. Dalam bab ini, kami mengkaji situasi yang mencakup lebih dari satu zona kepercayaan dan mendiskusikan pendekatan untuk memungkinkan pembelajaran gabungan di seluruh zona kepercayaan.

6.1 SKENARIO DENGAN BEBERAPA ZONA KEPERCAYAAN

Beberapa skenario di mana pembelajaran gabungan perlu diterapkan di berbagai zona kepercayaan mencakup penggunaan layanan berbasis cloud, konsorsium, aliansi, dan koalisi militer.

6.1.1 Server Fusion berbasis Cloud

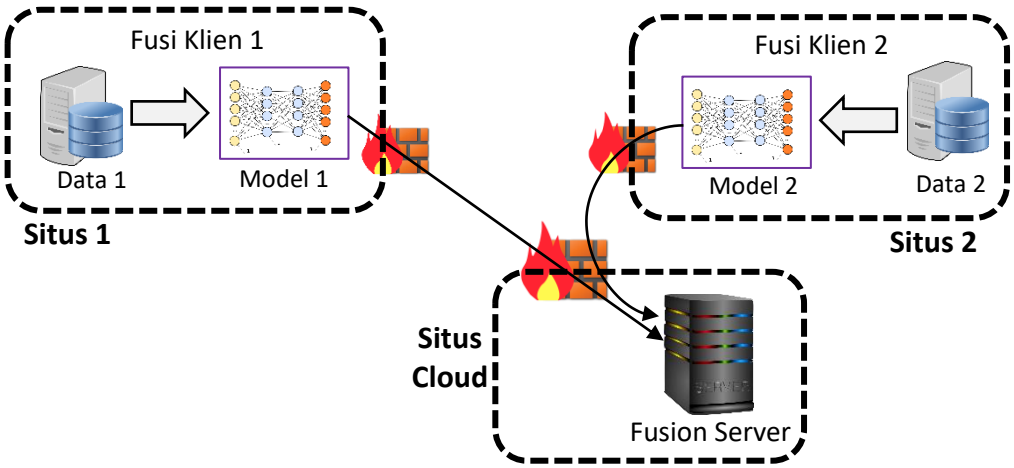
Komputasi awan telah menjadi salah satu mode dominan dalam menyediakan layanan dan infrastruktur komputasi. Dari pembahasan Bab 4 dan Bab 5, terlihat jelas bahwa klien fusi dan server fusi perlu mengimplementasikan serangkaian fungsi yang kompleks agar seluruh sistem pembelajaran gabungan dapat bekerja sama. Salah satu cara untuk menyederhanakan

kompleksitas adalah dengan menjalankan server fusi sebagai layanan yang tersedia untuk digunakan sebagai layanan yang dihosting di cloud.

Namun, ketika server fusi dijalankan dan dioperasikan sebagai layanan yang dihosting di cloud, dan klien data serta fusi ada di berbagai cabang perusahaan, kepercayaan antara server fusi dan klien fusi mungkin terbatas. Pertimbangkan sebuah bank yang mungkin memiliki gudang data lokal di beberapa lokasi berbeda, dan ingin membangun model AI umum di seluruh data di semua gudang. Penyedia cloud mungkin menawarkan server fusi sebagai layanan cloud kepada bank. Bank mungkin tertarik untuk memanfaatkan server fusi di cloud, karena hal ini akan menghemat waktu dalam mendapatkan model AI yang baik untuk digunakan dalam proses bisnis. Namun, bank mungkin khawatir tentang transmisi data atau bahkan model ke layanan cloud, dan paparan keamanan atau kebocoran informasi dari model tersebut.

Situasinya ditunjukkan pada Gambar 6.1. Menggunakan server fusi di cloud akan memberikan banyak utilitas berguna bagi perusahaan, yang dapat mengatasi masalah perbedaan format data dan distorsi data yang dibahas di Bab 4 dan Bab 5. Layanan cloud akan mencakup kemampuan untuk menentukan kebijakan kualitas data, kebijakan untuk mengubah data ke dalam format umum, identifikasi label yang hilang dan berbagai fungsi lainnya. Meskipun bank dapat mereplikasi layanan-layanan ini dalam lingkungan komputasinya sendiri, bank mungkin akan merasa lebih nyaman, bijaksana dan hemat biaya jika hanya memanfaatkan fungsi-fungsi ini sebagai layanan paket yang disediakan oleh penyedia layanan cloud.

Ada yang berpendapat bahwa server fusi di cloud tidak pernah bisa melihat data mentah bank, dan hanya memiliki visibilitas ke parameter model. Tingkat perlindungan tersebut mungkin dapat diterima oleh beberapa bank. Namun, beberapa bank mungkin merasa tidak nyaman bahkan mengirimkan model ke penyedia layanan cloud.



Gambar 6.1 Skenario federasi berbasis cloud.

Ada beberapa makalah yang menunjukkan bagaimana informasi dapat bocor dari model yang terlatih. Seseorang dapat melakukan serangan inversi model pada algoritma pembelajaran mesin. Serangan inversi model memungkinkan seseorang menentukan

masukan yang digunakan untuk melatih model. Salah satu pendekatan untuk melakukan serangan inversi model adalah dengan memeriksa skor kepercayaan sampel uji terpilih yang melewati model terlatih. Ini telah digunakan untuk merekonstruksi wajah yang disediakan untuk pelatihan model pengenalan wajah.

Selain itu, jaringan saraf dapat dilatih untuk membedakan antara sampel yang termasuk dalam kumpulan data dan sampel yang tidak menggunakan teknik seperti Generative Adverserial Networks atau GANs. Jika digunakan secara jahat, GAN dapat digunakan untuk membuat ulang bagian dari kumpulan data pelatihan. Ini berarti penyedia layanan cloud akan dapat membuat ulang beberapa kumpulan data pelatihan, yang dapat menyebabkan kebocoran informasi yang tidak diinginkan.

Perlu dicatat bahwa hampir semua serangan dan kebocoran informasi yang dipublikasikan dalam literatur ilmiah hanya dapat terjadi dalam kondisi yang dirancang dengan cermat. Proses bisnis normal dan praktik operasi penyedia layanan cloud biasanya menawarkan perlindungan yang kuat terhadap kebocoran data yang disengaja. Oleh karena itu, kekhawatiran mengenai kebocoran informasi mungkin hanya bersifat akademis dan bukan kekhawatiran nyata dalam praktik. Namun, bank mungkin khawatir tentang kemungkinan kompromi dalam keamanan penyedia layanan cloud. Meskipun kemungkinannya kecil, bank mungkin masih ragu untuk mengirimkan model tersebut ke penyedia layanan cloud secara jelas dan khawatir akan kebocoran data mereka.

Dalam skenario khusus ini, klien fusi yang berbeda milik bank berada dalam satu zona kepercayaan, sedangkan server fusi di cloud berada dalam zona kepercayaan yang berbeda. Untuk menangani situasi ini, kita memerlukan solusi untuk pembelajaran gabungan di mana modelnya tidak jelas-jelas melintasi zona kepercayaan.

6.1.2 Situs Cloud Multi-penyewa

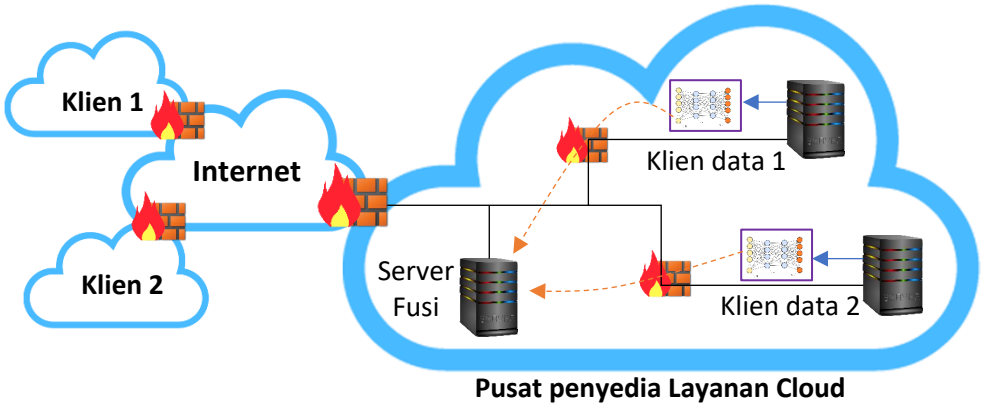
Salah satu lingkungan di mana kami menunjukkan bahwa pembelajaran gabungan akan sangat berguna adalah dalam konteks berbagi informasi tentang penyewa berbeda yang mungkin dihosting oleh penyedia layanan cloud yang sama, seperti yang dijelaskan dalam Bagian 2.4.3 di Bab 2. Khususnya, jika Jika penyedia layanan cloud mengelola aplikasi untuk banyak penyewa yang berbeda, mereka akan dapat melatih model AI berdasarkan kesalahan yang mungkin terlihat dalam log yang dihasilkan oleh aplikasi yang dihosting dari penyewa yang berbeda. Log ini dapat membantu penyedia yang dihosting di cloud mendapatkan peringatan dini tentang potensi kegagalan aplikasi yang dihosting, dan akan dapat memanfaatkan serta menggabungkan informasi di beberapa penyewa untuk meningkatkan kemampuannya dalam mengambil tindakan secara proaktif terhadap kemungkinan kesalahan. Situasinya ditunjukkan pada Gambar 6.2.

Seperti yang ditunjukkan pada Gambar, penyedia layanan cloud berhati-hati untuk tidak memindahkan data mentah apa pun atau keluar dari firewall yang melindungi server milik masing-masing klien, klien 1 atau klien 2. Hanya model yang dipindahkan dari lingkungan klien , dilindungi oleh firewall klien, dan server fusi, yang terletak di situs manajemen penyedia layanan cloud yang digunakan untuk menggabungkan model. Penyedia layanan dapat

meyakinkan setiap klien bahwa log aplikasi mereka, atau informasi apa pun tentang pengelolaan server mereka, hanya disimpan dalam kantong cloud mereka.

Meskipun pembuatan model yang dapat membantu pengoperasian layanan yang dihosting dan meningkatkan pengoperasian penyedia layanan cloud sangat menjanjikan, hal ini juga menimbulkan potensi kekhawatiran di antara klien yang dilayani oleh penyedia layanan cloud. Klien yang dihosting oleh penyedia layanan cloud mungkin tidak menyukai kenyataan bahwa model yang dilatih berdasarkan data mereka dicampur dengan model yang dilatih berdasarkan data dari pesaing mereka. Kekhawatiran mereka mungkin bukan pada kebocoran data apa pun ke penyedia layanan cloud. Bagaimanapun, mereka memercayai penyedia layanan untuk menjalankan server dan aplikasi mereka. Namun, mereka mungkin khawatir tentang penggunaan data dari klien lain yang dihosting di lingkungan cloud yang sama. Mereka mungkin tidak ingin situs lain memperoleh informasi apa pun tentang data mereka yang digunakan untuk membuat model bersama.

Hubungan kepercayaan di antara berbagai pihak dalam sistem federasi sangat berbeda dalam kasus-kasus ini. Klien fusion mempercayai server fusion. Namun, klien fusi yang berbeda tidak saling percaya.



Gambar 6.2 Skenario federasi multi-tenancy.

6.1.3 Konsorsium dan Aliansi

Di banyak industri, konsorsium dibentuk untuk memungkinkan pertukaran pengetahuan dan informasi antar organisasi yang berbeda. Sebagai contoh, tidak jarang organisasi layanan kesehatan di berbagai negara membentuk aliansi untuk berbagi informasi tentang data layanan kesehatan satu sama lain. Beberapa organisasi gabungan telah dibentuk untuk berbagi informasi tentang pertanian, isu-isu sosial, dan demografi di antara organisasi-organisasi anggota.

Aliansi juga dapat dibentuk di antara organisasi akademis, industri, dan pemerintah yang berbeda untuk melakukan penelitian kolaboratif. Beberapa contohnya termasuk Aliansi Teknologi Internasional dalam Ilmu Jaringan, Aliansi Teknologi Internasional dalam Analisis Terdistribusi dan Ilmu Informasi, aliansi penelitian kolaboratif robotika, dll. Semua aliansi ini menyatukan para peneliti dari organisasi yang berbeda, dan berbagi kegiatan penelitian satu sama lain. Anggota program penelitian berkolaborasi dengan berbagi data, pengetahuan dan

pengalaman untuk memecahkan suatu masalah bersama. Meskipun organisasi penelitian bekerja sama, mereka mungkin tidak selalu memiliki kebebasan penuh untuk berbagi data atau model satu sama lain. Anggota industri akan mempunyai pembatasan tambahan atas partisipasinya karena kepentingan komersial perusahaan. Anggota pemerintah akan dikenakan pembatasan tambahan pada operasional mereka karena persyaratan peraturan yang harus mereka penuhi. Hal ini menciptakan situasi dengan terbatasnya kepercayaan di antara berbagai pihak.

Konsorsium dibentuk untuk berbagi informasi antar organisasi yang berbeda. Namun, beberapa anggota mungkin merasa bahwa sebagian data mereka mungkin berisi rincian sensitif yang tidak boleh dibagikan kepada orang lain. Berbagi model di antara anggota aliansi mungkin diizinkan, atau bahkan diaktifkan oleh layanan cloud milik aliansi, namun anggota aliansi mungkin tidak ingin mitra lain melihat parameter model yang mereka kontribusikan. Kekhawatiran mengenai pembagian parameter mungkin disebabkan oleh risiko kebocoran informasi dari model. Jika server fusi berada di bawah kendali anggota konsorsium, anggota konsorsium lain mungkin memiliki kepercayaan terbatas terhadap server fusi. Mereka mungkin tidak ingin server fusi melihat parameter modelnya dengan jelas.

Zona kepercayaan dalam konsorsium dapat terwujud dalam dua cara berbeda. Cara pertama adalah pada Bagian 6.1.2 di mana klien fusi mungkin merasa nyaman berbagi parameter model mentah dengan server fusi yang dihosting oleh konsorsium, namun tidak dengan klien fusi milik anggota lain. Cara lainnya adalah ketika sekelompok anggota konsorsium mungkin saling percaya, namun tidak sepenuhnya mempercayai anggota konsorsium ketiga atau sekelompok anggota konsorsium lainnya.

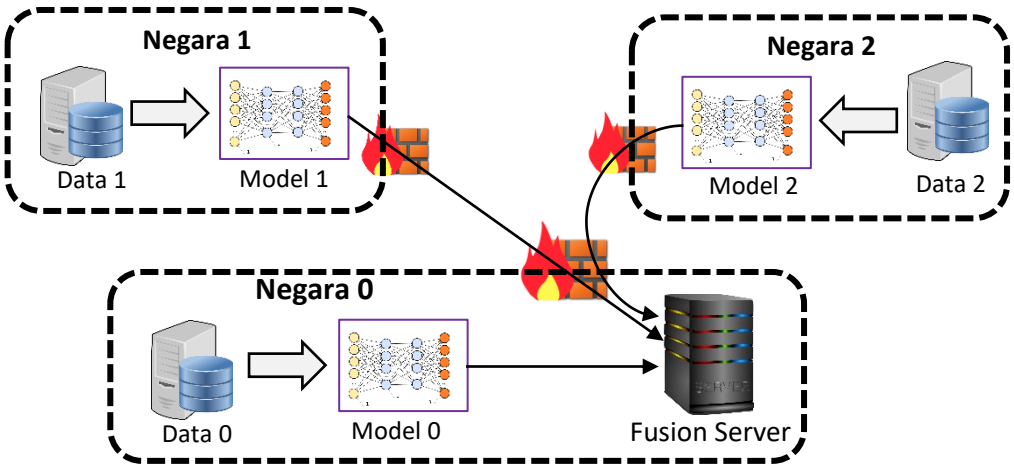
6.1.4 Koalisi Militer

Operasi militer modern sering kali dilakukan dalam koalisi, di mana lebih dari satu negara bergabung untuk menjalankan misi secara kolaboratif. Operasi koalisi semacam ini merupakan hal yang umum dalam upaya pemeliharaan perdamaian, di mana banyak negara bekerja sama untuk menjaga perdamaian di wilayah yang rawan konflik. Mereka juga digunakan untuk operasi kemanusiaan jika terjadi bencana alam.

Operasi koalisi mempunyai tantangan teknisnya sendiri di berbagai bidang yang perlu dieksplorasi, termasuk tantangan dalam pembelajaran gabungan. Saat koalisi bekerja sama, hampir setiap negara anggota mengumpulkan data selama menjalankan operasinya. Data tersebut dapat berupa rekaman video dari pengintaian, rekaman audio dan pembacaan seismik dari sensor, atau data tabel yang disimpan secara manual oleh personel yang terlibat dalam upaya bersama. Sebagian besar data ini dapat dibagikan kepada anggota koalisi untuk meningkatkan operasi mereka. Sebagai contoh, rekaman video mengenai pemberontak yang dikumpulkan oleh negara-negara anggota dapat dibagikan untuk meningkatkan model AI yang digunakan untuk mendeteksi pemberontakan dan memperingatkan pasukan penjaga perdamaian.

Anggota koalisi mungkin tidak dapat berbagi data mentah, namun mungkin dapat berbagi parameter model satu sama lain. Salah satu kasus spesifik yang mungkin muncul ditunjukkan pada Gambar 6.3, di mana salah satu anggota koalisi sedang melatih model AI

namun menggunakan informasi yang tersedia dari semua mitra koalisi. Masing-masing mitra koalisi akan bersedia untuk berbagi model, namun hal ini tidak akan terjadi jika parameter model sudah jelas. Zona kepercayaan untuk server federasi mencakup agen federasi yang berasal dari negara yang sama, namun masing-masing agen lainnya berada di zona kepercayaannya masing-masing.



Gambar 6.3 Skenario federasi koalisi.

Situasi yang mirip dengan operasi koalisi juga dapat muncul dalam situasi darurat lainnya di mana berbagai lembaga pemerintah bekerja sama dalam operasi gabungan, misalnya dalam operasi gabungan. bereaksi terhadap bencana alam, atau merencanakan acara khusus. Berbagai jenis peraturan mungkin menghalangi pembagian data mentah atau parameter model yang tidak dilindungi secara menyeluruh di antara lembaga-lembaga tersebut, namun mereka akan dapat melatih model dan berbagi model satu sama lain, terutama jika parameter model dapat dienkripsi selama proses fusi.

6.2 KONFIGURASI ZONA KEPERCAYAAN

Dari pembahasan berbagai skenario di Bagian 6.1, kita dapat mengidentifikasi empat kemungkinan konfigurasi hubungan kepercayaan yang terjadi dalam pengaturan pembelajaran gabungan yang berbeda. Keempat kemungkinan konfigurasi ini ditunjukkan pada Tabel 6.1 dan menggambarkan hubungan dari perspektif klien fusi tunggal dengan server fusi dan klien fusi lainnya.

Tabel 6.1: Kemungkinan konfigurasi kepercayaan.

Nomor	Server Fusi	Klien Fusion lainnya
1	Tepercaya	Tepercaya
2	Tepercaya	Tidak terpercaya
3	Tidak terpercaya	Tepercaya
4	Tidak terpercaya	Tidak terpercaya

Kasus dimana setiap orang dipercaya hanya membutuhkan satu zona kepercayaan. Tiga konfigurasi lainnya perlu memiliki beberapa zona kepercayaan untuk pembelajaran gabungan. Kita perlu memiliki pendekatan untuk melakukan pembelajaran gabungan untuk masing-masing konfigurasi ini untuk memastikan bahwa data atau model tidak jelas melintasi batas-batas zona kepercayaan. Ketiga konfigurasi tersebut dapat diberi nama sebagai (i) Situs fusi tidak tepercaya dengan Klien Fusion tepercaya (ii) Situs fusi tepercaya dengan klien fusi tidak tepercaya dan (iii) situs fusi tidak tepercaya dengan klien fusi tidak tepercaya.

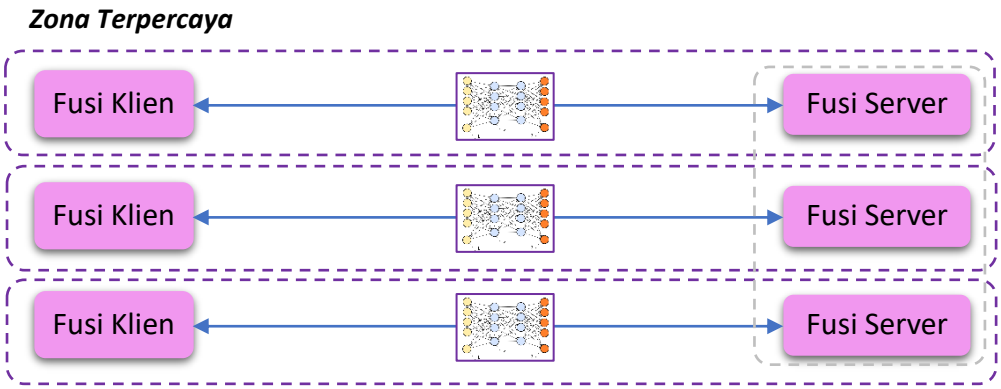
Perhatikan bahwa kata sifat tidak tepercaya dalam situasi zona kepercayaan tidak berarti bahwa klien fusi atau server fusi mana pun bertindak jahat. Sistem yang tidak tepercaya hanya termasuk dalam zona kepercayaan lain di mana data atau model tidak dapat dikirim dengan jelas. Kekhawatirannya bukan pada mitra jahat, melainkan kebocoran model atau data bersama yang tidak disengaja, yang memerlukan penambahan lapisan keamanan tambahan.

6.2.1 Server Fusion Tepercaya dengan Klien Fusion Tidak Tepercaya

Dalam konfigurasi ini, yang ditunjukkan pada Gambar 6.4, setiap klien fusi mempercayai server fusi dan bersedia berbagi model dan statistik lainnya dengan server. Namun, klien fusi tidak percaya satu sama lain.

Untuk konfigurasi ini, terdapat banyak zona kepercayaan yang berbeda, masing-masing zona kepercayaan termasuk klien fusi dan server fusi. Server fusi ditugaskan dengan tantangan agar klien rekan tidak dapat memperoleh informasi tentang model yang disediakan oleh klien fusi individu lainnya. Perhatikan bahwa, dalam situasi khusus ini, server fusi adalah milik semua zona kepercayaan, dan memiliki tanggung jawab untuk memastikan bahwa data atau model tidak melintasi zona kepercayaan.

Konfigurasi ini akan muncul dalam skenario yang dijelaskan dalam Bagian 6.1.2. Perhatikan bahwa hal ini juga dapat terjadi dalam beberapa skenario konsorsium (Bagian 6.1.3). Sebuah konsorsium dapat dibentuk untuk memberikan layanan khusus kepada masing-masing anggota. Meskipun masing-masing anggota bersedia memercayai konsorsium (yang didirikan sebagai entitas independen), mereka belum tentu memiliki tingkat kepercayaan yang sama dengan anggota konsorsium lainnya. Contoh dari konsorsium tersebut adalah *Federal National Mortgage Association* (FNMA) yang juga dikenal sebagai *Fannie Mae*. Ini memberikan layanan umum kepada bank dan perusahaan keuangan yang menjalankan bisnis menawarkan pinjaman hipotek di Amerika Serikat. Fannie Mae didukung oleh pemerintah AS dan beroperasi sebagai perusahaan independen. Jika Fannie Mae menjadi tuan rumah server fusi untuk menyediakan layanan pembuatan model AI guna mendukung sekelompok bank kecil guna meningkatkan proses hipotek mereka, setiap bank akan memiliki tingkat kepercayaan dalam menyediakan modelnya kepada Fannie Mae. Namun, bank-bank tersebut mungkin menginginkan jaminan dari Fannie Mae bahwa model mereka tidak akan diperlihatkan kepada bank-bank pesaing.

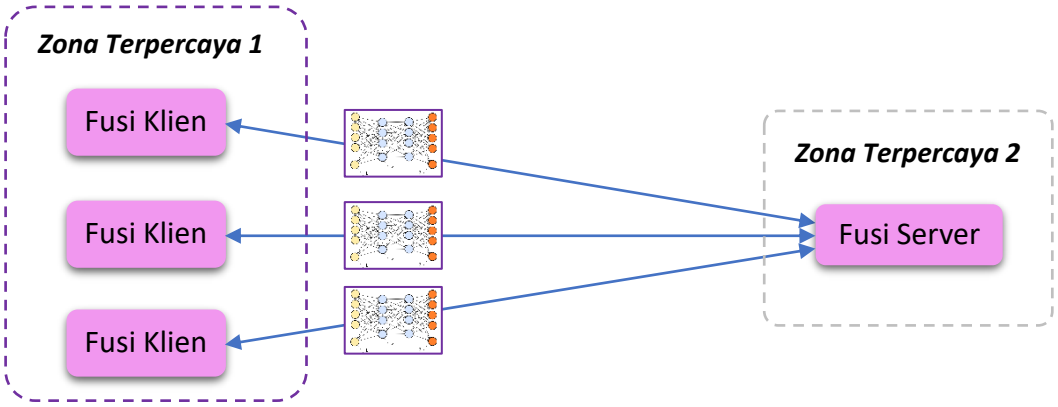


Gambar 6.4 Server fusi tepercaya dengan klien fusi yang tidak tepercaya.

6.2.2 Server Fusion Tidak Tepercaya dengan Klien Fusion Terpercaya

Dalam konfigurasi yang melibatkan beberapa zona kepercayaan ini, klien fusi saling percaya, tetapi tidak memiliki kepercayaan penuh pada server fusi. Konfigurasi kepercayaan ini tipikal untuk skenario yang dihosting di cloud yang dijelaskan di Bagian 6.1.1. Situasi serupa juga dapat muncul dalam operasi koalisi jika beberapa sub-unit dari satu negara memanfaatkan infrastruktur militer negara lain untuk meningkatkan model AI mereka. Dalam koalisi militer yang melibatkan satu negara yang memiliki infrastruktur yang signifikan untuk melakukan operasi di beberapa wilayah geografis yang membantu negara yang mungkin tidak memiliki teknologi maju, situasi ini mungkin muncul. Contohnya adalah koalisi yang dipimpin oleh Amerika Serikat untuk membantu negara fiksi Gao dalam skenario koalisi hipotetis Binni [85] atau Amerika Serikat membantu negara fiksi Holistan dalam skenario Holistan [86]. Dalam skenario ini, Amerika Serikat akan menjadi pangkalan operasi untuk membantu operasi penjaga perdamaian di wilayah negara tuan rumah (Gao atau Holistan). Pasukan AS akan memiliki data di banyak lokasi, yaitu mereka akan memiliki klien fusi di basis yang berbeda-beda dan mungkin memanfaatkan server fusi yang dihosting oleh negara tuan rumah untuk membuat model gabungan.

Server fusi tidak tepercaya dengan konfigurasi klien fusi tepercaya ditunjukkan pada Gambar 6.5. Klien fusi yang berbeda termasuk dalam zona kepercayaan yang sama yang ditandai sebagai zona kepercayaan 2. Namun, server fusi termasuk dalam zona kepercayaan yang berbeda, yang ditandai sebagai zona kepercayaan 1. Di antara dua zona kepercayaan, model dan ringkasan data adalah sedang dipertukarkan. Untuk melakukan tugas seperti fusi model dan negosiasi kebijakan lainnya, server fusi perlu beroperasi sedemikian rupa tanpa melihat parameter model secara jelas.



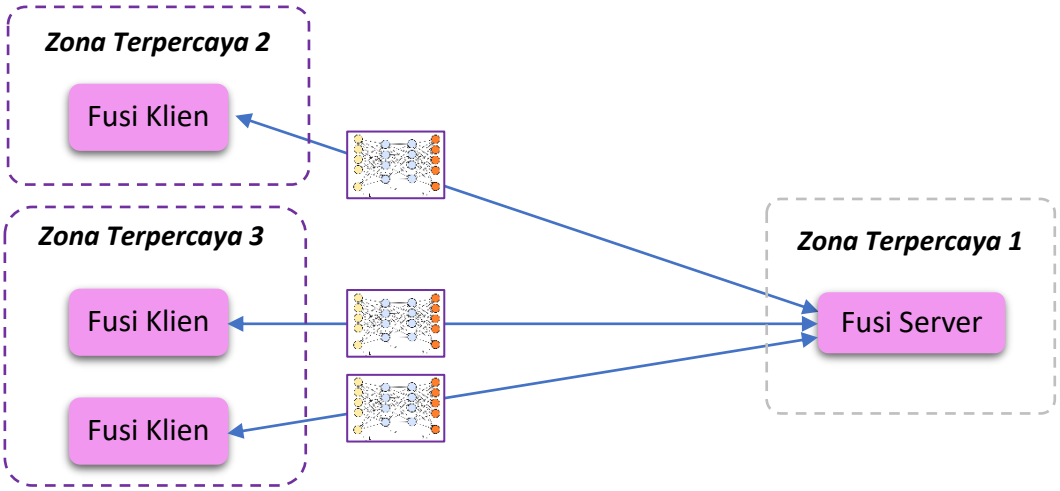
Gambar 6.5 Server fusi tidak tepercaya dengan klien fusi tepercaya.

6.2.3 Server Fusion Tidak Tepercaya dengan Klien Fusion Tidak Tepercaya

Konfigurasi ini akan muncul di lingkungan di mana klien fusi tidak mempercayai server fusi, juga tidak mempercayai klien fusi rekannya. Konfigurasi ini ditunjukkan pada Gambar 6.6. Setiap klien fusi dan server fusi berada di zona kepercayaan yang berbeda. Perhatikan bahwa beberapa klien fusi mungkin saling percaya, namun kami kemudian dapat mengelompokkannya bersama-sama karena berada dalam zona kepercayaan yang sama. Pada gambar, zona kepercayaan server ditandai sebagai nomor 1 dan zona kepercayaan klien diberi nomor 2 dan seterusnya. Klien Fusion ingin melindungi model mereka dari klien lain, serta server fusion.

Konfigurasi ini dapat terjadi dalam skenario konsorsium, aliansi, dan koalisi militer. Dalam koalisi militer antara beberapa negara yang semuanya memiliki akses terhadap teknologi canggih, salah satu sekutu mungkin menawarkan untuk menjalankan server fusi. Sekutu lain dalam koalisi akan menjalankan klien fusi mereka, namun mereka akan memiliki kepercayaan yang terbatas terhadap klien fusi lain, atau dengan server fusi lainnya.

Dalam konsorsium yang dibentuk antara berbagai perusahaan untuk berbagi informasi layanan kesehatan atau demografi, skenario kepercayaan terbatas yang serupa mungkin muncul. Setiap anggota akan mengoperasikan klien fusi mereka, dan memiliki kepercayaan terbatas pada server fusi. Demikian pula, mereka tidak mau mempercayai klien fusi lain yang dioperasikan oleh perusahaan lain.



Gambar 6.6 Server fusi tidak tepercaya dengan klien fusi yang tidak tepercaya.

Berbagai skenario yang dijelaskan di Bagian 6.1 dapat dipetakan ke konfigurasi berbeda dengan beberapa zona kepercayaan, seperti yang ditunjukkan pada Tabel 6.2.

Tabel 6.2 Pemetaan skenario untuk mempercayai konfigurasi.

Skenario	Konfigurasi Zona Kepercayaan
Server Fusion berbasis cloud	Server Fusion Tidak Tepercaya dengan Klien Fusion Tepercaya
Situs Cloud multi-penyewa	Server Fusion Tepercaya dengan Klien Fusion Tidak Tepercaya
Konsorsium dan Aliansi	Server Fusion Tidak Tepercaya dengan Klien Fusion Tidak Tepercaya
Koalisi Militer	Server Fusion Tidak Tepercaya dengan Klien Fusion Tidak Tepercaya

Pada beberapa bagian berikutnya, kita akan membahas pendekatan untuk melakukan pembelajaran gabungan di bawah konfigurasi zona kepercayaan yang berbeda-beda.

6.3 MENGATASI MASALAH KEPERCAYAAN DENGAN PERJANJIAN BISNIS

Salah satu solusi untuk mengatasi masalah zona kepercayaan adalah dengan menghilangkan zona kepercayaan dan membangun zona kepercayaan tunggal di mana semua klien dan server fusi dapat beroperasi. Pengaturan bisnis dapat digunakan untuk membangun zona kepercayaan tersebut. Pendekatan ini secara efektif menghindari masalah melintasi batas-batas zona kepercayaan dan bukan menyelesaikannya.

Meskipun pendekatan ini bukan solusi teknis, pendekatan ini terbukti efektif dalam banyak skenario bisnis yang melibatkan kepercayaan terbatas. Batasan kepercayaan muncul karena fakta bahwa entitas yang berbeda dalam proses pembelajaran gabungan, yaitu klien fusi dan server fusi, berasal dari organisasi yang berbeda. Kekhawatiran utama organisasi dalam berbagi data atau model dengan organisasi atau organisasi lain adalah kemungkinan kebocoran informasi. Jika ada data sensitif yang bocor, mungkin terdapat implikasi finansial, yang mungkin timbul karena pelanggaran persyaratan peraturan, biaya yang dikeluarkan

untuk tindakan mitigasi jika terjadi kebocoran data, atau karena hilangnya bisnis karena publisitas buruk yang timbul karena kebocoran data. kebocoran data.

Di masing-masing dari tiga konfigurasi zona kepercayaan yang diidentifikasi di Bagian 6.2, interaksi utama terjadi antara klien fusi dan server fusi. Meskipun dimungkinkan untuk melakukan tugas apa pun yang dilakukan dengan cara klien-server dengan pendekatan peer-to-peer tanpa melibatkan klien mana pun, kompleksitas pelaksanaannya jauh lebih sulit dibandingkan dengan arsitektur klien-server. Oleh karena itu, kami berasumsi bahwa interaksi dan pengaturan bisnis terjadi secara murni klien-server.

Pengaturan bisnis dalam kasus ini biasanya terdiri dari kontrak antara klien dan server di mana masing-masing organisasi akan menentukan cara penanganan data mereka, dan sanksi keuangan apa pun yang mungkin dikenakan oleh salah satu pihak jika terjadi kesepakatan. cara tidak diikuti dengan benar. Kedua organisasi akan sepakat untuk berbagi informasi yang memadai untuk saling memeriksa dan memastikan bahwa perjanjian telah dipatuhi dengan baik. Setelah pengaturan tersebut dibuat, model dan data dapat dipertukarkan secara jelas di seluruh zona kepercayaan.

Pengaturan seperti ini sudah menjadi praktik umum di banyak bisnis, dan meskipun bukan merupakan solusi teknis, seringkali merupakan pendekatan yang paling tepat untuk bekerja di berbagai zona kepercayaan yang berbeda. Istilah umum yang digunakan untuk pengaturan tersebut adalah perjanjian tingkat layanan, yang dapat didefinisikan untuk berbagai aspek keamanan data dan konten digital lainnya (misalnya model) yang akan dibagikan di antara para mitra. Perjanjian tingkat layanan telah ditentukan untuk berbagai jenis pengaturan, misalnya. keamanan, layanan yang dihosting di cloud dan jaringan komunikasi komputer. Jenis perjanjian yang sama dapat disediakan dan diatur dalam pembelajaran gabungan.

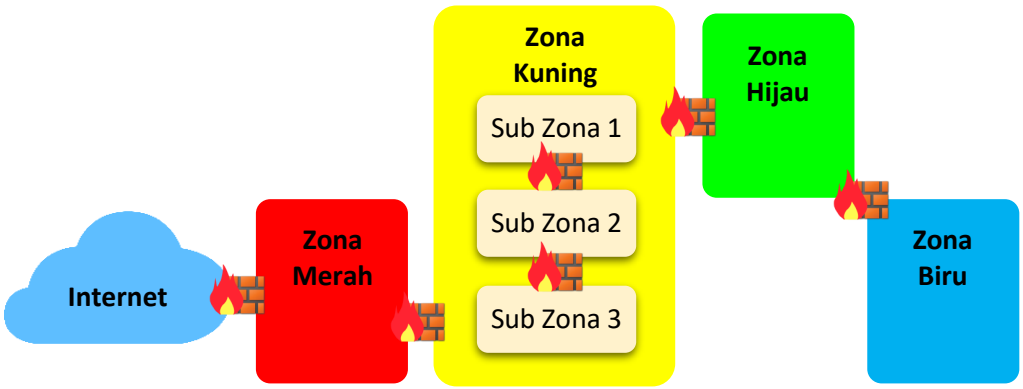
6.4 MENGATASI MASALAH KEPERCAYAAN DENGAN TEKNOLOGI INFRASTRUKTUR

Meskipun pengaturan bisnis dan perjanjian tingkat layanan memberikan insentif bagi orang-orang yang bekerja dalam suatu organisasi untuk memitigasi masalah apa pun yang muncul selama tugas membangun data atau berbagi model, hal tersebut perlu dilengkapi dengan menciptakan infrastruktur untuk model pelatihan dan berbagi informasi. yang akan meminimalkan kemungkinan data atau model disusupi. Ada banyak teknologi infrastruktur, dan semuanya ditargetkan untuk memastikan bahwa isu-isu berbeda yang diuraikan dalam perjanjian kontrak antar bisnis tetap terjaga.

Mekanisme keamanan tradisional untuk pengelolaan data dan model perlu diikuti untuk memastikan kepatuhan terhadap tujuan yang ditentukan dalam pengaturan bisnis. Hal ini dapat mencakup ketentuan seperti statistik data dan parameter model yang dipertukarkan oleh sistem dan tidak disimpan dalam sistem file, yang disimpan hanya dalam memori atau penyimpanan sistem yang mudah menguap dan bahwa semua salinan statistik dan model disimpan. dihancurkan setelah tugas pembelajaran gabungan selesai. Mereka juga dapat menetapkan bahwa akses terhadap komputer apa pun yang terlibat dalam tugas pembelajaran gabungan dibatasi hanya untuk personel yang berwenang.

Beberapa perusahaan mungkin memilih untuk berpartisipasi dalam pembelajaran gabungan hanya dengan membuat zona khusus dalam infrastruktur TI mereka yang dirancang untuk dapat diakses oleh orang-orang di luar perusahaan. Bukan hal yang aneh bagi perusahaan untuk mengklasifikasikan infrastruktur TI mereka ke dalam zona berbeda yang mungkin memiliki sebutan seperti zona merah, zona kuning, zona hijau, dan zona biru. Zona merah terdiri dari komputer yang boleh diakses publik, misalnya. server web yang menghadap publik. Zona kuning akan terdiri dari mesin-mesin yang boleh diakses oleh orang-orang yang tidak bekerja di lokasi perusahaan, misalnya mesin-mesin. jika ada pengaturan untuk kemitraan dengan perusahaan lain dan beberapa server dibuat hanya dapat diakses oleh karyawan terpilih dari perusahaan lokal atau perusahaan mitra.

Zona hijau akan terdiri dari komputer yang digunakan oleh karyawan di lokasi yang diperbolehkan mengaksesnya baik dengan membangun jaringan pribadi virtual, atau dengan kehadiran fisik di gedung milik perusahaan. Zona biru dapat terdiri dari komputer yang hanya dapat diakses ketika hadir secara fisik di dalam gedung perangkat, misalnya. sistem yang mencakup catatan keuangan sensitif perusahaan. Semua zona ini akan dilindungi oleh serangkaian firewall dan perangkat keamanan lainnya, yang memungkinkan komunikasi bolak-balik terbatas antara berbagai komputer dan sistem di zona keamanan lainnya. Mungkin ada sub-zona lain dalam masing-masing zona keamanan ini, misalnya. jika suatu perusahaan bermitra dengan banyak mitra, maka perusahaan tersebut mungkin mempunyai sub-zona dalam zona kuning untuk setiap pengaturan kemitraan, dan dapat mengikuti serangkaian protokol keamanan yang berbeda untuk setiap sub-zona tersebut. Struktur zona dalam suatu perusahaan hipotetis ditunjukkan pada Gambar 6.7.

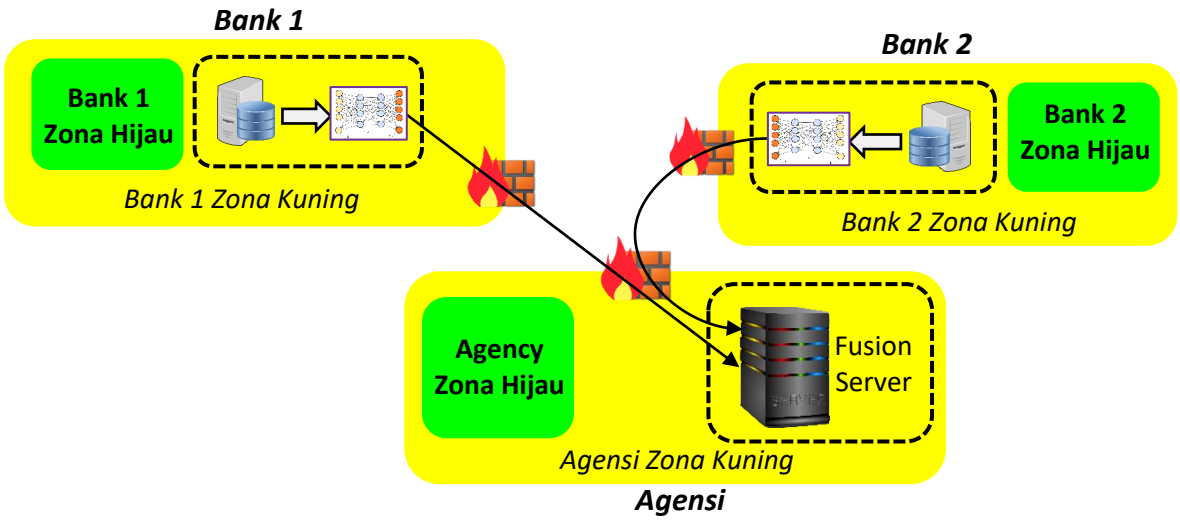


Gambar 6.7 Zona keamanan dalam suatu perusahaan.

Organisasi dan bisnis yang berbeda mungkin merujuk pada beberapa zona keamanan dengan kode warna, skema penomoran, atau nama yang berbeda. Namun, infrastruktur komputer dan komunikasi sering kali dibagi ke dalam zona-zona ini, dengan masing-masing zona disediakan mekanisme alat pemantauan keamanan dan prosedur administratif yang sesuai untuk memastikan bahwa tidak ada hal buruk yang terjadi di dalam perusahaan.

Ketika organisasi yang tidak sepenuhnya percaya satu sama lain terlibat dalam berbagi model atau ringkasan statistik, mereka mungkin memilih untuk menggunakan teknologi isolasi infrastruktur untuk memastikan bahwa mereka tidak terkena kebocoran informasi sensitif.

Perusahaan yang menyediakan layanan fusi data akan menempatkan servernya ke dalam zona kuning, menggunakan firewallnya untuk memastikan bahwa zona tersebut hanya diakses oleh server yang teridentifikasi dari situs penghasil data yang berpartisipasi dalam pengaturan kemitraan untuk berbagi data, dan memelihara log akses ke server mana pun yang mengakses server fusi data. Situs penghasil data itu sendiri juga dapat memasukkan data apa pun yang ingin mereka bagikan, atau model apa pun yang dibangun berdasarkan data tersebut, ke dalam zona atau subzona kuning (atau yang setara) untuk memastikan bahwa situs tersebut terlindungi dari server fusi. Untuk masing-masing zona ini, mereka juga akan mengerahkan perangkat pemantauan keamanan yang sesuai dan langkah-langkah lain untuk meminimalkan kemungkinan kebocoran data atau komponen model dalam proses pembelajaran gabungan.



Gambar 6.8 Contoh pendekatan keamanan infrastruktur.

Situasi hipotetis ketika dua bank bekerja sama dengan sebuah lembaga (misalnya Fannie Mae) untuk menciptakan model bersama ditunjukkan pada Gambar 6.8. Bank-bank tidak saling percaya karena mereka bersaing satu sama lain. Namun, bank mempercayai lembaga tersebut untuk membagikan model mereka kepada lembaga tersebut. Namun, karena kepercayaan tersebut tidak bersifat mutlak, bank akan menempatkan data yang digunakan untuk membangun model di zona kuning, dan membagikan data tersebut kepada lembaga tersebut dengan menggunakan semua mekanisme keamanan yang sesuai yang mereka miliki untuk zona kuning. Badan tersebut juga akan memiliki zona kuningnya sendiri dan hanya akan menempatkan server fusi data dan sumber daya komputasi minimum serta data yang diperlukan untuk melatih model umum. Hanya situs bank yang berpartisipasi yang diperbolehkan mengakses server fusi yang disediakan oleh lembaga. Baik bank maupun lembaga juga akan memantau zona kuning untuk memastikan bahwa situs mereka digunakan dengan baik oleh peserta lain dan tidak terjadi kebocoran informasi.

6.5 AUDIT DAN PENCATATAN

Jika data dan model melintasi zona kepercayaan yang berbeda, teknologi untuk mengaudit data yang dikirim, dan membuktikan bahwa data dan model ditangani dengan benar merupakan persyaratan yang sering kali ditentukan dalam perjanjian bisnis. Untuk berbagi data dan model, berbagai situs yang terlibat dalam pembelajaran gabungan harus menerapkan audit dan pencatatan permintaan dan aliran data yang tepat dalam situs mereka. Semua permintaan yang dibuat di seluruh situs harus dicatat, dan ketika diaudit oleh pihak independen, harus menunjukkan bahwa prosedur yang tepat untuk menangani data dan model yang berasal dari mitra dan situs lain telah diikuti.

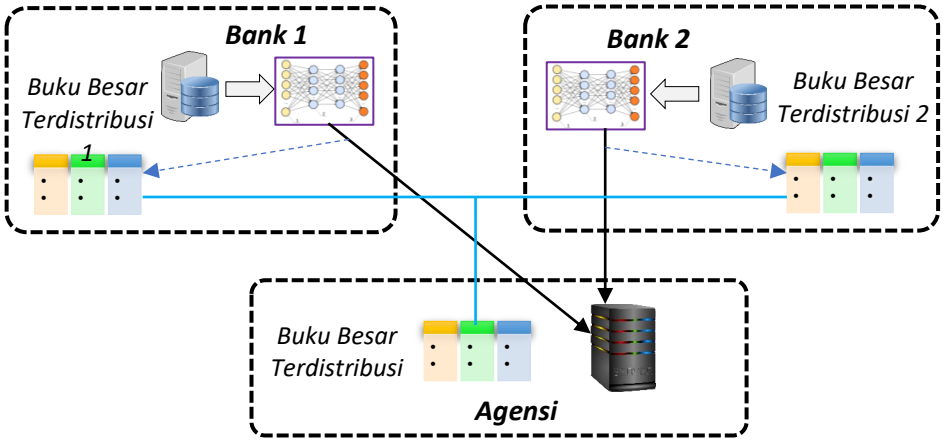
Salah satu pendekatan untuk melacak evolusi model dan data yang digunakan untuk melatih mereka dalam konteks pembelajaran gabungan dapat diberikan dengan menggunakan blockchain. Blockchain atau teknologi buku besar terdistribusi adalah teknologi yang dapat menyimpan catatan tentang beberapa transaksi dalam buku besar terdistribusi, yang tidak dimiliki oleh satu organisasi mana pun, namun dikelola menggunakan konsensus terdistribusi di antara sekelompok besar peserta peer to peer. Buku besar yang terdistribusi ini dapat disimpan untuk melacak bagaimana data digunakan untuk pelatihan, dan parameter model apa yang dipertukarkan di berbagai zona kepercayaan. Dengan mempertahankan informasi tentang proses pembangunan model pembelajaran gabungan, seseorang dapat mempertahankan silsilah informasi yang digunakan untuk membuat model bersama.

Alasan untuk menyimpan informasi garis keturunan dalam buku besar yang terdistribusi adalah karena tidak ada satu entitas pun yang mengendalikan informasi tersebut, sehingga setiap pelanggaran terhadap operasi yang tepat atau diharapkan oleh suatu mesin ketika berkomunikasi dengan mesin di zona kepercayaan lainnya dapat ditangkap. dalam audit pasca operasi atas log ini. Hal ini secara efektif memberikan pemeriksaan terhadap sistem yang memberikan disinsentif besar bagi pihak mana pun yang dengan sengaja melanggar ketentuan dalam perjanjian bisnis.

Informasi silsilah mencatat parameter model yang dilaporkan oleh klien berbeda saat mereka mengakses server, dan parameter model konsolidasi yang dikembalikan oleh server fusi. Informasi garis keturunan juga dapat menangkap parameter lain dan pertukaran data yang terjadi antara dua pihak yang berada di zona kepercayaan berbeda, termasuk pencatatan setiap parameter dan kebijakan yang dipertukarkan selama tahap pra-pemrosesan pembelajaran gabungan, yaitu berbagai prosedur yang dijelaskan dalam Bab 4.

Secara arsitektural, pemeliharaan dan penyimpanan operasi penebangan ditunjukkan secara sederhana pada Gambar 6.9. Contoh yang ditampilkan sama seperti pada Gambar 6.8 dengan dua bank dan sebuah agen. Interaksi dalam pembelajaran gabungan terjadi melalui interaksi yang terjadi pada garis hitam yang terjual yang memanggil perintah pada server di lingkungan berbeda, meminta fungsi yang sesuai untuk dijalankan. Interaksi ini biasanya akan melewati firewall dan perangkat keamanan lainnya yang tidak ditampilkan pada gambar, namun akan selalu ada di infrastruktur apa pun. Selain itu, setiap perusahaan yang terlibat dalam tugas pembelajaran gabungan mengoperasikan buku besar yang didistribusikan di antara mereka sendiri. Satu bagian dari buku besar terdistribusi berjalan di masing-masing

perusahaan, dan semuanya berinteraksi bersama untuk memelihara buku besar terdistribusi. Perusahaan dapat mencatat semua informasi yang dikirim atau diterima dalam operasi bersama satu sama lain, dan menyimpannya dalam buku besar untuk audit selanjutnya. Protokol buku besar terdistribusi memastikan bahwa catatan transaksi ini disimpan dengan cara yang tidak dapat dipalsukan atau disangkal dengan mudah.



Gambar 6.9 Pencatatan log berbasis buku besar terdistribusi.

Teknologi audit dan logging menyediakan mekanisme bagi mitra untuk memvalidasi dan menyatakan bahwa mereka mematuhi perjanjian bisnis yang diberikan kepada mereka untuk berbagi data untuk pembelajaran terdistribusi. Hal ini juga memberikan disinsentif bagi siapa pun yang mencoba memanipulasi data atau model yang ditawarkan dalam proses pembelajaran gabungan. Satu-satunya masalah dalam mempertahankan silsilah semua interaksi yang terjadi dengan cara ini adalah biaya overhead yang terkait dengan pencatatan semua informasi dalam buku besar yang didistribusikan. Biaya dapat dikurangi dengan tidak mencatat setiap interaksi yang melintasi zona kepercayaan, namun hanya mencatat sebagian interaksi. Alternatifnya, tergantung pada tingkat kepercayaan antar pihak, pencatatan dan pemantauan garis keturunan dapat dilakukan oleh pihak ketiga, yang dipercaya oleh semua pihak, dan dapat melacak transaksi yang terjadi di antara mereka.

Keuntungan mendapatkan data tambahan untuk pelatihan perlu diseimbangkan dengan biaya tambahan dan biaya pemeliharaan informasi garis keturunan yang didistribusikan, dan penilaian biaya-manfaat perlu dilakukan untuk menentukan tindakan mana yang terbaik dari sudut pandang bisnis, untuk melakukan pembelajaran gabungan tanpa pencatatan apa pun, untuk tidak melakukan pembelajaran gabungan, atau untuk melakukan pembelajaran gabungan dengan pencatatan garis keturunan.

6.6 PENDEKATAN BERBASIS ENKRIPSI

Tujuan yang mendasari perjanjian bisnis, keamanan infrastruktur, dan kemampuan audit adalah untuk memberikan jaminan yang cukup kepada mitra yang terlibat dalam tugas pembelajaran gabungan bahwa terdapat risiko minimal dalam pertukaran model di berbagai zona kepercayaan. Dalam banyak konteks bisnis, hal ini mungkin cukup untuk meyakinkan

mitra agar berbagi model mereka dan terlibat dalam pelatihan model bersama. Sebagian besar dunia usaha menyadari bahwa tidak mungkin mendapatkan kepercayaan mutlak, sehingga risiko kebocoran informasi harus dibandingkan dengan manfaat membangun pengetahuan dari data tambahan. Selama risiko kehilangan informasi model lebih rendah daripada nilai bisnis yang diperoleh dengan membuat model bersama, banyak bisnis akan menerima risiko kehilangan informasi dan bertukar parameter model satu sama lain untuk membuat model bersama. Namun, mungkin ada beberapa bisnis yang mungkin tidak bersedia membagikan model mentah di luar zona kepercayaan mereka, bahkan dengan semua prosedur yang ada.

Untuk bisnis seperti itu, pendekatan berbagi data dan model berdasarkan teknologi enkripsi mungkin memberikan solusi yang dapat diterima. Saat menggunakan enkripsi, parameter model dan bahkan ringkasan statistik tentang data tidak ditransfer secara jelas ke seluruh zona kepercayaan. Sebaliknya, hanya formulir terenkripsi yang ditransfer ke seluruh zona kepercayaan. Enkripsi adalah proses yang mengubah setiap bagian data (teks biasa) menjadi representasi alternatif (teks sandi) sehingga hanya orang yang memiliki akses ke beberapa rahasia (kunci) yang dapat mengekstraksi teks biasa asli dari teks sandi. . Secara umum, enkripsi menghancurkan struktur informasi yang terkandung dalam teks biasa, dan teks tersandi tampak seperti data acak. Informasi sensitif yang dipertukarkan melalui Internet biasanya dienkripsi.

Teknik enkripsi modern di Internet biasanya mengikuti kombinasi dua teknik (i) enkripsi berbasis kunci simetris dan (ii) enkripsi berbasis kunci publik. Dalam enkripsi berbasis kunci simetris, kedua pihak yang terlibat dalam komunikasi mengetahui kunci rahasia. Pengirim menggunakan kunci rahasia untuk mengubah teks asal menjadi teks sandi dan penerima menggunakan kunci rahasia yang sama untuk melakukan transformasi kebalikan dari teks sandi menjadi teks biasa. Dalam skema berbasis kunci publik, masing-masing pihak memiliki sepasang kunci, satu kunci dirahasiakan (kunci privat) dan kunci lainnya (kunci publik) tersedia bagi siapa saja yang ingin berkomunikasi dengan pihak tersebut. Pasangan kunci dihasilkan sehingga teks yang dienkripsi dengan kunci pribadi dapat diubah menjadi teks biasa dengan kunci publik. Pengirim akan mengenkripsi teks biasa dengan kunci publik penerima, dan penerima melakukan transformasi terbalik menggunakan kunci pribadi rahasia. Kunci publik biasanya tersedia dalam representasi digital standar yang disebut sertifikat yang berisi kunci publik yang terkait dengan suatu pihak. Sertifikat ditandatangani dengan kunci pribadi dari beberapa situs tepercaya yang kunci publiknya diketahui semua orang. Dalam komunikasi berbasis Internet, klien akan menggunakan sertifikat untuk mendapatkan kunci publik server, menggunakan kriptografi kunci publik untuk menegosiasikan kunci rahasia dengan server, dan menggunakan kunci rahasia untuk mengenkripsi dan mendekripsi pesan selama beberapa waktu. Kunci rahasia dapat disegarkan secara berkala. Karena enkripsi dan dekripsi berbasis kunci publik lebih mahal secara komputasi dibandingkan enkripsi kunci simetris, mekanisme ini memberikan skema yang aman dan efisien untuk komunikasi yang aman. Survei pendekatan enkripsi dapat ditemukan dalam referensi.

Saat menggunakan algoritma enkripsi tradisional, teks tersandi menjadi mendekati representasi acak dari teks biasa, dan tidak dapat diproses. Namun, bentuk algoritma enkripsi baru telah diusulkan yang memungkinkan operasi seperti penjumlahan dan perkalian dilakukan pada teks sandi terenkripsi. Algoritme enkripsi kelas ini menunjukkan homomorfisme, yang didefinisikan sebagai kemampuan untuk mempertahankan hubungan yang ada dalam bentuk asli suatu data dalam bentuk data yang diubah (dalam hal ini dienkripsi). Kelas algoritma enkripsi ini, yang disebut algoritma enkripsi homomorfik, memungkinkan berbagi model dan data dengan pihak lain yang dapat melakukan operasi pada model dan data tersebut tanpa mengungkapkan teks biasa. Ada dua kategori skema enkripsi homomoprhic, yang dikenal sebagai enkripsi homomoprhic penuh dan enkripsi homomorfik parsial, seperti yang dijelaskan dalam subbagian berikut.

6.6.1 Enkripsi Sepenuhnya Homomorfik

Skema enkripsi yang sepenuhnya homomorfik adalah skema enkripsi yang memenuhi dua properti berikut:

- Homomorfi Aditif: Versi terenkripsi dari jumlah dua teks biasa adalah jumlah versi terenkripsi dari dua nilai teks biasa
- Homomorfi Perkalian: Versi terenkripsi dari produk dua teks biasa adalah produk dari versi terenkripsi dari dua nilai teks biasa

Dinyatakan dalam versi alternatif, seharusnya x dan y adalah dua angka dan versi terenkripsinya adalah $E\ (x)$ dan $E\ (y)$. Maka kedua properti tersebut adalah:

- Homomorfi Aditif: $E\ (x + y) = E\ (x) + E\ (y)$
- Homomorfi Perkalian: $E\ (x.y) = E\ (x).E\ (y)$

Algoritma enkripsi yang memenuhi kedua sifat homomoprhik ini dikenal sebagai algoritma enkripsi homomorfik penuh. Algoritme enkripsi homomorfik pertama yang sepenuhnya diusulkan pada tahun 2009, dengan penelitian selanjutnya mengarah pada beberapa algoritma enkripsi lain yang memenuhi properti tersebut. Perkembangan algoritma tersebut berarti bahwa setiap operasi yang dapat dilakukan pada data yang jelas juga dapat dilakukan pada data yang dienkripsi. Setiap operasi algoritmik dapat dilakukan sebagai kombinasi dari berbagai operasi penjumlahan dan perkalian. Hasilnya, algoritma homomorfik sepenuhnya memungkinkan komputasi pada data yang dienkripsi. Hal ini memungkinkan perusahaan mengirim data ke server cloud untuk melakukan pemrosesan kompleks tanpa mengungkapkan data ke server cloud. Kemampuan ini memungkinkan mitra yang terlibat dalam pembelajaran gabungan untuk bertukar parameter model dan statistik data dalam format terenkripsi.

Tantangan utama yang terkait dengan enkripsi homomoprhic sepenuhnya adalah kinerja yang dihasilkannya dalam sistem. Implementasi standar dari operasi yang berjalan menggunakan algoritma ini berjalan 14 kali lipat lebih lambat (yaitu 1014 atau 100 triliun kali lebih lambat) dibandingkan operasi teks biasa. Sejak itu, beberapa peneliti telah mengusulkan peningkatan implementasi algoritmik menggunakan akselerator perangkat keras dan pemrosesan paralel, yang dapat mengurangi operasi terenkripsi agar bekerja dalam 2 kali lipat (2 kali lebih lambat). Namun, hal ini masih merupakan biaya tambahan yang signifikan. Di

masa depan, skema yang mengeksploitasi enkripsi homomoprhic sepenuhnya akan dapat dilaksanakan seiring dengan tercapainya lebih banyak peningkatan dalam penerapannya.

6.6.2 Model Pembelajaran Homomorfik Parsial

Dalam algoritma homomorfik parsial, hanya satu dari dua sifat homomoprhic yang dipertahankan, yaitu algoritma enkripsi mematuhi properti homomoprhic aditif atau hanya mematuhi properti homomoprhic perkalian. Beberapa algoritma yang diusulkan untuk enkripsi menunjukkan salah satu sifat berikut. Ritme algo RSA, yang merupakan salah satu algoritma enkripsi kunci publik pertama, menunjukkan homomorfisme multiplikatif. Algoritma El Gamal adalah algoritma enkripsi kunci publik populer lainnya yang menunjukkan homomorfisme multiplikatif. Algoritma Paillier menunjukkan sifat homomorfik aditif. Keuntungan dari algoritma homomorfik parsial adalah bahwa overhead kinerja mereka dalam enkripsi dan dekripsi dapat diterima dalam pelaksanaan bisnis normal dan algoritma ini digunakan dalam aplikasi dan implementasi saat ini.

Karena hanya satu operasi, baik penjumlahan atau perkalian yang dapat dilakukan pada konten terenkripsi sambil menjaga hubungannya, enkripsi homomorfik sebagian dapat digunakan untuk mengaktifkan bentuk pembelajaran gabungan di mana operasi yang dilakukan oleh server federasi dilakukan pada konten terenkripsi. Pendekatan ini telah diusulkan untuk mengatasi konfigurasi di mana server fusi tidak dipercaya. Tugas-tugas yang harus dilakukan oleh server fusi direstrukturisasi sehingga hanya perlu menjalankan satu jenis aplikasi, baik multiplikatif atau aditif. Jika algoritma enkripsi menunjukkan homomorfi aditif, maka operasinya disusun menjadi penjumlahan saja. Jika ritme algo enkripsi menunjukkan homomorfi perkalian, maka operasinya disusun menjadi perkalian saja.

Misalnya, jika suatu algoritme menggunakan rata-rata gabungan parameter jaringan saraf seperti dijelaskan di Bagian 3.4, server federasi melakukan tugas menjumlahkan parameter model dan kemudian membaginya dengan jumlah peserta untuk mendapatkan rata-rata. Operasi ini mencakup penjumlahan dan perkalian (dengan $1/N$ dimana N adalah jumlah peserta). Operasi dapat disusun sehingga hanya penambahan yang dilakukan di server, dan mengingat jumlah situs yang berpartisipasi dalam proses pembelajaran gabungan, operasi perkalian dapat dilakukan oleh masing-masing situs itu sendiri. Dengan asumsi algoritma enkripsi perkalian aditif digunakan, klien fusi dapat mengirim model ke server fusi dalam format terenkripsi, dan server dapat melakukan operasi penambahan pada konten terenkripsi. Jika algoritma homomorfik perkalian digunakan, eksponen parameter perlu dikirim, karena hasil perkalian komponen terenkripsi, dan pengambilan logaritma pada klien akan menghasilkan operasi penjumlahan.

Agar skema dapat berfungsi, kunci enkripsi harus dinegosiasikan sehingga semua klien fusi menggunakan kunci enkripsi yang sama. Kunci ini dapat disediakan oleh server di zona kepercayaan klien, atau dengan memilih salah satu situs klien sebagai generator kunci. Server di zona kepercayaan lain dapat membantu dalam pemilihan ini, atau klien fusi pertama dapat secara otomatis menjadi server kunci untuk semua klien lain yang bergabung selanjutnya.

Ketika zona kepercayaan berbeda, homomoprhy parsial mengharuskan operasi yang akan dilakukan dirumuskan ulang sedemikian rupa sehingga dilakukan hanya dengan

menggunakan semua operasi penjumlahan atau perkalian ketika perhitungan perlu dikirim ke entitas di luar zona kepercayaan. Dalam beberapa kasus, hal ini dapat menambah kompleksitas tugas secara signifikan, dan terkadang bahkan tidak dapat dilakukan. Untuk zona kepercayaan dimana reformulasi ini dimungkinkan, enkripsi homomorfik parsial dapat memberikan pendekatan yang sangat layak.

6.7 PENDEKATAN BERBASIS PRIVASI DIFERENSIAL

Dalam konfigurasi zona kepercayaan di mana klien tidak saling percaya satu sama lain, mereka mungkin tidak ingin mengirimkan parameter model ke server untuk dibagikan dengan klien lain. Ketika hanya ada satu klien lain, pembagian tersebut dapat mengungkapkan parameter ke klien lain. Demikian pula, jika klien lain berkolusi bersama, mereka dapat menentukan parameter model klien. Jika penting bagi klien fusi untuk tidak memaparkan parameter modelnya kepada klien lain, klien fusi dapat memasukkan beberapa derau ke parameter model, namun menambahkan derau dengan cara yang tidak berdampak pada hasil akhir jaringan saraf yang sedang dibangun.

Area umum yang membahas bagaimana kebisingan harus dieksplorasi adalah bidang privasi diferensial, yang melihat berbagai sifat statistik dari sistem, dan bagaimana kebisingan dapat diperkenalkan dengan cara yang cerdas. Daripada membahas kompleksitas semua aspek privasi yang berbeda, kami hanya akan mengilustrasikan masalahnya dengan kasus di mana properti statistik dari sistem, misalnya. rata-rata dan varians rentang suatu fitur perlu ditentukan pada seluruh data yang disimpan di semua lokasi. Hal ini diperlukan untuk mendapatkan penskalaan data yang konsisten di semua lokasi hingga rata-rata nol dan variansi 1, seperti yang dijelaskan dalam pendekatan yang dijelaskan di Bagian 4.1.2 untuk memastikan penskalaan yang konsisten di seluruh lokasi. Namun, situs mana pun tidak ingin orang lain mengetahui nilai rata-rata atau variansnya masing-masing, namun mereka semua ingin mempelajari nilai mean dan varians di seluruh situs secara bersamaan.

Jika situs saling mempercayai, setiap situs klien fusi dapat melaporkan mean dan variansnya beserta jumlah titik data yang dimilikinya. Server fusi akan menghitung rata-rata tertimbang dari mean dan varians yang disediakan untuk semua pihak. Namun, jika klien tidak mempercayai server, dan khawatir akan kebocoran informasi ke klien atau server lain, klien mungkin tidak ingin melaporkan datanya secara jelas. Penambahan noise memungkinkan penghitungan nilai tersebut di semua lokasi tanpa kebocoran informasi.

Untuk mendapatkan rata-rata tertimbang dengan cara privat yang berbeda, situs-situs tersebut akan berpartisipasi dalam dua putaran proses rata-rata privat diferensial. Pada bagian pertama, mereka masing-masing akan mencari jumlah rata-rata titik di seluruh lokasi. Hal ini memungkinkan mereka mendapatkan jumlah total titik data di semua situs, dengan mengalikan rata-rata ini dengan jumlah situs yang berpartisipasi. Pada putaran berikutnya, situs-situs tersebut akan menghitung rata-rata rata-rata tertimbang dari situs mereka, yaitu mengalikan rata-rata nilai yang mereka miliki dengan pecahan dari jumlah keseluruhan titik data, dan menghitung rata-rata privat diferensial dari titik data yang sama.

Ketika konfigurasi kepercayaan adalah server fusi yang tidak tepercaya dan klien fusi yang tidak tepercaya, salah satu cara untuk menghitung rata-rata privat diferensial secara keseluruhan adalah dengan setiap situs memasukkan beberapa kesalahan dalam rata-ratanya yang mereka laporkan ke situs fusi. Namun, ada koordinasi dalam cara munculnya kebisingan, misalnya dengan menggunakan kendaraan. setiap situs telah sepakat bahwa mereka akan memperkenalkan kebisingan yang diambil dari distribusi Gaussian dengan rata-rata nol dan varian umum yang disepakati oleh semua situs klien fusi. Jika setiap lokasi menambahkan kebisingan acak ke nilai rata-rata berdasarkan distribusinya, maka rata-rata kebisingan di seluruh lokasi kemungkinan akan menghilangkan satu sama lain, dengan asumsi jumlah lokasi masuk akal. Dengan situs fusi N, rata-rata kebisingan yang ditambahkan secara kumulatif diharapkan akan hilang dengan varians sebesar 1 + N kali varians yang dipilih semua orang untuk digunakan. Dengan mengulangi proses tersebut beberapa kali, rata-rata kumulatif dapat dihitung, yang kemungkinan besar akurat meskipun tidak ada situs yang pernah mengirimkan rata-rata sebenarnya ke situs fusi. Demikian pula, klien fusi lainnya tidak dapat menebak nilai rata-rata yang benar di situs lain mana pun.

Di luar contoh sederhana ini, teknik statistik dalam privasi diferensial menyediakan mekanisme untuk menambahkan gangguan sehingga jenis properti lain tentang distribusi parameter apa pun (misalnya nilai maksimum, nilai minimum, persentil, dll.) dapat dihitung tanpa membagikan informasi yang benar dari pihak mana pun. dari situs-situs tersebut. Hal ini memungkinkan adanya cara untuk melindungi parameter model, dan ringkasan informasi statistik saat berpartisipasi dalam proses fusi. Pendekatan penambahan noise akan memiliki overhead yang jauh lebih sedikit dibandingkan pendekatan berbasis enkripsi, dan mungkin dapat diterima untuk hubungan kepercayaan tertentu.

6.8 RINGKASAN

Secara umum, masalah kepercayaan di kalangan dunia usaha dapat diselesaikan melalui pengaturan bisnis dan mekanisme yang tepat untuk keamanan infrastruktur, serta kesepakatan di antara para pihak untuk mengaudit dan mencatat data yang dibagikan atau diubah. Jika mekanisme tersebut tidak dapat diterima, mekanisme penyembunyian data, termasuk penggunaan enkripsi homomorfik dan teknik privasi diferensial, dapat digunakan.

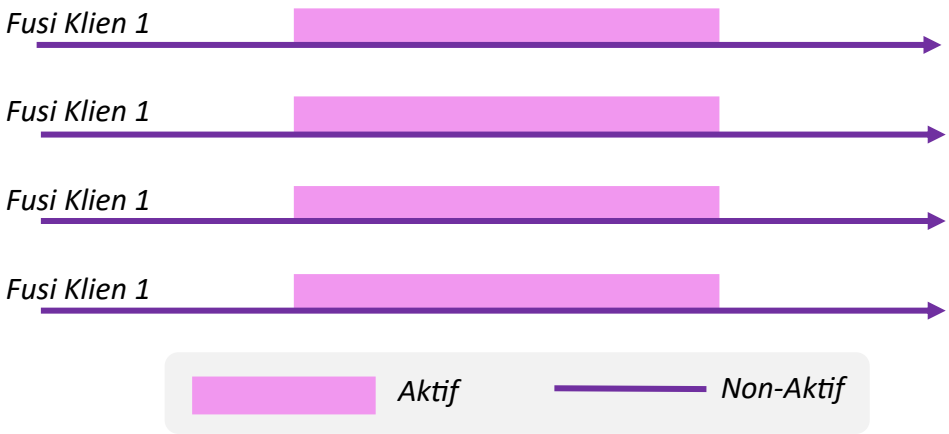
BAB 7
MENGATASI MASALAH SINKRONISASI DALAM PEMBELAJARAN FEDERASI

Salah satu asumsi yang dibuat dalam algoritma pembelajaran gabungan Naif yang dijelaskan dalam Bab 3 adalah bahwa klien fusi yang berbeda berjalan secara bersamaan dan tersinkronisasi satu sama lain. Operasi bersamaan ini menyederhanakan pengoperasian server fusi. Namun, dari sudut pandang logistik dan operasional, pelatihan yang dilakukan secara bersamaan menimbulkan kompleksitas yang signifikan dan mengurangi kelangsungan solusi pembelajaran gabungan. Solusi pembelajaran gabungan yang praktis harus memungkinkan klien fusi untuk aktif pada waktu berbeda dan jadwal yang ditentukan oleh klien fusi, dibandingkan beroperasi di bawah kendali server fusi jarak jauh.

Pada bagian pertama bab ini, kita membahas isu-isu yang terkait dengan sinkronisasi. Bagian lain dari bab ini membahas beberapa pendekatan yang dapat digunakan untuk mengatasi permasalahan tersebut dan menangani situasi di mana operasi secara bersamaan tidak memungkinkan.

7.1 IKHTISAR MASALAH SINKRONISASI

Untuk mengilustrasikan masalah sinkronisasi, mari kita asumsikan bahwa ada beberapa N klien fusi dan satu server fusi. Agar algoritme naif pada Bab 3 berfungsi, asumsinya adalah semua klien fusi dan server fusi aktif secara bersamaan. Diagram waktu yang menunjukkan kapan klien dan server fusi yang berbeda harus aktif akan serupa dengan yang ditunjukkan pada Gambar 7.1. Semua situs fusi dan server fusi harus aktif dan berjalan pada waktu yang sama.



Gambar 7.1 Diagram aktivasi untuk pembelajaran gabungan yang naif.

Klien fusi di semua situs harus dimulai secara fisik pada waktu yang hampir bersamaan dengan dimulainya server fusi. Sedikit perubahan yang mengejutkan pada waktu mulai mungkin diperbolehkan karena klien yang berbeda bergabung dengan server, namun server

harus menunggu hingga semua klien terhubung. Setelah semua klien terhubung ke server, mereka dapat menjalani prosedur yang diperlukan untuk mengatasi masalah ketidakcocokan data dan kemiringan data serta melatih model bersama-sama.

Meskipun sinkronisasi ini mungkin tampak sebagai persyaratan yang sepele, namun hal ini dapat menimbulkan tantangan besar dalam implementasi di lingkungan terdistribusi, khususnya ketika lokasi yang berbeda berada di bawah kendali administratif yang berbeda. Mengingat tata letak fisik lingkungan pembelajaran gabungan, lokasi klien fusi yang berbeda mungkin berada di negara berbeda, atau di kota berbeda di negara besar. Jika mereka berlokasi di dalam fasilitas industri atau pemerintah, mereka kemungkinan besar diamankan dengan perangkat keamanan yang memastikan bahwa data tidak berpindah ke luar fasilitas mereka, mencegah permintaan koneksi masuk dan memblokir sebagian besar lalu lintas, sehingga memerlukan izin eksplisit untuk komunikasi jaringan apa pun yang diizinkan.

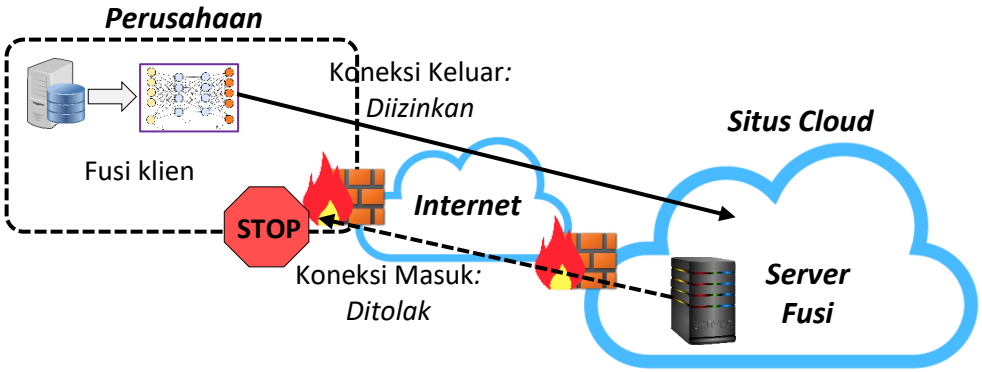
Kompleksitas operasional muncul karena kebutuhan untuk memulai beberapa klien fusi di lokasi berbeda pada waktu yang sama. Salah satu pilihannya adalah dengan meminta istrator admin di setiap situs untuk memulai program agar berjalan pada waktu tertentu. Administrator dapat menjalankan program pada waktu mulai yang telah diatur sebelumnya, atau dapat memerintahkan server untuk memulai program secara otomatis pada waktu tertentu. Terlepas dari bagaimana koordinasi ini dilakukan, ini merupakan langkah manual yang memerlukan sedikit overhead, dan ada kemungkinan bahwa kesalahan yang dilakukan oleh administrator dapat mengacaukan proses dengan tidak memulai beberapa layanan yang diperlukan untuk operasi ini. Bahkan dengan skrip otomatis, beberapa klien mungkin gagal untuk memulai proses pembuatan model. Selanjutnya, jika waktunya perlu diubah, mis. jika waktu yang dijadwalkan untuk pembelajaran model bertepatan dengan waktu henti server fusi yang dijadwalkan, penyesuaian skrip untuk waktu yang dimodifikasi perlu dikoordinasikan lagi.

Pilihan yang lebih baik adalah setiap klien fusi menyediakan mekanisme yang melaluinya situs jarak jauh, misalnya. situs server fusi, dapat memulai semua klien fusi pada waktu yang tepat. Jika situs klien fusi mengizinkan jenis pemanggilan masuk seperti itu, aktivasi tersinkronisasi akan menjadi hal yang mudah untuk memulai program. Daripada memiliki tim manusia yang terdistribusi, satu manusia dapat memulai proses pembelajaran gabungan dari server fusi. Tantangan operasional di sini berasal dari kebutuhan keamanan perusahaan. Untuk mendapatkan keamanan yang lebih baik, sebagian besar situs perusahaan membatasi pengguna jarak jauh untuk mengakses server di lokasi mereka. Mesin dapat terhubung ke server, mis. klien fusion dapat memulai koneksi jarak jauh ke server fusion, tetapi klien fusion tidak diperbolehkan menerima permintaan koneksi dari luar firewall. Jika server fusi beroperasi sebagai layanan yang dihosting di cloud, kebutuhan untuk menyesuaikan izin pada klien fusi akan menjadi hambatan yang signifikan dalam penggunaan layanan cloud.

Tantangan koordinasi seperti ini akan muncul jika konsorsium mencoba melatih model umum menggunakan pembelajaran gabungan. Mari kita asumsikan bahwa semua anggota koalisi telah mengumpulkan data dan memeliharanya sedemikian rupa sehingga model umum

dapat dibangun. Mereka semua memiliki sumber daya dan server di lokasi untuk melatih model dan menukarnya dengan server fusi, yang dihosting di layanan cloud milik konsorsium. Setiap perusahaan telah mengikuti praktik keamanan di lokasi mereka, seperti yang dijelaskan dalam Bab 6 dan menempatkan data dan mesin yang menjalankan klien fusi ke dalam zona keamanan yang sesuai. Agar ilmuwan data yang menjalankan server fusi yang dihosting di cloud dapat memanggil klien fusi untuk memulai proses pelatihan secara sinkron, setiap perusahaan harus mengizinkan panggilan masuk dari layanan cloud ke klien fusi mereka. Jenis pemanggilan layanan eksternal pada server ini biasanya sulit (walaupun bukan tidak mungkin) diperoleh di banyak perusahaan, mengingat kekhawatiran mereka mengenai paparan keamanan sistem yang dapat diakses melalui Internet. Sebagian besar tim keamanan perusahaan lebih memilih untuk hanya mengizinkan panggilan keluar dari server yang berada dalam kendali mereka.

Masalahnya diilustrasikan dari sudut pandang satu perusahaan yang menjalankan klien fusinya pada Gambar 7.2. Perusahaan telah menempatkan klien data dan fusi sehingga dapat mengakses server fusi yang berjalan di layanan cloud eksternal. Perusahaan tidak ingin datanya diungkapkan ke server fusi, atau ke pihak lain, itulah sebabnya perusahaan melakukan pembelajaran gabungan dibandingkan hanya mentransfer data ke situs untuk fusi. Ini akan mengerahkan perangkat keamanan untuk melindungi datanya. Sangat umum di perusahaan untuk menegakkan keamanan dengan hanya mengizinkan koneksi keluar dari lokasinya, dan membatasi koneksi masuk apa pun di server yang dihostingnya ke lokasi perusahaan. Oleh karena itu, perusahaan akan mengizinkan permintaan koneksi keluar dari lokasinya ke layanan cloud, namun menolak segala upaya koneksi masuk. Hal ini membuat tugas sinkronisasi dari server fusi menjadi sulit, karena server fusi tidak dapat memanggil klien fusi di perusahaan untuk memulai operasinya.

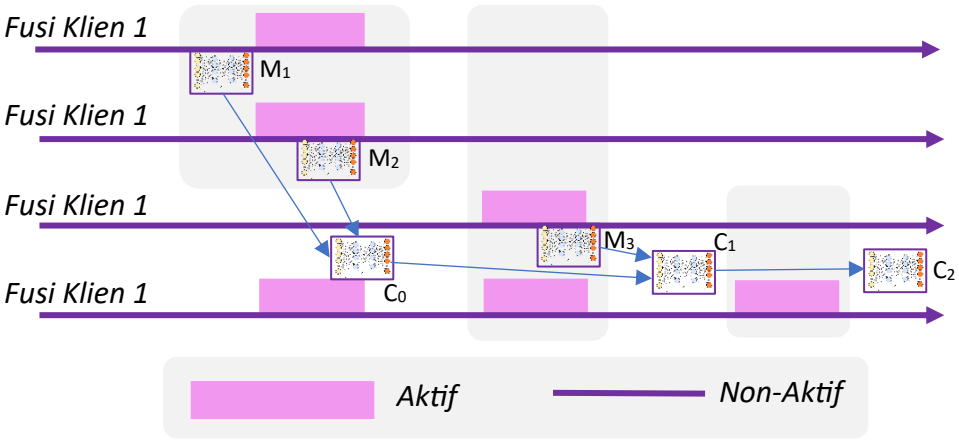


Gambar 7.2 Masalah keamanan untuk pembelajaran gabungan dalam konteks perusahaan.

Tantangan serupa muncul jika perusahaan penyedia layanan, seperti Fannie Mae, menawarkan layanan untuk membangun model prediksi risiko hipotek yang lebih baik bagi bank. Sebagai perusahaan tersendiri, meski dipercaya oleh bank, Fannie Mae akan kesulitan memastikan bahwa semua klien fusi bank siap dan beroperasi pada saat yang sama sehingga

model yang sesuai dapat dibangun. Dimungkinkan untuk memiliki waktu operasi terjadwal tersebut, namun akan jauh lebih mudah jika bank dapat dengan mudah mengunggah model terlatih mereka ke Server Fannie Mae Fusion sesuai keinginan mereka.

Ada beberapa teknik yang memungkinkan sinkronisasi bahkan ketika koneksi masuk tidak diperbolehkan. Klien fusi dapat meminta permintaan berkala ke server fusi untuk memeriksa apakah sudah waktunya memulai proses pelatihan aktif. Meskipun agak tidak efisien, hal ini memungkinkan adanya pendekatan untuk menyinkronkan operasi, setelah server fusi mengetahui bahwa semua klien fusi tersedia untuk sesi pelatihan yang disinkronkan. Pilihan lainnya adalah dengan menggunakan protokol seperti websocket yang memungkinkan pesan dikirim bolak-balik antara klien dan server dan memungkinkan terjadinya sinkronisasi. Protokol ini melintasi firewall melalui port yang digunakan untuk lalu lintas web, dan secara efektif mengimplementasikan mekanisme sinkronisasi pemeriksaan periode dalam implementasi protokol itu sendiri. Pendekatan ketiga yang tidak menggunakan teknologi, namun bergantung pada pengaturan bisnis, adalah menerapkan mekanisme keamanan, audit, dan pencatatan yang tepat sehingga server fusi diperbolehkan melakukan panggilan masuk.



Gambar 7.3 Perubahan grup menyebabkan masalah sinkronisasi.

Sekalipun pengaturan yang tepat untuk memungkinkan sinkronisasi operasi klien dan server yang berbeda diterapkan, masalah yang masih tetap ada adalah perubahan keanggotaan konsorsium seiring berjalannya waktu. Situasinya ditunjukkan pada Gambar 7.3. Konsorsium awalnya dimulai dengan 2 anggota, dan kedua anggota dapat melatih model bersama menggunakan algoritma pembelajaran gabungan. Model yang disediakan oleh kedua perusahaan adalah M_1 dan M_2, yang digabungkan oleh server fusi menjadi model C_0. Beberapa bulan setelah konsorsium terbentuk, ada perusahaan baru yang ingin bergabung dengan konsorsium. Perusahaan ketiga akan menyumbangkan model M_3 yang dilatih berdasarkan datanya sendiri. Namun, ketika model tersebut tersedia dari perusahaan ketiga, dua perusahaan lainnya mungkin telah menonaktifkan infrastruktur yang mereka siapkan untuk berbagi data dengan konsorsium. Pemimpin konsorsium harus menambah model C_0 dengan model baru M_3 yang disediakan oleh perusahaan baru, namun data dari dua mitra awal tidak

lagi tersedia. Pendekatan yang mengandalkan semua orang untuk melakukan pelatihan bersama-sama pada waktu yang sama tidak akan berhasil. Sebaliknya, diperlukan pendekatan yang dapat menggabungkan model C_0 dan M_3 untuk menghasilkan model baru C_1.

Tampaknya mengambil model C_0 dan meminta anggota untuk melatihnya lebih lanjut pada data mereka mungkin merupakan cara untuk mengatasi masalah tersebut. Sayangnya hal ini tidak berhasil pada sebagian besar kasus. Banyak algoritma pelatihan model yang populer, khususnya yang umum digunakan untuk jaringan saraf dalam, mengalami masalah yang disebut catastrophic forgetting. Ketika jaringan saraf yang sudah terlatih dilatih dengan data baru, mereka mempelajari pola-pola baru tetapi melupakan pola-pola yang telah mereka pelajari sebelumnya.

Lupa yang sangat parah adalah salah satu masalah utama yang membuat pembelajaran gabungan menjadi sulit ketika data tidak seimbang atau terpartisi seperti yang dijelaskan dalam Bab 5. Misalkan ada tiga situs, masing-masing dengan data yang hanya dimiliki oleh satu kelas, dan masing-masing situs memiliki data milik kelas yang berbeda. Saat model dilatih pada data dari situs 1, model hanya akan memprediksi setiap masukan sebagai milik kelas 1. Ketika model ini kemudian dilatih secara bertahap pada data milik situs 2, diharapkan model tersebut mampu membedakan antara kelas 1 dan kelas 2. Namun, model tersebut melupakan karakteristik kelas 1 dan, setelah melatih data dari situs 2, sebagian besar akan memprediksi keluaran sebagai milik kelas 2. Demikian pula, model akan sangat memprediksi masukan apa pun sebagai milik model 3 setelahnya. pelatihan di situs 3, sebagian besar melupakan kontribusi apa pun yang diberikan oleh salah satu dari dua situs sebelumnya. Lupa ini belumlah tuntas, dan jika seseorang terus mengulangi pelatihan data di semua situs selama beberapa putaran, model tersebut pada akhirnya akan belajar mengklasifikasikan ketiga kelas dengan benar. Retensi pengetahuan sisa adalah alasan mengapa pelatihan berulang selama beberapa putaran data berfungsi untuk algoritma yang menggunakan algoritma inkremental dan pelatihan berulang yang dijelaskan di Bagian 3.5.1. Pengulangan ini tidak mungkin dilakukan jika anggota baru bergabung dengan konsorsium.

Masalah mendasarnya adalah seseorang tidak dapat mengandalkan data yang tersedia dari kontributor asli ketika anggota baru bergabung. Seseorang mungkin ingin mengatasinya melalui perjanjian bisnis yang menetapkan bahwa data yang disumbangkan harus tersedia kembali ketika anggota baru bergabung. Namun hal ini membuat kewajiban bergabung dalam konsorsium semakin memberatkan.

Masalah lain dapat muncul jika salah satu anggota konsorsium memutuskan untuk keluar, dan bersikeras agar pola dari datanya dihapus dari model konsorsium. Konsorsium kini mempunyai tantangan untuk membuat model yang tidak menyertakan data apa pun dari anggota yang keluar untuk mengubah model saat ini C_1 menjadi model C_2 yang tidak menyertakan bagian mana pun dari model M_1, namun hanya merupakan hasil penggabungan model M_2 dan model M_3. Masalah ini tidak dapat diatasi dengan algoritma yang telah kita bahas di Bab 3. Jika konsorsium dibangun satu mitra pada satu waktu, situasi dapat muncul dimana masing-masing mitra menyediakan modelnya satu per satu. Hal ini memerlukan

kemampuan untuk menyusun model satu per satu tanpa ada sinkronisasi di antara model tersebut.

7.2 MASALAH KETIDAKCOCOKAN DATA ASINKRON

Pendekatan untuk menangani ketidakcocokan data yang dijelaskan dalam Bab 4 mengasumsikan bahwa semua situs fusi aktif. Hal ini memungkinkan situs untuk bertukar informasi satu sama lain tentang format data yang mereka miliki, dan format terbaik yang digunakan untuk pelatihan model dapat ditentukan. Demikian pula, untuk menskalakan semua fitur masukan dengan benar, properti statistik dari semua fitur dikirim ke server fusi dan metode penskalaan yang konsisten diperoleh. Pemeriksaan kualitas data memerlukan komputasi matriks kebingungan lintas lokasi, dan perkiraan kualitas data berdasarkan kebijakan dapat dilakukan.

Jika situs menyediakan modelnya pada waktu yang berbeda, potensi ketidakcocokan yang dapat terjadi dengan situs yang akan bergabung di masa mendatang tidak diketahui. Akibatnya, rekonsiliasi apa pun antara format, pendekatan normalisasi, atau kualitas hanya dapat dilakukan di antara klien fusi yang hadir pada awal proses fusi. Hal ini mengarah pada penciptaan model umum awal yang mengharapkan masukan diukur berdasarkan konvensi tertentu, dan format masukan tertentu harus diikuti. Model apa pun yang disediakan oleh klien fusi yang bergabung setelah model umum awal harus mematuhi norma-norma yang telah diputuskan oleh kelompok klien fusi pertama yang berpartisipasi.

Penentuan awal fungsi penskalaan berfungsi dengan baik untuk fitur numerik, namun untuk fitur kategorikal, ada kemungkinan bahwa kategori baru akan muncul ketika klien baru bergabung dalam proses fusi di masa mendatang. Pengkodean yang konsisten sebagai nilai biner dapat memberikan perlindungan terbatas untuk nilai baru yang mungkin muncul di masa depan. Situasi yang dapat timbul dengan pengkodean variabel kategori saat anggota baru bergabung ditunjukkan pada Tabel 7.1.

Pada Tabel, diasumsikan bahwa awalnya hanya dua anggota yang bergabung dalam konsorsium, dan untuk fitur kategorikal tertentu, kedua anggota tersebut memiliki nilai seperti yang ditunjukkan pada tabel. Awalnya, ada empat nilai kategorikal di antara dua partisipan yang berbeda. Mereka dapat, dengan menggunakan algoritma yang diuraikan dalam Bagian 4.1.2, menyetujui serangkaian pengkodean nilai yang ditunjukkan pada subtabel kanan atas. Baris bawah tabel menunjukkan kumpulan nilai baru ketika klien fusi baru bergabung dengan konsorsium. Klien fusi ini memiliki nilai baru, Hijau, namun tidak dapat dikodekan ke dalam kumpulan nilai yang sudah ada yang hanya mengizinkan empat kemungkinan nilai untuk fitur tersebut, dan dikodekan sebagai kumpulan yang semuanya nol. Nilai baru Pink lainnya, yang diberikan oleh anggota baru lainnya dikodekan sebagai himpunan semua nol.

Nilai apa pun yang tidak ada dalam kumpulan nilai asli akan dikodekan sebagai nol, dan sistem dapat terus berfungsi tanpa peringatan. Namun, perbedaan di antara nilai-nilai baru tidak lagi dapat dipertahankan. Hijau dan Merah Muda dikodekan dengan cara yang sama dalam pendekatan ini. Ini mungkin bukan situasi yang diinginkan jika diskriminasi antar nilai-

nilai baru juga penting. Sedangkan pengkodean One-Hot menyediakan cara untuk membuat kategori catch-all yang dapat menangkap item baru yang kini ada dalam data yang membentuk informasi asli untuk menangani situasi tersebut. Lebih dari satu nilai kategori tersebut dapat disimpan untuk diperluas di masa mendatang, karena data asli tidak pernah memiliki nilai tersebut, dan seseorang dapat menugaskannya ke nilai baru yang tersedia.

Contoh pengkodean tambahan yang memungkinkan munculnya 4 nilai baru yang tidak diketahui di masa mendatang ditunjukkan pada Tabel 7.2. Hal ini memungkinkan 2 nilai baru dikodekan, menyisakan ruang untuk 2 nilai lagi di masa mendatang. Perluasan pengkodean untuk memiliki lebih banyak nilai menimbulkan beban komputasi tambahan dan perlu diseimbangkan dengan kebutuhan untuk mendukung nilai-nilai baru yang diantisipasi.

Tabel 7.1 Masalah pengkodean kategoris dengan klien fusi baru.

Konsorsium dengan Anggota Awal	
Klien Fusi	Nilai-Nilai yang Berbeda
Klien A	Biru, Merah, Emas
Klien B	Merah, Biru, Kuning

Konsorsium dengan Anggota Baru
Klien Fusi
Klien A
Klien B
Klien C
Klien D

Konsorsium dengan Anggota Awal	
Nilai	**Pengkodean**
Biru	[1 0 0 0]
Merah	[0 1 0 0]
Emas	[0 0 1 0]
Kuning	[0 0 0 1]

Konsorsium dengan Anggota Baru	
Nilai	**Pengkodean**
Biru	[1 0 0 0]
Merah	[0 1 0 0]
Emas	[0 0 1 0]
Kuning	[0 0 0 1]
Hijau	[0 0 0 0]
Merah Jambu	[0 0 0 0]

Tabel 7.2 Masalah pengkodean kategoris dengan klien fusi baru.

Konsorsium dengan Anggota Awal	
Klien Fusi	**Nilai-Nilai yang Berbeda**
Klien A	Biru, Merah, Emas
Klien B	Merah, Biru, Kuning

Konsorsium dengan Anggota Baru
Klien Fusi
Klien A
Klien B
Klien C
Klien D

Konsorsium dengan Anggota Awal	
Nilai	**Pengkodean**
Biru	[1 0 0 0 0 0 0 0]
Merah	[0 1 0 0 0 0 0 0]
Emas	[0 0 1 0 0 0 0 0]
Kuning	[0 0 0 1 0 0 0 0]

Konsorsium dengan Anggota Baru	
Nilai	**Pengkodean**
Biru	[1 0 0 0 0 0 0 0]
Merah	[0 1 0 0 0 0 0 0]
Emas	[0 0 1 0 0 0 0 0]
Kuning	[0 0 0 1 0 0 0 0]
Hijau	[0 0 0 0 1 0 0 0]
Merah Jambu	[0 0 0 0 0 1 0 0]

Saat memeriksa konsistensi nilai, konsep matriks kebingungan lintas situs masih dapat digunakan, kecuali kelas dan label data situs baru dibandingkan dengan model yang ada, dan label yang sudah ada perlu digunakan. Salah satu tantangannya adalah jika situs baru memberikan label yang berbeda, situs tersebut hanya dapat ditangkap sebagai satu kelas baru yang tidak terlihat. Model awal tidak akan memprediksi keluaran apa pun untuk kelas yang tidak terlihat. Namun, situs baru mungkin memiliki data untuk kelas baru. Untuk mengatasi situasi ini, model awal perlu membuat ketentuan agar model asli dapat memprediksi kelas yang tidak terlihat.

Pengklasifikasi standar tidak dapat dengan mudah menangani konsep kelas yang tidak terlihat atau kelas baru. Selain itu, tidak seperti kasus di mana seseorang perlu mendeteksi apakah input yang dilihatnya adalah kelas baru, tantangan untuk fusi awal adalah bahwa beberapa titik yang sesuai dengan kelas baru harus dihasilkan. Salah satu pendekatan untuk

melakukannya adalah dengan meminta setiap situs asli untuk melatih N model biner untuk setiap N label kelas asli. Masing-masing model ini memprediksi mana suatu input termasuk dalam suatu kelas atau tidak. Pendekatan ini memerlukan beberapa poin yang tidak termasuk dalam kelas mana pun. Titik-titik ini dapat diidentifikasi dalam kumpulan data asli karena masing-masing model biner N akan memprediksinya sebagai bukan milik kelasnya. Jika tidak ada titik seperti itu, maka beberapa titik yang tidak ada dalam kelas mana pun perlu dibangun. Augmentasi data asli dengan poin-poin baru ini memberikan kemampuan model asli untuk siap memprediksi kelas baru. Pendekatan berbasis kebijakan untuk menentukan kualitas data dapat digunakan tanpa masalah apa pun di lingkungan asinkron.

7.3 PENDEKATAN BERBASIS ENSEMBLE

Salah satu pendekatan untuk mengatasi masalah sinkronisasi adalah dengan menggunakan kumpulan model, daripada mencoba menggabungkan dan menggabungkan semua model menjadi satu. Konsep ansambel berbasis kebijakan telah dibahas di Bagian 5.3. Menggunakan pendekatan berbasis ansambel untuk pembelajaran gabungan, masing-masing model fusi menyediakan model yang dilatih secara independen ke server fusi. Server fusi akan mengumpulkannya menjadi sebuah ansambel yang akan digunakan untuk membuat prediksi sebagai agregat.

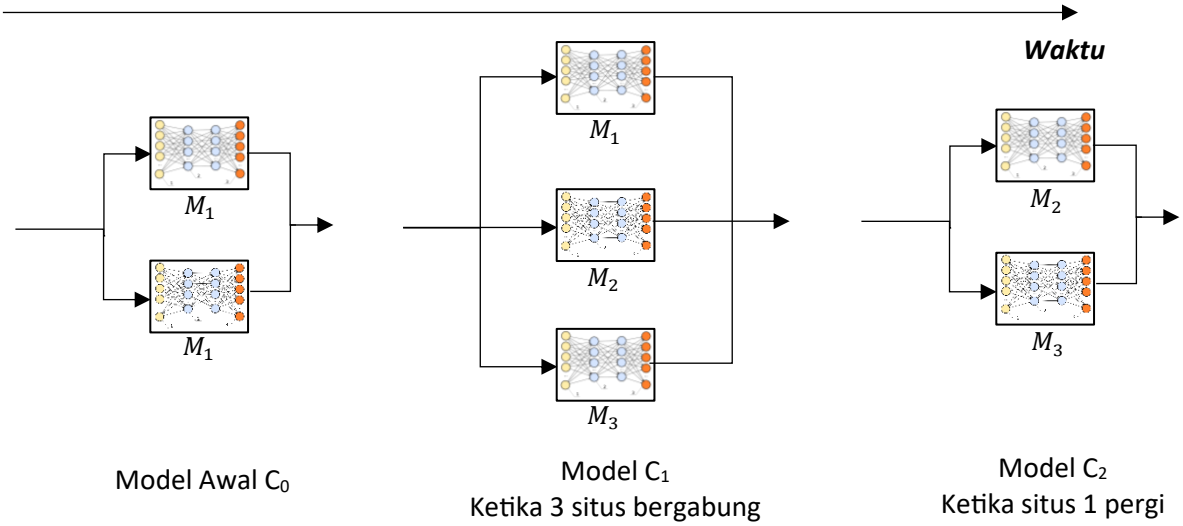
Ensembel yang akan dibuat akan menggunakan pendekatan yang dijelaskan pada Bagian 5.3. Namun, jumlah ansambel yang akan digunakan akan bervariasi dari waktu ke waktu. Untuk kasus spesifik lokasi fusi yang bergabung dan keluar dari konsorsium seperti yang ditunjukkan pada Gambar 7.3, ansambel akan berkembang seperti yang ditunjukkan pada Gambar 7.4. Model yang disediakan oleh masing-masing lokasi akan ditambahkan ke ansambel seiring dengan bergabungnya berbagai lokasi ke dalam konsorsium. Ketika sebuah situs meninggalkan konsorsium dan ingin menghapus model dan datanya, modelnya dapat dihapus dari ansambel. Hal ini memberikan kemampuan untuk dengan mudah menangani perubahan keanggotaan yang terjadi seiring waktu.

Ensembel mempunyai keuntungan yang signifikan dibandingkan pendekatan yang memadukan atau menggabungkan model-model secara bersamaan sehingga lokasi yang berbeda tidak perlu berkoordinasi dan memilih arsitektur model yang umum. Selama nilai masukan untuk masing-masing model diskalakan secara konsisten, dan mereka memprediksi keluaran yang sama, yaitu setiap model dalam ansambel memperkirakan fungsi yang sama, komponen model dapat memiliki arsitektur internal apa pun. Salah satu situs dapat menyediakan pohon keputusan, salah satu situs dapat menyediakan jaringan saraf dengan sepuluh lapisan neuron, dan situs lainnya dapat menyediakan jaringan saraf dengan dua puluh lapisan. Fleksibilitas ini berarti bahwa model dapat dilatih sepenuhnya secara mandiri, dan tidak memerlukan koordinasi apa pun dalam arsitektur model sama sekali.

Pendekatan ansambel akan berhasil dengan baik, namun memiliki masalah yaitu ukuran ansambel akan bertambah seiring semakin banyaknya situs yang bergabung dengan ansambel tersebut. Untuk situs yang informasinya dikumpulkan dari beberapa situs yang digunakan, sebuah ansambel sudah cukup. Namun, jika ada beberapa situs yang

menyumbangkan model fusi, ansambelnya bisa menjadi sangat besar dan berat. Jika ada batasan seberapa besar model yang seharusnya digunakan untuk inferensi, ansambel dengan beberapa komponen bisa menjadi masalah.

Ketika dikombinasikan dengan pendekatan untuk menghasilkan kebijakan untuk memilih model yang berbeda dalam ansambel, seperti dijelaskan dalam Bagian 5.3, hal ini dapat menghasilkan peningkatan yang signifikan dalam kinerja ansambel.



Gambar 7.4 Evolusi model ansambel dengan keanggotaan dinamis.

7.4 KONVERSI KE MODEL BERBASIS ATURAN

Persyaratan dasar untuk melakukan pembelajaran gabungan secara asinkron adalah memiliki kemampuan untuk menggabungkan model AI yang telah dilatih sebelumnya ketika data yang digunakan untuk melatih model AI tidak dapat diakses. Jika kita memiliki kemampuan untuk mendefinisikan operasi penambahan dan penghapusan pada model, maka pelatihan asinkron dapat didukung.

Operator penjumlahan akan mengambil dua model terlatih dan menghasilkan model gabungan yang menangkap jumlah semua pola yang disertakan dalam kedua model. Operasi penjumlahan semantik dapat ditentukan berdasarkan data pelatihan yang digunakan untuk membuat model. Misalkan model M_1 dibuat dengan pelatihan kumpulan data D_1 dan model M_2 dibuat dengan pelatihan pada kumpulan data D_2. Penambahan kedua model M_1 + M_2 adalah model M_3 yang akan menjadi model yang dibuat menggunakan data pelatihan $D_1 \cup D_2$.

Operasi penghapusan hanya dapat didefinisikan pada model yang disusun dari operasi penambahan. Pada contoh di atas, efek penghapusan model M_2 dari model M_3 akan menjadi model M_1, yaitu model yang dilatih dari semua data yang secara unik hanya digunakan untuk melatih model M_2.

Meskipun definisi operasi penambahan dan penghapusan didasarkan pada data pelatihan yang digunakan, operasi penambahan dan penghapusan harus dilakukan tanpa akses ke data pelatihan. Operasi ini sesuai dengan tindakan yang perlu dilakukan ketika anggota yang berbeda bergabung atau keluar dari konsorsium.

Kemampuan untuk melakukan operasi penambahan atau penghapusan pada model bergantung pada sifat model. Untuk model jaringan saraf, operasi penambahan dan penghapusan sulit dilakukan, dan tidak jelas bagaimana melakukannya dalam mode asinkron tanpa akses apa pun ke data pelatihan asli. Namun, untuk model jenis lain, misalnya ketika model merupakan seperangkat aturan pencarian, operasi penambahan dan penghapusan cukup mudah untuk dilakukan.

Salah satu pendekatan untuk melakukan federasi secara asinkron untuk semua jenis model adalah dengan mengubah model tersebut menjadi model setara yang didasarkan pada seperangkat aturan. Seperangkat aturan kemudian dapat dilakukan operasi penambahan dan penghapusan dengan relatif mudah. Untuk lingkungan operasi koalisi, di mana pembelajaran gabungan perlu dilakukan, pendekatan ini telah digunakan untuk mengubah jaringan saraf menjadi aturan yang setara. Konsep perhatian adalah dasar fundamental untuk mendefinisikan aturan yang dapat ditafsirkan. Untuk setiap keluaran yang diprediksi oleh jaringan saraf, mekanisme perhatian mengidentifikasi bagian mana dari ruang masukan yang paling relevan untuk membuat prediksi. Mekanisme aturan yang dapat diinterpretasikan mendefinisikan pemetaan dari perhatian ke prediksi sebagai seperangkat aturan yang menyediakan model yang setara dengan jaringan saraf asli.

Model AI lainnya juga dapat dikonversi ke kumpulan aturan yang setara, misalnya. pohon keputusan dapat diubah menjadi aturan dan aturan tersebut kemudian dapat digabungkan bersama untuk menciptakan mekanisme di mana beberapa pohon keputusan yang dipelajari secara independen dapat digabungkan bersama.

Untuk melakukan operasi penambahan pada kumpulan aturan, server fusi perlu menyimpan kumpulan aturan berbeda yang disediakan bersama. Setiap kumpulan aturan memprediksi keluaran tertentu dalam kombinasi kondisi. Aturan dapat digabungkan, dan konflik di antara aturan tersebut dapat terjadi, ketika aturan yang berbeda memprediksi keluaran yang berbeda, misalnya. dengan memeriksa tumpang tindih antara ruang fitur yang ditentukan oleh mereka. Konflik-konflik tersebut kemudian dapat diselesaikan dengan menggunakan masukan manusia atau aturan-aturan otomatis untuk membuat prioritas di antara rangkaian-rangkaian berbeda yang disediakan oleh berbagai pihak. Hasil akhirnya adalah seperangkat aturan yang setara dengan model yang diberikan sebagai masukan.

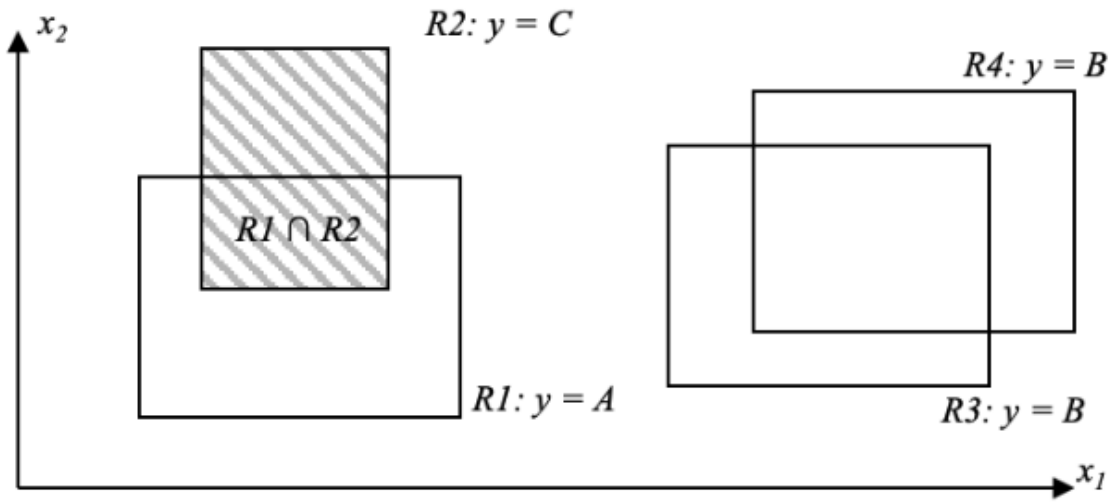
Kami mengilustrasikan proses menggabungkan kumpulan aturan yang berbeda menggunakan turunan dari algoritma yang dirinci dalam. Untuk kumpulan data pelatihan yang terdiri dari beberapa n fitur masukan x_1, x_2, . . . x_n dan keluaran y, setiap aturan mengidentifikasi beberapa area dalam ruang fitur yang ditentukan dengan menandai setiap dimensi dengan salah satu fitur masukan. Demi penyederhanaan, kita akan berasumsi bahwa fitur-fitur yang berbeda sepenuhnya independen satu sama lain. Meskipun asumsi ini secara umum tidak benar, ada banyak pendekatan, seperti analisis komponen utama dan analisis korespondensi berganda, yang dapat digunakan untuk mengubah fitur dependen menjadi serangkaian fitur independen. Oleh karena itu, untuk tujuan penjelasan, asumsi independensi dapat dibuat tanpa kehilangan keumumannya.

Setiap aturan secara efektif mendefinisikan sekumpulan kondisi yang terdiri dari sekumpulan kombinasi fitur masukan, dan keluaran yang sesuai y sesuai dengan kombinasi tersebut. Hal ini dapat dilihat sebagai pendefinisian wilayah geometris dalam ruang fitur, dan pasangan aturan apa pun yang dipertimbangkan mungkin tumpang tindih dalam ruang fiturnya, atau sama sekali tidak tumpang tindih dalam ruang fitur. Jika tidak ada tumpang tindih, mereka tidak saling berkonflik. Jika terdapat tumpang tindih, dan aturan memprediksi keluaran yang sama, maka aturan tersebut tidak bertentangan satu sama lain. Namun, jika terdapat tumpang tindih dan peraturan memperkirakan keluaran yang berbeda, maka terdapat potensi konflik.

Situasinya diilustrasikan pada Gambar 7.5. Empat aturan $R1$ hingga $R4$ ditampilkan dan mencakup wilayah berbeda dalam ruang fitur yang ditentukan oleh dua fitur masukan. Dua fitur masukan memudahkan ilustrasi dalam sebuah bidang meskipun konsepnya dapat dengan mudah digeneralisasikan ke banyak dimensi. Pada gambar, aturan $R1$ dan $R2$ tumpang tindih, begitu pula aturan $R3$ dan $R4$. Namun, tumpang tindih antara aturan $R3$ dan $R4$ tidak menyebabkan konflik karena kedua aturan memprediksi keluaran akan bernilai sama (yaitu B). Di sisi lain, wilayah yang tumpang tindih antara aturan $R1$ dan $R2$ mengalami konflik karena satu aturan memprediksi keluaran di wilayah tersebut adalah A sedangkan aturan lainnya memprediksi keluaran menjadi C.

Pemeriksaan wilayah tumpang tindih antar ruang fitur dapat mengidentifikasi potensi konflik antar aturan, khususnya aturan yang diperoleh dari klien fusi yang berbeda. Konflik-konflik ini kemudian dapat dihilangkan dengan menentukan semacam pedoman penentuan prioritas mengenai aturan mana yang harus dipilih jika terjadi konflik. Algoritme yang ada untuk mempelajari aturan biasanya akan menghasilkan seperangkat aturan yang bebas konflik. Namun, ketika aturan dikumpulkan dari klien yang berbeda, beberapa situasi konflik mungkin muncul.

Penyelesaian konflik dapat didasarkan pada jumlah klien fusi berbeda yang memberikan prediksi yang sama, atau beberapa kriteria lain dengan mempertimbangkan reputasi klien yang memberikan aturan, atau bahkan menggunakan masukan ahli secara manual. Kemudahan resolusi konflik antar kumpulan aturan yang disediakan oleh klien fusi berbeda berarti bahwa tugas menambahkan model relatif sederhana. Server fusi dapat mengambil semua aturan yang disediakan oleh klien fusi yang berbeda, menganalisisnya untuk menemukan konflik, menyelesaikan konflik, dan mendapatkan seperangkat aturan baru yang terkonsolidasi.



Gambar 7.5: Ilustrasi konflik peraturan.

Menghapus kontribusi yang diberikan oleh pihak yang modelnya harus dihapus juga dapat dilakukan dengan mudah jika server fusi melacak semua model yang disediakan oleh masing-masing peserta. Saat peserta keluar, model dari kontributor tersebut dapat dihapus dan model lainnya diubah menjadi kumpulan aturan yang setara dan digabungkan.

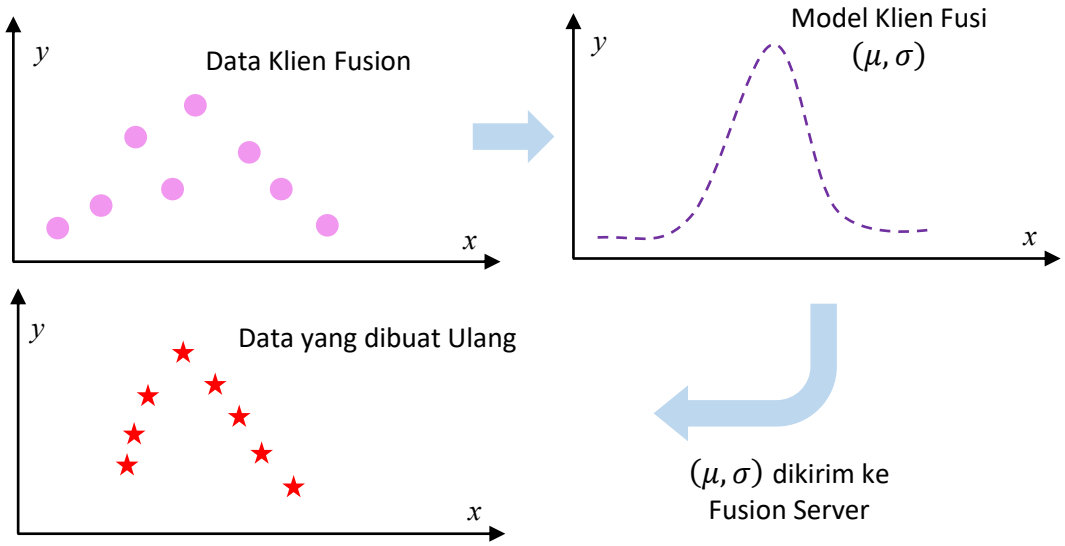
Konversi ke model berbasis aturan mempunyai keuntungan lain. Kumpulan aturan dapat diperiksa secara manual untuk menentukan apakah telah terjadi penyesuaian yang berlebihan pada data pelatihan. Model AI terkadang mempelajari pola yang mungkin hanya merupakan konstruksi dari data yang digunakan untuk melatihnya. Beberapa dari over-fitting tersebut dapat diidentifikasi dengan pemeriksaan aturan secara manual, namun menentukan over-fitting tersebut pada jenis model AI lainnya, misalnya. jaringan saraf atau pohon keputusan mungkin tidak mudah.

7.5 PENGGABUNGAN MODEL BERBASIS PENGHASIL DATA

Model AI mempelajari pola yang ada dalam data pelatihan, yang secara umum kita nyatakan sebagai pembelajaran suatu fungsi. Setelah fungsinya dipelajari, fungsinya dapat digunakan dalam banyak cara. Meskipun fokus utama buku ini adalah penggunaan fungsi tersebut dalam proses bisnis aktual, fungsi yang dipelajari juga dapat digunakan untuk tugas lain, dan salah satu tugas tersebut adalah menghasilkan data baru yang sesuai dengan fungsi yang dipelajari.

Ide dasar dalam fusi model berbasis generator data adalah mempelajari model generator atas data yang ada di situs mana pun. Model generator dapat digunakan untuk membuat ulang titik data di situs server fusi. Setelah model generator dari semua titik data digunakan untuk membuat ulang representasi data yang tersedia di semua lokasi, sistem dapat melatih model baru yang didasarkan pada data yang mewakili semua data di lokasi berbeda.

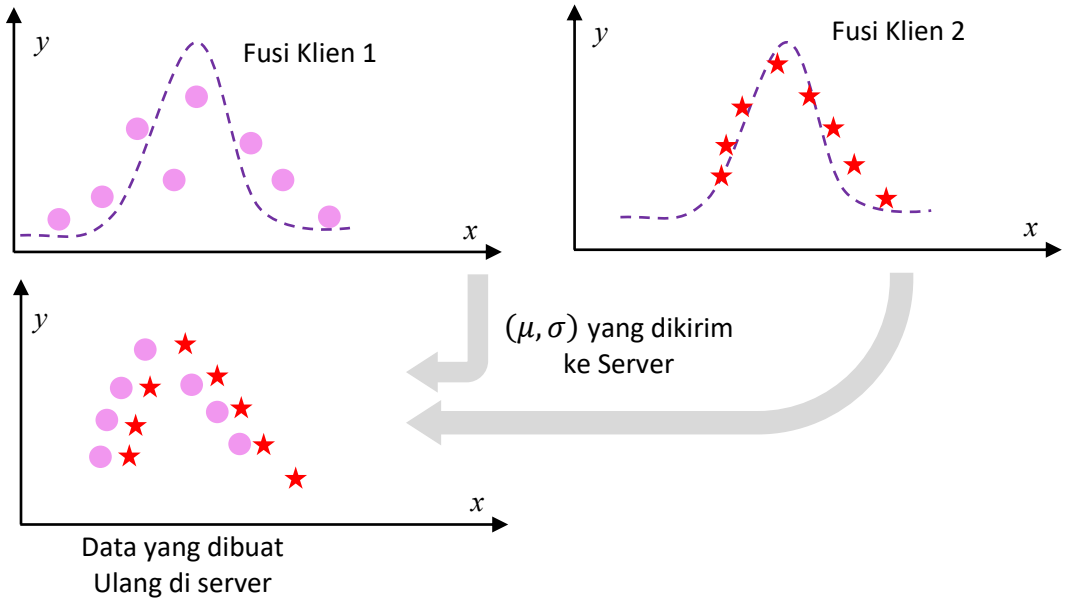
Contoh sederhana ditunjukkan pada Gambar 7.6. Klien fusi memiliki kumpulan data dengan beberapa titik data dengan satu fitur masukan tunggal x dan keluaran y, dan lokasi titik data ditunjukkan pada gambar di kiri atas gambar. Dari domain tersebut, klien fusi mengetahui bahwa hubungan antar input dapat dimodelkan sebagai distribusi Normal yang dicirikan oleh dua parameter, mean μ dan varians σ. Klien fusion akan menghitung dua nilai ini dan mengirimkannya ke server fusion. Server fusi akan menghasilkan sejumlah titik data yang diambil dari distribusi Normal dan membuat ulang data secara lokal. Meskipun datanya tidak persis sama dengan data klien fusi, data tersebut cukup mirip untuk digunakan sebagai proksi untuk data tersebut. Mengkarakterisasi data sebagai distribusi probabilitas menciptakan model generator karena distribusi probabilitas dapat digunakan untuk menghasilkan titik-titik pada data.



Gambar 7.6 Contoh model generator.

Jika ada beberapa klien fusi dan semuanya mengirimkan model generator ke server fusi, server fusi dapat memanggil masing-masing model generator untuk membuat ulang data yang serupa dengan data klien fusi. Data yang sepenuhnya dibuat ulang ini kemudian dapat digunakan untuk menjalankan fungsi apa pun, termasuk tugas melatih model AI. Proses pembuatan ulang data untuk pembelajaran gabungan dari data di dua lokasi berbeda ditunjukkan pada Gambar 7.7.

Meskipun contoh yang ditunjukkan menggunakan model generator hanya untuk distribusi sederhana, ada banyak pendekatan untuk membuat model generator. Pendekatan serupa telah digunakan untuk menghitung pengelompokan terdistribusi dan untuk menghitung komponen utama data terdistribusi yang didistribusikan di beberapa lokasi. Salah satu contoh model generator yang dapat digunakan untuk membuat ulang data kompleks adalah teknologi untuk set inti, yang efektif untuk melatih model pembelajaran mesin.



Gambar 7.7 Rekonstruksi data menggunakan model generator.

Reproduksi data di lokasi berbeda memiliki banyak keuntungan untuk pembelajaran gabungan. Model generator dibuat dan dikirim dari setiap klien fusi data ke server fusi, yang berarti tidak diperlukan koordinasi dalam cara data diskalakan atau dinormalisasi di berbagai situs fusi yang berbeda. Model penghasil data bertindak sebagai mekanisme yang efisien untuk membuat ulang data sesuai kebutuhan, dan model baru dapat dilatih dengan relatif mudah ketika anggota yang sudah ada menarik datanya, atau ketika klien baru bergabung. Semua klien tidak perlu menyetujui arsitektur model umum. Jika data telah dibuat ulang, maka setiap klien fusi dapat diberikan model terlatih menggunakan arsitektur yang diminta atau diinginkan untuk klien fusi.

Efektivitas pendekatan ini dibatasi oleh ketepatan model generator dan kemampuannya untuk menciptakan kembali pola-pola dalam data asli. Karena data asli tidak sama dengan data yang dibuat ulang, ada kemungkinan beberapa pola terlewatkan sementara pola palsu baru dapat dibuat ulang. Terlepas dari keterbatasan ini, kemampuan pelatihan asinkron dengan koordinasi minimal yang disediakan oleh penggunaan model generator berarti bahwa pendekatan ini mungkin menjadi salah satu cara paling efektif untuk pembelajaran gabungan dalam lingkungan bisnis nyata.

7.6 RINGKASAN

Ada banyak situasi bisnis di mana kumpulan klien fusi yang membagikan model mereka dan menggabungkannya berubah seiring waktu. Dalam bab ini, kita melihat beberapa pendekatan yang dapat digunakan untuk menggabungkan model dari klien fusi baru tanpa mengharuskan mereka menyediakan datanya. Dalam situasi ini, penskalaan parameter yang ditetapkan pada awalnya perlu digunakan untuk klien masa depan. Nilai-nilai kategoris perlu dikodekan dengan pandangan ke masa depan.

Teknik ansambel, menggunakan sistem berbasis aturan, dan model berbasis generator data menyediakan teknik untuk membuat model yang dapat dilatih secara asinkron menggunakan teknik pembelajaran gabungan.

BAB 8
MENGATASI PARTISI VERTIKAL DALAM PEMBELAJARAN FEDERASI

Konsep partisi data diperkenalkan pada Bab 5. Salah satu kasus partisi data yang sulit ditangani adalah partisi vertikal. Partisi vertikal terjadi ketika tabel yang mewakili data tidak memiliki semua kolom di setiap klien fusi. Melihat data yang digunakan sebagai contoh pada Tabel 5.1 di Bab 5, kita dapat membuat contoh tentang apa yang mungkin terjadi pada data dalam tabel tersebut jika terjadi partisi vertikal.

Dalam partisi vertikal, setiap record mungkin ada di semua situs, namun tidak semua fitur ada di semua situs. Contoh partisi vertikal ditunjukkan pada Tabel 8.1. Dalam partisi vertikal, masing-masing catatan ada di setiap situs, namun setiap situs kehilangan beberapa kolom. Pembagian data secara vertikal ini bermasalah karena masukan ke setiap model berbeda di setiap lokasi. Akibatnya, model di setiap lokasi didasarkan pada masukan yang berbeda, dan tidak dapat digabungkan.

Dalam praktiknya, tidak semua catatan data akan ada di semua situs, dan data sebenarnya akan dipartisi sebagai campuran partisi horizontal dan partisi vertikal. Oleh karena itu, solusi praktis untuk AI gabungan dalam konteks bisnis perlu menggunakan kombinasi teknologi yang dijelaskan dalam Bab 5 dan bab ini.

Tabel 8.1 Contoh partisi vertikal.

Situs A			
Indeks	**F1**	**F2**	**Output**
1	A	X	L0
2	B	Y	L1
3	C	Z	L1
4	A	X	L2
5	B	X	L0
6	B	Y	L2

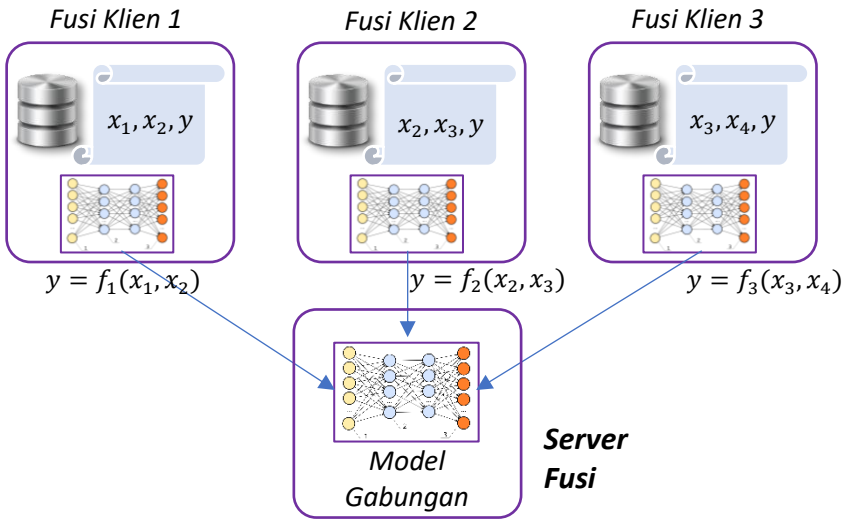
Situs B			
Indeks	**F2**	**F3**	**Output**
1	X	0.3	L0
2	Y	0.2	L1
3	Z	0.3	L1
4	X	0.4	L2
5	X	0.8	L0
6	Y	0.9	L2

Situs C			
Indeks	F1	F2	Output
1	A	X	L0
2	B	Y	L1
3	C	Z	L1
4	A	X	L2
5	B	X	L0
6	B	Y	L2

8.1 PENDEKATAN UMUM PENANGANAN PARTISI VERTIKAL

Ketika data dipartisi secara vertikal, rangkaian fitur yang tersedia di setiap situs berbeda. Skenario yang representatif ditunjukkan pada Gambar 8.1. Tiga situs ditampilkan, masing-masing memiliki data pelatihan unggulan yang terdiri dari dua fitur dan prediksi keluaran. Prediksi keluaran y sama di ketiga situs. Namun, rangkaian fitur berbeda di setiap situs. Saat berlatih secara mandiri, masing-masing dari tiga lokasi akan mempelajari fungsi yang berbeda, salah satunya memprediksi keluaran menggunakan dua masukan berbeda.

Pada contoh yang ditunjukkan pada gambar, situs pertama memiliki data pelatihan yang memungkinkannya memprediksi keluaran y sebagai fungsi dari dua fitur masukan x_1 dan x_2, situs pertama memiliki data pelatihan yang memungkinkannya memprediksi keluaran y sebagai sebuah fungsi dari dua fitur masukan x_2 dan x_3 dan situs 3 memiliki data pelatihan yang memungkinkannya memprediksi keluaran y sebagai fungsi dari dua fitur masukan x_3 dan x_4. Masing-masing dari ketiga situs tersebut mempelajari fungsi yang berbeda, yang berarti bahwa algoritma yang disebutkan dalam Bab 3 tidak dapat digunakan dengan cara yang telah dibahas sejauh ini.



Gambar 8.1 Contoh partisi vertikal.

Perusahaan perlu memutuskan apakah masuk akal untuk menggabungkan ketiga fungsi yang berbeda ini dalam konteks operasinya. Ada beberapa cara berbeda di mana bisnis

mungkin ingin menggunakan hasil yang berasal dari penggabungan data dari semua lokasi berbeda:

1. ***Memperbaiki Model Lokal:*** Sebagai hasil dari penggabungan model-model di lokasi pusat, masing-masing lokasi bisa mendapatkan model yang lebih baik daripada yang bisa mereka latih secara lokal. Jika parameter input x_3 dan x_4 memberikan informasi tambahan yang dapat membuat model yang digunakan oleh situs pertama $y = f_1(x_1, x_2)$ menjadi lebih baik, mungkin berguna untuk mencoba mengekstrak model dari situs lain. Dalam pengaturan ini, situs pusat dapat mempelajari fungsi yang lebih luas $y = f x_1, x_2, x_3, x_4$, dan kemudian, bergantung pada asumsi yang dibuat mengenai nilai x_3 dan x_4 di situs 1, dapat menghasilkan hasil yang berbeda dan lebih baik. ramalan. Sebagai contoh, sebuah bank yang mengumpulkan informasi berbeda tentang transaksi perbankan di berbagai negara dapat menggabungkan model berbeda dari banyak negara dengan menggunakan teknik seperti augmented feature mapper (dijelaskan nanti di Bagian 8.4).
2. ***Ubah Model Lokal:*** Jika, sebagai hasil penggabungan model, situs fusi menemukan bahwa kumpulan salah satu nilai masukan, misalnya. x_3 sangat signifikan dalam prediksi keluaran y, masing-masing lokasi dapat memilih untuk mengubah prosedur dan kebijakan pengumpulan datanya sehingga mereka juga mengumpulkan masukan x_3.
3. ***Menggabungkan Wawasan:*** Jika entitas yang sama diamati dengan atribut berbeda oleh dua situs berbeda, maka kedua situs tersebut dapat menggabungkan wawasannya untuk mempelajari lebih lanjut tentang entitas tersebut. Dalam pengaturan ini, situs pusat mempelajari fungsi yang lebih luas $y = f(x_1, x_2, x_3, x_4)$ dan membantu dalam inferensi yang diperlukan entitas dengan mengumpulkan informasi dari beberapa situs. Sebagai contoh, jika bank dan operator telepon memutuskan untuk bekerja sama untuk mencegah penipuan dan penipuan yang dilakukan pada pelanggannya, mereka dapat menggabungkan pengetahuan mereka untuk tujuan ini. Bank mendeteksi informasi yang berbeda dengan operator telepon, namun menggabungkan model mereka dapat membantu mereka melakukan pekerjaan pencegahan penipuan yang lebih baik. Jenis perbaikan serupa dapat diperoleh dalam lingkungan di mana entitas yang sama diamati dengan modalitas penginderaan yang berbeda oleh pengamat yang berbeda, dan masing-masing pengamat melaporkan serangkaian fitur yang berbeda.

Saat model lokal disempurnakan, penggunaan model AI pada tahap inferensi adalah pemanggilan fungsi berdasarkan fitur yang tersedia di setiap situs. Fungsi lebih luas yang dipelajari perlu disesuaikan sehingga setiap situs lokal dapat menggunakannya hanya dengan menggunakan fitur-fitur yang ada secara lokal. Ketika wawasan digabungkan, fungsi yang lebih luas dipelajari tetapi mungkin ada masalah dengan berbagi fitur individual suatu entitas selama fase inferensi. Skenario pastinya menentukan batasan yang dikenakan pada jenis informasi yang dapat dibagikan oleh setiap entitas, dan kita akan mengeksplorasi skenario tersebut lebih lanjut di Bab 9.

Jika pembelajaran gabungan dilakukan dalam konteks di mana terdapat cukup banyak situs penghasil data (misalnya 10 atau lebih), kemungkinan besar banyak situs tersebut memiliki serangkaian fitur yang sama. Seseorang kemudian dapat mengidentifikasi kelompok situs pembuatan data yang menggunakan fitur masukan yang sama. Hal ini memungkinkan pembangunan model gabungan yang biasa terjadi hanya dalam kelompok individu. Namun, ada beberapa situasi ketika kelompok tersebut tidak terlihat, atau mungkin terdapat beberapa kelompok dengan hanya satu situs anggota. Dalam kasus tersebut, mempartisi semua lokasi ke dalam kelompok terpisah di mana pembelajaran gabungan dapat dilakukan mungkin tidak dapat dilakukan, dan skema lain yang dijelaskan dalam bab ini perlu digunakan. Di sisa bab ini, kita akan melihat berbagai cara untuk menangani situasi partisi vertikal.

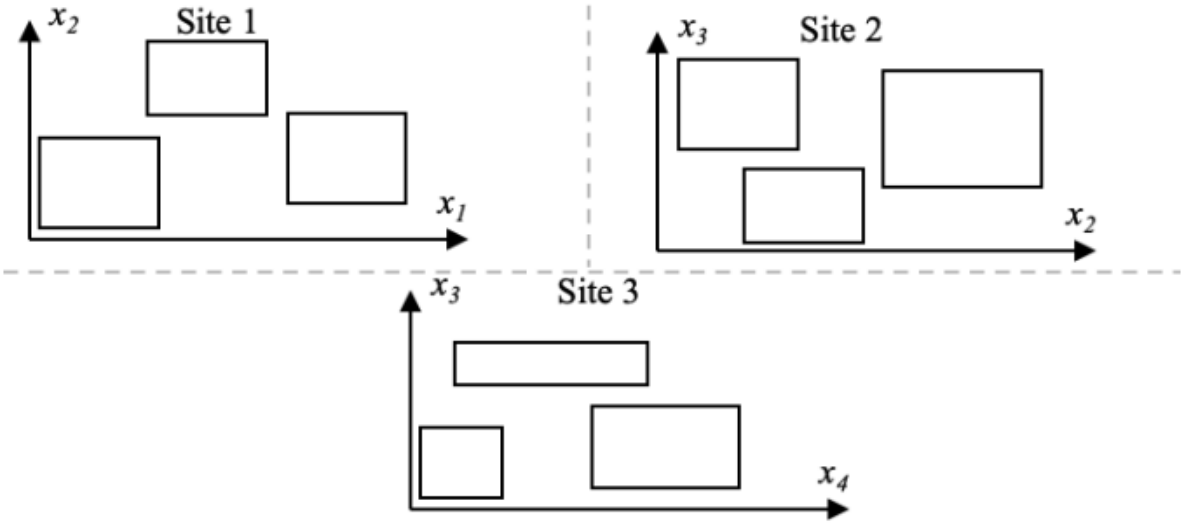
8.2 PENDEKATAN BERBASIS ATURAN

Aturan pengambilan keputusan merupakan bidang pembelajaran mesin tradisional yang sudah ada sejak tahun 1970an. Keuntungan terbesar dari aturan keputusan adalah mudah dipahami, dijelaskan, dan digabungkan. Kemampuan ini memungkinkannya digunakan untuk pembelajaran gabungan asinkron, seperti yang dibahas di Bagian 7.4 Bab 7.

Ketika model yang disediakan oleh klien berbeda menggunakan masukan berbeda, aturan akan menggunakan rangkaian masukan berbeda untuk membuat prediksi. Untuk contoh spesifik yang ditunjukkan pada Gambar 8.1, kumpulan aturan dari situs 1 akan terdiri dari berbagai aturan yang kondisinya ditentukan oleh kombinasi dua fitur masukan x_1 dan x_2 dan bersifat agnostik terhadap nilai s dari x_3 atau x_4. Demikian pula, situs 2 akan memiliki kumpulan aturan yang bergantung pada nilai f pada x_2 dan x_3 dan mengabaikan nilai x_1 dan x_4, sedangkan situs 3 memiliki kumpulan aturan sebagai kombinasi dua fitur masukan x_3 dan x_4 dan tidak bergantung pada nilai apa pun yang ditentukan oleh x_1 atau x_2. Proyeksi kumpulan aturan dari lokasi berbeda pada dua dimensi yang relevan diilustrasikan pada Gambar 8.2. Peraturan yang diberikan oleh masing-masing lokasi akan bebas dari konflik internal, sehingga wilayah yang dipetakan oleh masing-masing peraturan tidak tumpang tindih satu sama lain.

Karena ada empat fitur masukan, nilai maksimum dan minimum dari setiap fitur masukan oleh situs mana pun dapat dihitung. Rentang untuk x_1 hanya akan dihitung berdasarkan aturan yang disediakan oleh situs 1, sedangkan batasan untuk x_2 akan disediakan oleh situs 1 dan 2. Demikian pula, batasan untuk x3 akan disediakan oleh situs 2 dan 3, dan batasan untuk x_4 hanya disediakan oleh situs 3.

Kumpulan aturan yang disediakan oleh masing-masing situs kini dapat diperluas untuk mencakup keempat fitur masukan, dengan asumsi bahwa nilai tersebut valid untuk semua nilai fitur yang hilang. Sebagai contoh, setiap aturan dari situs 1, yang hanya terdiri dari dua fitur tertentu, diasumsikan valid untuk semua nilai x_3 dan x_4, dengan ekstensi analog untuk situs lainnya. Hasilnya, kami memiliki seperangkat aturan yang ditentukan untuk keempat fitur masukan dari masing-masing situs.



Gambar 8.2 Kumpulan aturan dengan partisi vertikal.

Rangkaian peraturan ini kemudian dapat digabungkan dan setiap konflik yang terjadi di antara berbagai peraturan dan bidang-bidang yang tumpang tindih dapat diidentifikasi dan diselesaikan. Hasil akhirnya adalah seperangkat aturan konsisten yang didefinisikan dalam keempat fitur masukan. Untuk setiap lokasi, seseorang dapat mengambil proyeksi dari rangkaian aturan yang dihasilkan hanya pada fitur-fitur yang tersedia di situs tersebut, dan rangkaian aturan terkait (sekali lagi setelah menyelesaikan konflik apa pun) diberikan kembali ke masing-masing situs sebagai tambahan pada model yang telah mereka berikan. Jika wawasan perlu digabungkan, model global dengan semua fitur masukan dapat digunakan secara terpusat atau model terdistribusi untuk menyediakan kombinasi ini.

Pendekatan alternatifnya adalah dengan melakukan pelatihan gabungan aturan di seluruh kumpulan data menggunakan algoritma penambangan aturan asosiasi terdistribusi. Hal ini menghasilkan seperangkat aturan yang merupakan kombinasi dari semua fitur yang tersedia di situs mana pun. Setiap situs kemudian dapat menggunakan subset yang hanya menggunakan fitur yang tersedia secara lokal.

8.3 PENDEKATAN PREDIKSI FITUR

Tantangan utama dalam menggunakan aturan untuk menggabungkan wawasan dari situs yang berbeda adalah bahwa aturan tersebut berasumsi bahwa fitur yang berbeda bersifat independen dan, dengan demikian, memperluas fitur yang hilang dalam aturan agar valid untuk semua kombinasi fitur adalah valid. Meskipun teknik untuk mengubah fitur menjadi komponen independen telah diketahui, mungkin ada beberapa kumpulan data yang konversinya mungkin tidak cukup untuk menangkap pola data dengan benar. Dalam kasus tersebut, skema untuk dapat memprediksi fitur yang hilang di setiap situs mungkin dapat berjalan dengan baik.

Mengingat tugas mempelajari model global sebagai fungsi yang didasarkan pada gabungan semua fitur yang tersedia di seluruh situs, kami menggunakan pendekatan

pembelajaran gabungan yang berulang di seluruh situs untuk membangun model untuk fitur yang berbeda. Proses ini terdiri dari tiga langkah:

1. *Tentukan urutan prediksi fitur:* Ini akan menghasilkan urutan fitur yang hilang di situs mana pun yang harus diprediksi.
2. *Gunakan Pembelajaran Gabungan Naif untuk membuat model prediktor fitur*: Hal ini menghasilkan serangkaian model yang dapat digunakan setiap situs untuk menambah datanya agar mencakup semua fitur.
3. *Perluas dan Gabungkan*: Pelajari model gabungan untuk memprediksi keluaran di semua fitur di semua data.

Agar pendekatan ini berhasil, setidaknya satu fitur harus umum di semua situs. Fitur yang hilang dari sejumlah situs terkecil dapat dipilih menjadi fitur pertama yang diprediksi. Kemudian model gabungan yang dapat digunakan untuk memprediksi fitur yang hilang dari fitur umum di seluruh situs dilatih. Model gabungan ini sekarang dapat digunakan untuk memprediksi fitur ini sehingga jumlah fitur umum di seluruh situs bertambah. Kemudian fitur selanjutnya yang akan diprediksi dapat dipilih. Setelah semua fitur diprediksi, mekanisme pembelajaran gabungan untuk memprediksi keluaran dari semua fitur dapat dipelajari. Tentu saja, urutan prediksi fitur lainnya, misalnya berdasarkan ukuran relatif fitur yang hilang, atau metrik lainnya, juga dapat digunakan dalam proses ini.

Sebagai contoh, misalkan ada empat situs dengan konfigurasi berikut:

- Situs A memiliki fitur x_1, x_2, x_3, x_4 dan keluaran y
- Situs B memiliki fitur x_1, x_3, x_4, x_5 dan keluaran y
- Situs C memiliki fitur x_1, x_2, x_4, x_5 dan keluaran y
- Situs D memiliki fitur x_1, x_4, x_5 dan keluaran y

Dalam kasus khusus ini, fitur x_1 dan x_4 bersifat umum di keempat situs. Fitur x_2 hilang di dua situs, fitur x_3 hilang di dua situs, dan fitur x_5 hilang di satu situs. Salah satu kemungkinan urutan hilangnya pembuatan fitur adalah dengan memprediksi fitur $x5$ terlebih dahulu, diikuti dengan prediksi fitur x_2, dan terakhir prediksi fitur x_3.

Untuk membuat prediksi ini, keempat lokasi pada awalnya akan membuat model gabungan M_5 yang akan mengambil masukan x_1 dan x_4 dan memprediksi keluaran x_5. Dengan kata lain, model gabungan akan dibuat dengan menggunakan algoritma pembelajaran gabungan Naif untuk mempelajari fungsi f_5 sehingga $x_5 = f_5(x_1, x_4)$. Model M_5 dapat dilatih dengan menggunakan pembelajaran gabungan antara situs B, C, dan D. Model tersebut kemudian digunakan oleh situs A untuk menggunakan input x_1 dan x_2 guna menghasilkan nilai prediksi x_5 di A.

Sebagai hasil dari generasi ini, keempat situs tersebut kini memiliki konfigurasi berikut:

- Situs A memiliki fitur x_1, x_2, x_3, x_4, x_5 (dihasilkan menggunakan M_5) dan keluaran y
- Situs B memiliki fitur x_1, x_3, x_4, x_5 dan keluaran y
- Situs C memiliki fitur x_1, x_2, x_4, x_5 dan keluaran y
- Situs D memiliki fitur x_1, x_4, x_5 dan keluaran y

Kini, fitur fitur x_1, x_4, dan x_5 umum ditemukan di keempat situs, dengan fitur x_3 dan x_3 hilang di masing-masing dua situs. Proses pembelajaran gabungan kini dapat diulangi untuk

mempelajari fungsi f_2 yang dapat memprediksi x_2 sebagai suatu fungsi $x_2 = f_2(x_1, x_4, x_5)$. Model M_2 ini dapat dilatih menggunakan pembelajaran gabungan antara situs A dan C. Model yang dihasilkan M_2 dibagikan dengan situs B dan D, dan mereka kini dapat membuat estimasi untuk fitur x_2 secara lokal menggunakan model M_2.

Sebagai hasil dari generasi ini, keempat situs tersebut kini memiliki konfigurasi berikut:

- Situs A memiliki fitur x_1, x_2, x_3, x_4, x_5 (dihasilkan menggunakan M_5) dan keluaran y
- Situs B memiliki fitur x_1, x_2 (dihasilkan menggunakan M_2), x_3, x_4, x_5 dan keluaran y
- Situs C memiliki fitur x_1, x_2, x_4, x_5 dan keluaran y
- Situs D memiliki fitur x_1, x_2 (dihasilkan menggunakan M_2), x_4, x_5 dan keluaran y

Kini, fitur fitur x_1, x_2, x_4, dan x_5 umum ditemukan di keempat situs, dengan fitur x_3 hilang dari situs C dan D. Proses pembelajaran gabungan kini dapat diulangi untuk mempelajari fungsi f_3 yang dapat memprediksi x_3 sebagai suatu fungsi $x_3 = f_2(x_1, x_2, x_4, x_5)$. Model M_3 ini dapat dilatih menggunakan pembelajaran gabungan antara situs A dan B. Model yang dihasilkan M_3 dibagikan dengan situs C dan D, dan mereka sekarang dapat membuat estimasi untuk fitur x_3 secara lokal menggunakan model tersebut M_3.

Sebagai hasil dari generasi ini, keempat situs tersebut kini memiliki konfigurasi berikut:

- Situs A memiliki fitur x_1, x_2, x_3, x_4, x_5 (dihasilkan menggunakan M_5) dan keluaran y
- Situs B memiliki fitur x_1, x_2 (dihasilkan menggunakan M_2), x_3, x_4, x_5 dan keluaran y
- Situs C memiliki fitur x_1, x_2, x_3 (dihasilkan menggunakan M_3), x_4, x_5 dan keluaran y
- Situs D memiliki fitur x_1, x_2 (dihasilkan menggunakan M_2), x_3 (dihasilkan menggunakan M_3, x_4, x_5 dan keluaran y

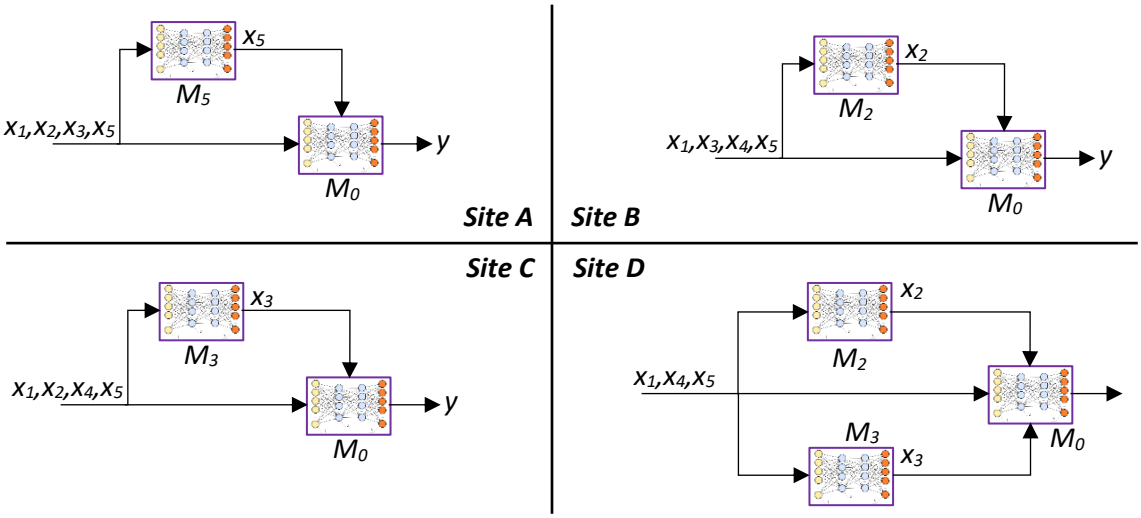
Sekarang, masing-masing situs memiliki semua fitur, dan putaran terakhir pembelajaran gabungan dapat digunakan untuk mempelajari model Mo yang dapat memprediksi keluaran y menggunakan masukan. Model akhir ini tersedia untuk semua situs.

Ketika situs yang berbeda harus membuat prediksi, masing-masing situs harus beroperasi dengan cara yang sedikit berbeda sebagai berikut:

- Situs A akan dimulai dengan nilai x_1, x_2, x_3, x_4, gunakan model M_5 untuk memprediksi x_5 dan gunakan model Mo untuk memprediksi keluaran y
- Situs B akan dimulai dengan nilai x_1, x_3, x_4, x_5, gunakan model M_2 untuk memprediksi x_2 dan gunakan model Mo untuk memprediksi keluaran y
- Situs C akan dimulai dengan nilai x_1, x_2, x_4, x_5, gunakan model M_3 untuk memprediksi x_3 dan gunakan model Mo untuk memprediksi keluaran y
- Situs D akan dimulai dengan nilai x_1, x_4, x_5, gunakan model M_2 untuk memprediksi x_2, gunakan model M_3 untuk memprediksi x_3, dan terakhir gunakan model Mo untuk memprediksi keluaran y

Daripada menggunakan satu model untuk prediksi, masing-masing lokasi kini memiliki pendekatan prediksi yang berbeda, yang didasarkan pada penggabungan model-model yang berbeda secara bersamaan. Pendekatan yang dihasilkan ditunjukkan pada Gambar 8.3. Masing-masing situs telah menggunakan pembelajaran gabungan Naif untuk membantu satu sama lain dalam melatih model perantara untuk memprediksi fitur, dan setelah semua fitur diprediksi, nilai prediksi digunakan untuk memperkirakan keluaran.

Keefektifan pendekatan ini bergantung pada kemampuan untuk memprediksi setiap fitur yang hilang dengan benar dari fitur yang tersedia di situs. Hal ini bergantung pada data dan domain aplikasi.



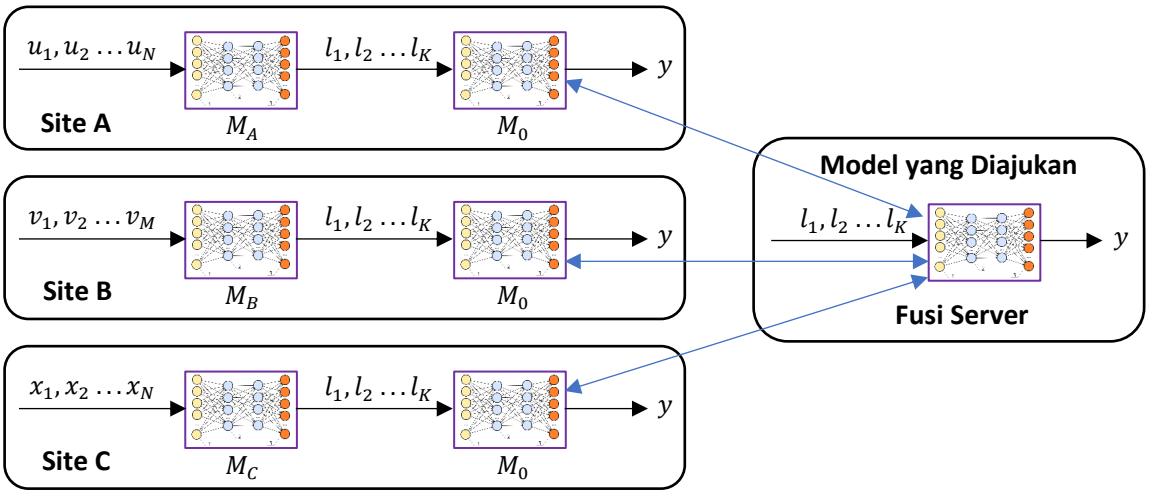
Gambar 8.3 Contoh dengan pendekatan prediksi fitur.

Varian lain dengan pendekatan berbeda untuk mengisi fitur yang hilang juga dapat digunakan, seperti menggunakan pemilihan fitur yang hilang secara acak di setiap lokasi atau menggunakan algoritma pembelajaran semi-supervisi untuk memprediksi fitur yang hilang.

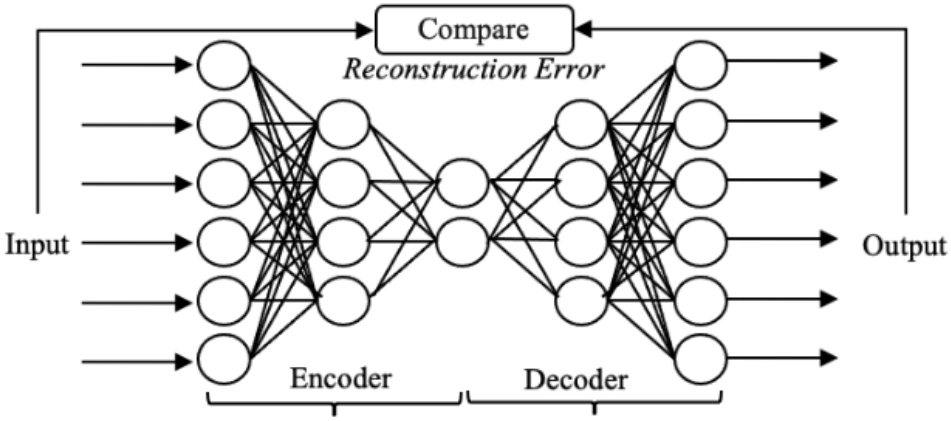
8.4 AUGMENTASI PEMETA FITUR

Konsep penggunaan model AI untuk memprediksi fitur yang hilang dan menggunakan beberapa model untuk menggeneralisasi masukan ke model umum dapat dipandang sebagai kasus khusus harmonisasi model, yaitu ketika setiap situs mempelajari aspek yang berbeda untuk memecahkan masalah yang sama, dan mereka dapat memperoleh manfaat dari berbagi wawasan mereka satu sama lain. Model umum akan dinyatakan dalam ruang laten, yaitu dalam kumpulan fitur yang tersembunyi atau laten, dan memberikan keluaran yang diinginkan. Kumpulan fitur berbeda yang tersedia di masing-masing situs mungkin berbeda tetapi dapat dipetakan ke ruang laten.

Diekspresikan dalam model estimasi fungsi, kita berasumsi bahwa terdapat sekumpulan K fitur laten $l_1, l_2 \ldots l_K$ dan fungsi umum sedang dipelajari di masing-masing situs, yaitu $y = f(l_1, l_2 \ldots l_K)$. Setiap situs juga memiliki serangkaian fitur yang berbeda, mis. situs A mungkin memiliki serangkaian N fitur $u_1, u_2, \ldots u_N$, situs B mungkin memiliki serangkaian M fitur $v_1, v_2, \ldots v_M$, situs C mungkin memiliki serangkaian P fitur $x_1, x_2, \ldots x_P$ dan seterusnya. Setiap situs akan memiliki model AI lokal yang memetakan fitur masukannya ke kumpulan fitur laten, dan akan ada model AI umum yang memetakan fitur laten ke keluaran umum y. Tantangannya adalah setiap situs melatih model sehingga model inti dilatih secara gabungan sementara masing-masing situs melatih pemeta fiturnya sendiri secara independen. Namun, setiap situs hanya memiliki data yang memprediksi keluarannya y sebagai fungsi dari fitur yang dikumpulkan secara lokal, sehingga parameter latennya tidak diketahui.



Gambar 8.4 Contoh dengan pendekatan feature mapper.

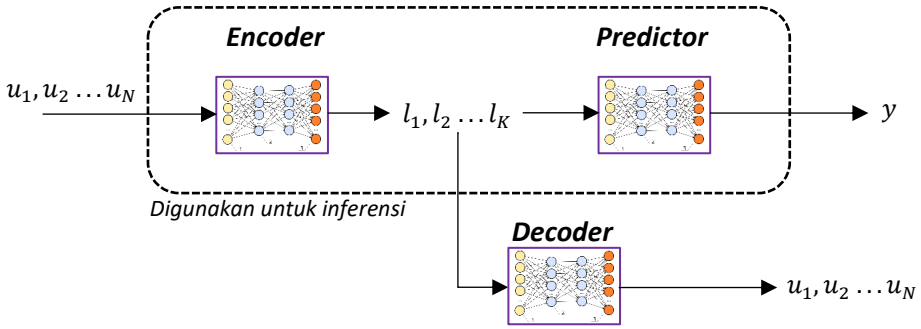


Gambar 8.5 Contoh auto encoder.

Situasinya ditunjukkan pada Gambar 8.4. Tiga situs ditampilkan dengan fitur masukannya dan pendekatan yang ingin mereka ikuti dengan memetakan fitur diikuti dengan model bersama untuk memprediksi keluaran. Model bersama Mo adalah satu-satunya bagian yang dapat difederasi karena menggunakan sekumpulan masukan dan keluaran yang sama. Pemeta fitur M_A, M_B, dan M_C sangat spesifik untuk situs yang memiliki data lokal, dan harus dilatih secara mandiri.

Salah satu cara untuk mengatasi masalah ini adalah dengan menggabungkan jaringan saraf yang dapat mengekstrak fitur ruang laten dengan jaringan saraf yang melakukan prediksi menggunakan fitur ruang laten. Metode umum untuk mengekstraksi fitur adalah penggunaan pembuat enkode otomatis. Auto-encoder adalah jaringan saraf yang terdiri dari dua bagian, bagian encoder yang memetakan masukan asli ke sekumpulan fitur ruang laten, dan decoder yang memetakan fitur ruang laten kembali ke aslinya. Perbandingan masukan yang direkonstruksi dapat digunakan sebagai ukuran kebaikan untuk menentukan kapan pembuat enkode otomatis telah dilatih sepenuhnya. Contoh auto-encoder ditunjukkan pada Gambar 8.5.

Pembuat enkode otomatis kemudian dapat digabungkan dengan jaringan saraf untuk melatih jaringan yang melakukan atau memprediksi keluaran y. Dua parameter, yaitu jumlah fitur laten yang akan digunakan, dan bobot relatif antara keakuratan proses ekstraksi fitur dan keakuratan prediktor, ditentukan terlebih dahulu. Sistem sekarang dapat dilatih oleh masing-masing anggota dengan cara di mana prediktor dilatih menggunakan pendekatan gabungan, sedangkan pembuat enkode otomatis dilatih secara lokal, dengan proses pembelajaran yang secara bersama-sama mengoptimalkan kedua bagian jaringan saraf.



Gambar 8.6 Model di setiap lokasi untuk mempelajari pemetaan fitur.

Setelah jaringan saraf dilatih, hanya bagian encoder yang digunakan dengan prediktor selama proses inferensi untuk setiap situs lokal. Hal ini memberikan masing-masing model yang sesuai dengan masukan mereka. Tugas memetakan fitur ke representasi umum dipelajari oleh auto-encoder. Model dari perspektif suatu lokasi ditunjukkan pada Gambar 8.6. Tiga jaringan saraf, encoder, decoder dan prediktor digabungkan dengan cara seperti yang ditunjukkan pada gambar. Selama fase pelatihan, keluaran dari prediktor dan decoder digunakan untuk menentukan bobot neuron di jaringan saraf. Selama fase inferensi, hanya dua jaringan, encoder dan prediktor, yang digunakan, dan ditampilkan dalam persegi panjang putus-putus di dalam gambar.

8.5 INFERENSI TERFEDERASI

Federasi dan kolaborasi antar situs yang berbeda tidak perlu dilakukan hanya pada tahap pembelajaran model, tetapi juga dapat digunakan pada tahap inferensi. Lagi pula, alasan pelatihan dan pembuatan model AI adalah untuk menggunakannya selama tahap inferensi. Ketika fitur masukan dipartisi, penggunaan federasi selama tahap inferensi mungkin lebih efektif daripada mencoba membuat model bersama.

Kasus penggunaan inferensi gabungan yang sederhana (walaupun mungkin sedikit futuristik pada saat penulisan buku ini) adalah ketika mobil tanpa pengemudi mendekati persimpangan. Rambu di persimpangan mungkin tersumbat sebagian, misalnya. dari dahan pohon yang tumbuh terlalu besar, atau karena rambu tersebut mungkin telah bengkok sehingga kamera mobil tidak dapat menangkap sudut yang baik pada rambu tersebut. Jika ada mobil lain di dekatnya yang dapat melihat objek dengan lebih baik, mobil-mobil tersebut dapat berkomunikasi satu sama lain untuk bertukar perspektif tentang apa yang dimaksud dengan

tanda tersebut, dan masing-masing mobil dapat lebih percaya diri dalam mengambil keputusan.

Ketika entitas yang sama diamati oleh dua lokasi berbeda yang mengumpulkan jenis informasi berbeda tentang entitas tersebut, model umum tidak dapat dibangun di kedua lokasi karena masukannya berbeda. Namun, jika kedua situs berkolaborasi bersama, mereka mungkin masih dapat berbagi output dari model mereka tentang entitas selama tahap inferensi, dan berbagi hasil inferensi akan membantu masing-masing situs dalam mengambil keputusan. Ini adalah penggunaan Federasi selama tahap inferensi dibandingkan dengan tahap pembuatan model.

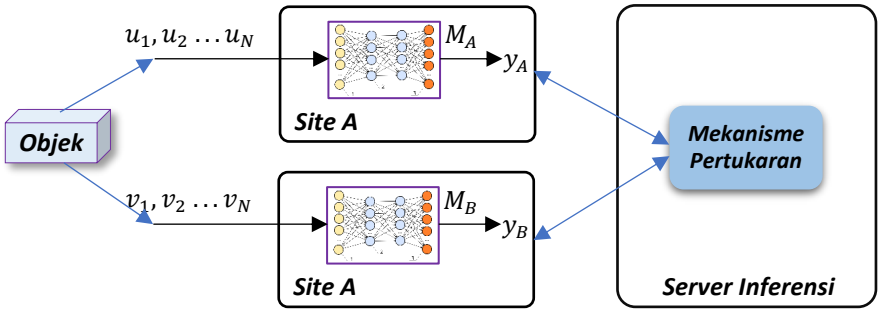
Dalam skenario bisnis yang lebih realistis saat ini, penjahat menargetkan warga sipil yang tidak bersalah untuk mencoba mendapatkan akses terhadap uang mereka dengan melakukan panggilan telepon dan mengirimkan email spam kepada mereka. Meskipun penipuan pada akhirnya terjadi ketika korban mentransfer uang dari rekening mereka, yang melibatkan bank atau perusahaan keuangan lain, prosedur ini juga melibatkan interaksi penjahat dengan perusahaan telekomunikasi atau Penyedia Layanan Internet. Karena entitas yang sama diamati oleh beberapa perusahaan, mereka dapat berbagi informasi tentang tersangka kriminal satu sama lain untuk melakukan tugas yang lebih baik dalam pencegahan penipuan. Meskipun pembagian informasi pribadi secara normal di antara penyedia layanan seperti bank dan penyedia layanan telepon mungkin dilarang secara umum berdasarkan peraturan privasi seperti GDPR, hal ini diperbolehkan untuk tujuan yang ditargetkan secara sempit seperti pencegahan penipuan.

Kasus penggunaan praktis lainnya adalah fusi informasi sensor multi-modal. Ketika objek yang sama diamati melalui beberapa jenis sensor yang berbeda, masing-masing sensor memberikan indikasi tentang apa yang dilihatnya. Ini dapat digunakan untuk mengidentifikasi jenis objek yang dilihat melalui modalitas yang berbeda, misalnya. menggunakan suara dan penglihatan untuk mengidentifikasi hewan yang diamati di alam liar, atau menggunakan suara dan penglihatan untuk menentukan apakah mesin di lantai pabrik beroperasi secara normal, atau mengalami situasi tidak normal. Model AI yang akan digunakan untuk setiap modalitas penginderaan mungkin berbeda, namun hasilnya dapat digabungkan.

Inferensi gabungan dapat dilihat sebagai implementasi dari ansambel model AI terdistribusi di mana setiap model dalam ansambel tersebut dilatih sepenuhnya. Hasil dari model ansambel yang berbeda perlu digabungkan bersama. Penggabungan tersebut dapat dilakukan dengan menggunakan suara mayoritas, atau menggunakan variasi teknik berbasis kebijakan, seperti dijelaskan dalam Bagian 5.3 pada Bab 5.

Proses inferensi gabungan antar lokasi yang berbeda biasanya terjadi seperti yang ditunjukkan pada Gambar 8.7. Beberapa situs mengamati suatu objek dan masing-masing mengekstrak serangkaian fitur berbeda dari objek tersebut. Dalam contoh spesifik yang ditampilkan, fitur $u_1, u_2, \ldots u_N$ diekstraksi oleh situs A, dan fitur $v_1, v_2, \ldots v_M$ diekstraksi dari situs B. Mungkin ada situs lain yang juga mengekstrak fiturnya sendiri. Masing-masing situs memiliki modelnya sendiri yang telah dilatih sebelumnya untuk menggunakan fitur yang mereka ekstrak. Situs A mempunyai keputusan $y(A)$ sedangkan Situs B mempunyai keputusan

$y(B)$. Semua situs bertukar keputusan satu sama lain menggunakan mekanisme pertukaran tertentu. Mekanisme pertukaran yang paling mudah adalah penggunaan server untuk bertukar informasi, namun ada beberapa mekanisme lain untuk pertukaran tersebut yang tidak memerlukan server. Setelah masing-masing lokasi mempunyai informasi mengenai semua keputusan lainnya, maka situs tersebut dapat mengambil keputusan berdasarkan suara mayoritas atau mekanisme resolusi lain untuk menentukan keputusan akhir mana yang akan digunakan.



Gambar 8.7 Proses inferensi gabungan.

Ketika model yang berbeda digunakan, metadata tentang model yang berbeda dapat digunakan untuk menentukan seberapa besar kepercayaan yang diberikan terhadap setiap kesimpulan yang diberikan oleh model tersebut. Mekanisme pembobotan yang dipinjam dari konsep ansambel berbasis kebijakan dapat dilakukan dengan melihat jarak titik inferensi dari wilayah dalam ruang fitur tempat model dilatih. Setiap situs perlu menyediakan informasi tersebut selain kesimpulannya. Metrik lain yang dapat disediakan mencakup keyakinan terhadap prediksi masing-masing model.

Inferensi gabungan memberikan kemampuan untuk menggabungkan wawasan dari banyak lokasi berbeda untuk membuat keputusan yang lebih baik.

8.6 RINGKASAN

Partisi data Vertikal muncul dalam banyak konteks bisnis. Ini adalah situasi di mana situs yang berbeda telah mengumpulkan informasi untuk memprediksi fungsi umum yang akan digunakan dalam konteks bisnis mereka, namun rangkaian masukan fitur yang digunakan dalam model berbeda di setiap situs. Untuk menangani situasi ini, seseorang dapat mencoba memperluas fitur sehingga semua situs dapat menggunakan model umum menggunakan prediksi fitur atau mekanisme pemetaan fitur. Seseorang dapat mencoba mempelajari aturan di berbagai fitur dari situs berbeda juga. Daripada mencoba membangun model umum atau memperkaya model lokal mereka, pilihan lain yang layak untuk banyak operasi bisnis adalah dengan menggunakan hasil inferensi dari banyak lokasi berbeda, sehingga menghasilkan inferensi gabungan.

BAB 9
KASUS PENGGUNAAN

Pada Bab 2, kita membahas beberapa industri yang memberikan motivasi untuk mengembangkan AI gabungan. Dalam masing-masing industri ini, terdapat beberapa kasus penggunaan AI gabungan. Dalam bab ini, kami menyatukan berbagai pendekatan yang dijelaskan dalam bab-bab sebelumnya untuk membahas bagaimana AI gabungan dapat digunakan dalam beberapa kasus penggunaan di dunia nyata. Kasus penggunaan di bagian ini hanyalah contoh yang menggabungkan pendekatan yang dibahas sebelumnya untuk memecahkan masalah bisnis. Banyak kombinasi lain yang dapat dilakukan untuk mengatasi masalah bisnis lainnya.

9.1 DETEKSI PENIPUAN KOLABORATIF

Pada bagian ini, kita melihat tantangan spesifik dalam mendeteksi penjahat yang mungkin mencoba menipu masyarakat. Kasus penggunaan spesifik yang akan kami targetkan adalah orang-orang yang menyerukan kepada masyarakat untuk mencoba membuat mereka mentransfer uang kepada para penipu. Penipu dapat menggunakan berbagai teknik untuk tujuan ini, mendorong korban untuk mentransfer dana kepada mereka menggunakan mekanisme yang sulit dilacak dan dibalik, misalnya melalui transfer bank. pembelian kartu hadiah. Kasus penggunaan ini dieksplorasi atas permintaan sebuah bank di Eropa yang sedang menyelidiki pendekatan untuk mengurangi penipuan di kalangan nasabahnya.

Untuk mencegah penipuan semacam ini, diusulkan untuk membentuk konsorsium bank dan operator telepon. Meskipun berbagai bank bersaing satu sama lain, mereka juga akan berkolaborasi satu sama lain untuk mencegah penipuan yang berdampak secara kolektif pada bank-bank tersebut. Demikian pula, operator telepon akan bersedia berkolaborasi dalam usaha ini karena hal ini akan memberikan perlindungan terhadap penyalahgunaan jaringan mereka.

Tujuannya adalah untuk mengidentifikasi penipu melalui analisis kolaboratif terhadap perilaku mereka. Kita mungkin mengira seorang penipu mempunyai pola panggilan yang berbeda dibandingkan dengan masyarakat pada umumnya. Namun, perilaku penipu dalam menelepon mungkin tidak jauh berbeda dengan perilaku perusahaan pemasaran atau survei yang menghubungi sebagian besar masyarakat untuk meminta pendapat mereka, atau untuk ikut serta dalam aktivitas promosi seputar produk mereka. Jika perusahaan telekomunikasi mencoba menandai penipu berdasarkan pola panggilan mereka saja, maka mereka akan mendapatkan tingkat positif palsu yang sangat tinggi dalam pendeteksian tersebut.

Demikian pula, bank mungkin memperhatikan bahwa beberapa klien mereka menunjukkan perilaku yang tidak biasa, seperti menerima transfer dalam jumlah besar yang tidak terduga ke rekening mereka, melakukan pembelian kartu hadiah, atau aktivitas lain yang di luar kebiasaan klien lain. Hal ini dapat diidentifikasi melalui algoritma deteksi anomali berbasis AI (lihat Bagian 1.8.3 di Bab 1). Jika penipu menggunakan rekening mereka sendiri

untuk menerima transfer, pola setoran mereka akan berbeda dibandingkan dengan pengguna biasa dan bank mungkin dapat mendeteksi mereka. Jika penipu tidak mentransfer dana langsung ke rekeningnya, namun membujuk korbannya untuk mengirimkan dana menggunakan kartu tunai/kartu hadiah yang diterbitkan oleh bank, bank dapat mendeteksi pola yang tidak biasa dalam penerbitan kartu tunai tersebut, atau pola yang tidak biasa dalam penerbitan kartu tunai tersebut. pembelanjaan yang dilakukan pada kartu tunai mereka memiliki masalah. Sekalipun penipu menggunakan transfer langsung ke rekeningnya sendiri, pola transaksi penipu mungkin serupa dengan bisnis yang mengirimkan produk secara online, dan mungkin sulit dibedakan dari pola transnasional pada umumnya. Akan lebih sulit lagi untuk mendapatkan pola yang teridentifikasi dari penggunaan atau penerbitan kartu tunai.

Meskipun penipu mungkin memiliki pola serupa dengan beberapa bisnis sah dalam panggilan telepon, dan penipu mungkin memiliki pola serupa dengan beberapa bisnis sah dalam transaksi keuangannya, kecil kemungkinannya mereka memiliki pola yang sama dengan bisnis sah di kedua industri. Jika konsorsium bank dan perusahaan telepon dapat disatukan untuk berkolaborasi dalam mendeteksi penipuan, kinerja mereka kemungkinan besar akan lebih baik.

Mari kita jelajahi bagaimana solusi untuk berkolaborasi dalam proses deteksi penipuan dapat dikembangkan.

9.1.1 Kolaborasi dalam Satu Industri

Mari kita pertimbangkan situasi awal di mana hanya bank yang akan menjadi bagian dari konsorsium untuk mencegah transaksi penipuan. Pembahasan pada bagian ini juga berlaku pada situasi ketika hanya perusahaan telepon yang berkolaborasi untuk mencegah transaksi penipuan. Karena perusahaan yang berkolaborasi berada dalam industri yang sama, mereka menjalankan fungsi serupa dan akan menjaga informasi serupa tentang klien mereka. Secara individual, masing-masing bank mungkin telah memilih format atau skema yang berbeda untuk menyimpan informasi. Untuk berkolaborasi bersama, bank perlu menyepakati skema yang sama. Mereka akan menerjemahkan datanya ke dalam skema umum. Dengan semua data dalam skema umum, bank dapat mempelajari model bersama untuk mendeteksi dan mencegah perilaku penipuan menggunakan algoritma pembelajaran gabungan yang dijelaskan dalam buku ini.

Selain menyepakati skema umum, para peserta perlu menyepakati beberapa hal lain, seperti frekuensi pembaruan model yang memprediksi perilaku penipuan, dan mekanisme inferensi untuk mendeteksi penipu menggunakan model tersebut. Setelah model bersama dipelajari, masing-masing bank dapat mengambil model bersama tersebut, dan memilih untuk menjalankan transaksi mereka melalui model tersebut secara lokal. Alternatifnya, mereka mungkin memutuskan bahwa mereka ingin server konsorsium memeriksa setiap transaksi penipuan. Yang terakhir ini kemungkinan akan menghadapi lebih banyak penolakan di dunia nyata, karena penggunaan inferensi akan memaparkan lebih banyak informasi pribadi dari bank yang berpartisipasi ke server konsorsium. Model sebelumnya, dimana bank dapat berbagi model mereka dan tidak berbagi informasi pribadi tertentu, kemungkinan besar akan

diadopsi. Namun demikian, tergantung pada sifat perjanjian bisnis antar perusahaan, salah satu opsi ini dapat dilakukan.

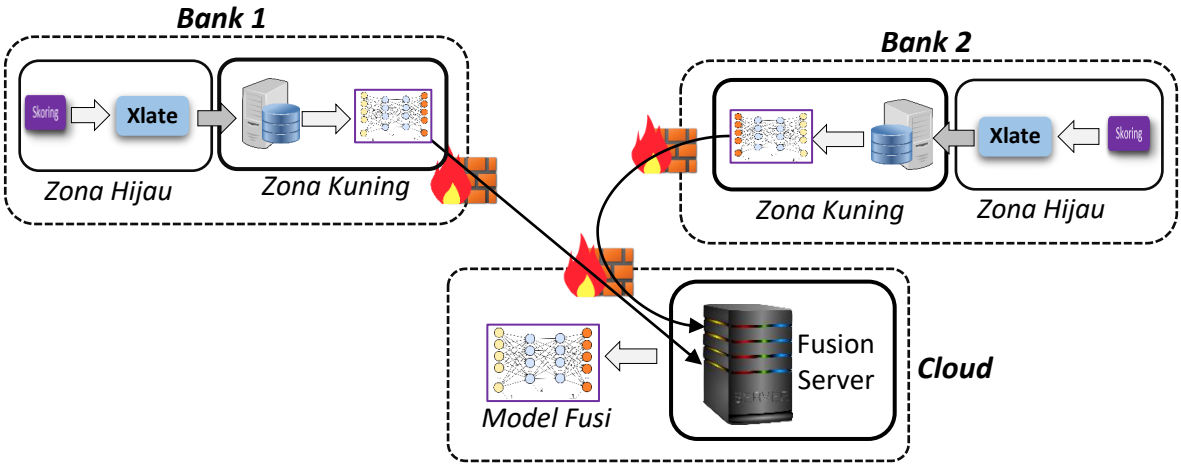
Sebagai gambaran, kita asumsikan bahwa model gabungan dilatih setiap bulan sekali, dan kemudian didistribusikan ke masing-masing bank. Bank kemudian akan menggunakan model tersebut untuk memeriksa transaksi mereka sendiri, dan memberikan pemeriksaan manual lapis kedua pada setiap transaksi yang tampaknya mencurigakan. Dengan pemeriksaan manual, dapat diterima (meskipun tidak diinginkan) jika beberapa transaksi normal ditandai sebagai penipuan karena pemeriksaan manual akan memperbaiki kesalahan tersebut.

Setelah bank setuju untuk berpartisipasi dalam tugas deteksi penipuan kolaboratif, masing-masing bank yang berpartisipasi juga perlu menyiapkan infrastruktur yang diperlukan untuk deteksi penipuan kolaboratif. Salah satu solusi yang mungkin dilakukan adalah konsorsium menjalankan server fusi model bersama dalam sistem penyedia cloud, yang digunakan untuk melatih model secara kolaboratif. Untuk pelatihan model, infrastruktur yang mungkin terlihat seperti yang ditunjukkan pada Gambar 9.1. Setiap bank menyimpan data transaksinya dalam format eksklusif di zona hijaunya. Perusahaan akan menerjemahkan informasi ke dalam format yang disepakati bersama ke servernya di zona kuning di lokasinya sebulan sekali untuk memungkinkan pemeriksaan model. Definisi zona kuning dan zona hijau akan dijelaskan pada Bagian 6.4. Zona kuning memberikan akses terbatas terhadap sistem eksternal, sedangkan zona hijau menampung sistem yang hanya dapat diakses di dalam bank. Sistem zona hijau akan mengonversi data dalam format umum ke dalam model yang dilatih berdasarkan data lokal dan menyalinnya ke sistem zona kuning. Pengaturan layanan di cloud publik digunakan untuk menggabungkan berbagai model yang disediakan oleh semua bank ke dalam model gabungan.

Sebelum model dilatih, sistem bank di zona kuning perlu berkoordinasi dengan server fusi di cloud untuk menentukan konvensi normalisasi datanya, seperti yang dijelaskan dalam Bagian 4.1.2 di Bab 4. Sistem bank di zona kuning memiliki peran klien fusi, sesuai struktur yang ditunjukkan pada Gambar 3.1. Dengan asumsi bahwa perjanjian bisnis memungkinkan panggilan masuk dari server fusi ke klien fusi, bank dapat beroperasi secara tersinkronisasi untuk membuat model menggunakan algoritme yang dijelaskan di Bab 3. Namun, bank mungkin lebih memilih untuk melatih model di secara asinkron, menggunakan teknik yang dijelaskan dalam Bab 7, jika mereka ingin bank dan perusahaan telepon baru bergabung dengan konsorsium di masa depan.

Setelah model dilatih, model tersebut diberikan kepada masing-masing bank yang berpartisipasi. Klien fusi dapat mengambilnya ke server fusi zona kuning, dan sistem di zona hijau dapat mengambilnya dari zona kuning. Model ini akan valid untuk digunakan selama sekitar satu bulan, setelah itu sesi pelatihan model berikutnya akan dilakukan. Selama bulan tersebut, setiap bank akan mengambil data transaksi mereka, menjalankannya melalui model untuk mengidentifikasi potensi perilaku penipuan, dan menandai setiap transaksi mencurigakan atau klien yang mencurigakan untuk penyelidikan manual lebih lanjut.

Identifikasi ini dapat dilakukan secara real-time berdasarkan transaksi per transaksi, atau calon nasabah penipu dapat diidentifikasi dengan memeriksa transaksinya setiap hari.



Gambar 9.1 Infrastruktur untuk pembangunan model kolaboratif.

9.1.2 Kolaborasi Lintas Industri

Meskipun bank dan operator telepon pada prinsipnya sepakat bahwa kolaborasi antar industri akan menghasilkan deteksi transaksi penipuan yang lebih baik, mereka harus mengatasi tantangan karena mereka berada di lini bisnis yang berbeda dan mengumpulkan jenis informasi yang berbeda. Meskipun bank dapat menyepakati skema umum dalam transaksinya, dan perusahaan telepon dapat menyepakati skema umum dalam transaksinya, namun tidak ada skema umum yang berlaku antara perusahaan telepon dan perusahaan bank. Akibatnya, model umum di berbagai industri tidak dapat dibangun.

Di kedua jenis industri tersebut, terdapat beberapa bidang yang sama untuk semua klien, karena perusahaan telepon dan perusahaan perbankan semuanya melayani populasi yang sama. Nomor telepon, alamat email, dan alamat surat yang diberikan sebagai informasi kontak ke perusahaan perbankan, biasanya juga diberikan kepada perusahaan telepon dan merupakan beberapa bidang umum di kedua jenis perusahaan. Beberapa perusahaan telepon mungkin juga memiliki nomor rekening dan informasi bank bagi klien mereka untuk membayar tagihan mereka. Bidang-bidang umum ini menyediakan mekanisme untuk mencocokkan klien di kedua industri ini.

Kesamaan ini berarti bahwa teknik yang digunakan untuk data yang dipartisi secara vertikal yang dijelaskan di Bagian 8 dapat digunakan untuk kolaborasi antar industri yang berbeda. Sebagai contoh, kita dapat menggunakan teknik inferensi gabungan (Bagian 8.5) sebagai metodologi untuk menggabungkan wawasan dari dua industri yang berbeda. Dengan asumsi bank dan operator telepon mengidentifikasi tersangka penipu di antara klien mereka dengan menganalisis transaksi setiap hari. Mereka dapat melakukan pengecekan terhadap tersangka penipu dengan bank lain atau operator telepon dengan berbagi fitur umum, dan mengumpulkan hasil inferensi dari pihak lain. Informasi yang dikumpulkan ini akan membantu mereka membuat keputusan yang lebih baik.

Untuk membagikan hasil dari peserta lain, masing-masing anggota konsorsium dapat memilih untuk membagikan identitas klien yang diduga melakukan penipuan kepada server fusi. Daftar tersangka ini dapat membagikan daftar yang dihasilkan kepada masing-masing anggota lainnya sesuai permintaan. Anggota konsorsium akan mendapatkan himpunan semua anggota yang dicurigai melakukan penipuan dari anggota lain, dan dapat membandingkannya dengan himpunan klien mereka sendiri untuk menentukan klasifikasi mayoritas anggota terhadap klien mereka. Perhatikan bahwa anggota mana pun yang tidak disebutkan secara eksplisit sebagai calon penipu oleh anggota lain dapat dianggap sebagai pengguna biasa. Hal ini memungkinkan pembagian hasil inferensi dengan bandwidth yang efisien (hanya sejumlah kecil tersangka penipu yang terlibat dalam pertukaran informasi) dan tidak mengharuskan peserta melakukan apa pun selain melakukan panggilan keluar ke situs cloud. Berbagi informasi untuk deteksi dan pencegahan penipuan biasanya diperbolehkan berdasarkan aturan privasi data, seperti GDPR Eropa.

Jika terdapat beberapa bank dan beberapa perusahaan telepon, bank dapat membuat model deteksi penipuan bank bersama menggunakan pembelajaran gabungan dan operator telepon dapat membuat model deteksi penipuan telepon bersama menggunakan pembelajaran gabungan. Perhatikan bahwa model yang sama dapat menandai pengguna yang sama sebagai pengguna yang melakukan penipuan di satu bank dan tidak melakukan penipuan di bank lain karena kedua bank tersebut memiliki rangkaian fitur yang berbeda untuk mengkarakterisasi pengguna. Bank dan perusahaan telepon dapat berkolaborasi satu sama lain menggunakan inferensi gabungan.

9.1.3 Efektivitas

Efektivitas model AI apa pun dalam kasus penggunaan apa pun bergantung pada pola yang terdapat dalam data yang digunakan untuk pelatihan dan pengujian model. Hasil dalam situasi apa pun akan bergantung pada data, algoritma pembelajaran model yang digunakan, arsitektur model yang dipilih, dan pendekatan pembelajaran gabungan yang dipilih. Oleh karena itu, hasil-hasil yang dibahas pada bagian ini tidak boleh dianggap berlaku dalam semua skenario dan keadaan. Namun, mereka memberikan poin data yang mewakili manfaat komparatif dari berbagai teknik dan pendekatan.

Hasil pada bagian ini menunjukkan keefektifan tiga pendekatan berbeda untuk mendeteksi penipu di suatu komunitas. Untuk hasil khusus ini, diasumsikan bahwa dua bank dan dua operator telepon merupakan bagian dari konsorsium, dan mereka memeriksa transaksi mereka setiap hari untuk mengidentifikasi klien yang mungkin berpotensi menjadi penipu. Hasil diberikan untuk tiga mode operasional yang berbeda. Transaksi sintetis dihasilkan untuk mensimulasikan perilaku sekelompok perusahaan normal, konsumen normal, dan penipu untuk melatih dan menguji model. Pembuatan data sintetis memberikan akses terhadap kebenaran dasar mengenai klien bank/perusahaan telepon mana dalam sistem yang merupakan pengguna biasa dan mana yang merupakan penipu.

Pendekatan pertama adalah operasi independen. Dalam mode khusus ini, setiap bank dan perusahaan telepon beroperasi secara independen. Mereka tidak berbagi data satu sama lain. Setiap perusahaan mengoperasikan sistemnya sendiri untuk mengidentifikasi penipu,

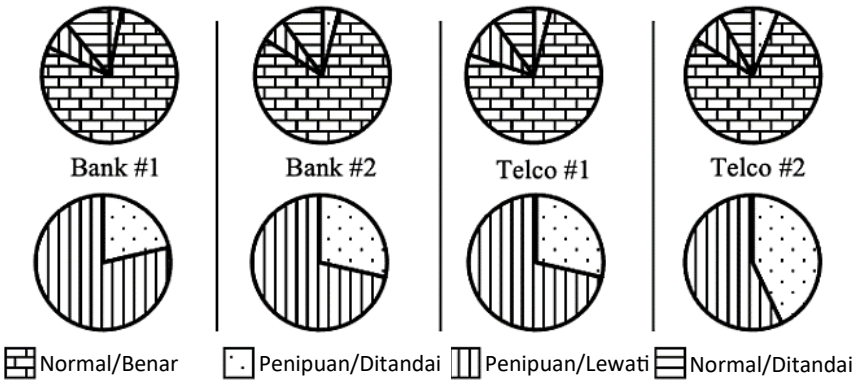
yang mengawasi transaksi anomali yang dilakukan oleh kliennya. Dalam analisis ini, algoritma yang digunakan adalah algoritma tanpa pengawasan yang menandai anomali melalui auto-encoder. Pembuat enkode otomatis dijalankan pada transaksi masing-masing perusahaan pada akhir hari untuk menentukan transaksi yang tidak wajar. Diagram lingkaran di bagian atas untuk setiap perusahaan menunjukkan rincian seberapa efektif perusahaan tersebut dalam mengkarakterisasi kliennya ke dalam kategori yang tepat. Karena setiap perusahaan memiliki kombinasi fitur dan model bersama yang berbeda, hasilnya mungkin berbeda untuk perusahaan yang berbeda. Diagram lingkaran di bagian bawah hanya mempertimbangkan kumpulan penipu aktif dan menunjukkan apakah penipu teridentifikasi dengan benar atau tidak.

Untuk diagram lingkaran di atas, ada empat kemungkinan hasil (i) Seorang penipu ditandai dengan benar sebagai penipu (ii) Seorang penipu terlewatkan dan dianggap sebagai pengguna biasa (iii) Pengguna biasa diidentifikasi dengan benar sebagai pengguna biasa dan (iv) Pengguna biasa salah ditandai sebagai penipu. Masing-masing dari empat kategori ditampilkan menggunakan corak potongan berbeda pada diagram lingkaran. Idealnya, sistem hanya terdiri dari kategori (i) dan (iii) namun karena tidak ada algoritma pembelajaran yang sempurna, beberapa kesalahan identifikasi dapat terjadi.

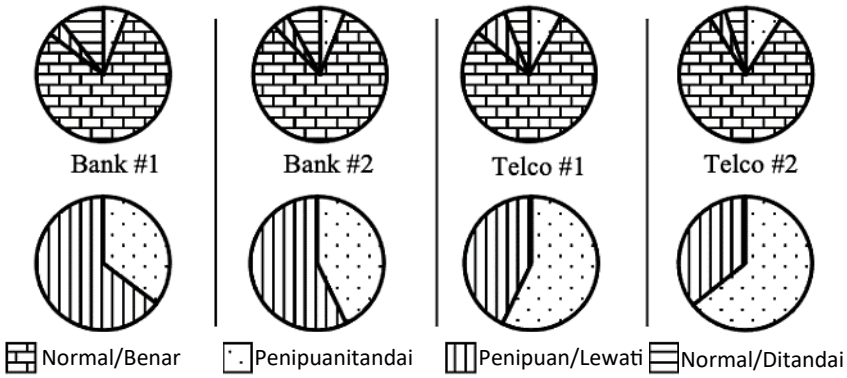
Mengingat adanya pemeriksaan manual lapis kedua, yang lebih penting adalah semua penipu diidentifikasi dengan benar dan beberapa orang normal yang salah ditandai sebagai penipu dapat diterima (tetapi tidak diinginkan). Diagram lingkaran di bawah menunjukkan hasil bagi para penipu, yang terbagi dalam dua kategori, apakah mereka ditandai dengan benar atau terlewatkan. Diagram lingkaran ini menunjukkan mekanisme yang lebih sederhana mengenai efektivitas mekanisme dengan menggunakan sebagian kecil penipu yang terdeteksi dengan benar sebagai metrik kinerja.

Jika perusahaan tidak berkolaborasi satu sama lain, kinerja masing-masing perusahaan dalam mengidentifikasi penipu menjadi tidak baik. Sebagian besar penipu terlewatkan, dan sebagian besar pengguna biasa salah ditandai sebagai penipu. Meskipun hal ini mungkin tidak berdampak pada kemampuan mereka untuk menyelesaikan transaksi, hal ini menyia-nyiakan waktu pegawai bank atau telepon saat mereka memprosesnya pada langkah berikutnya. Jika saja bank dan perusahaan telekomunikasi mampu membentuk konsorsium masing-masing, mereka bisa membangun model bersama untuk melacak para penipu. Model bersama akan mengidentifikasi ketidaknormalan penggunaan di beberapa perusahaan.

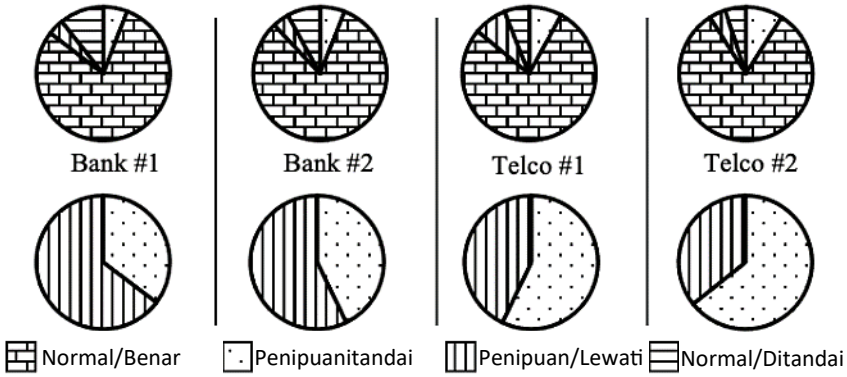
Dalam kasus khusus ini, kedua bank akan berbagi model mereka satu sama lain menggunakan pengaturan yang dijelaskan di Bagian 9.1.1. Hasilnya, kedua bank mempelajari model identifikasi penipu yang memahami pola transaksi keduanya. Demikian pula, masing-masing perusahaan telepon dapat mempelajari model identifikasi penipu yang menangkap pola panggilan telepon di semua perusahaan telepon. Diharapkan kemampuan model dalam mengidentifikasi penipu akan lebih baik.



Gambar 9.2 Hasil deteksi independen.



Gambar 9.3 Hasil deteksi melalui kolaborasi dalam industri.



Gambar 9.4 Hasil deteksi melalui kolaborasi antar industri.

Hasilnya, yang ditunjukkan pada Gambar 9.3, menegaskan bahwa terdapat perbaikan jika setiap perusahaan mencoba beroperasi sendiri, seperti yang ditunjukkan pada Gambar 9.2. Persentase penipu yang teridentifikasi dengan benar telah meningkat, dan persentase pengguna biasa yang salah diidentifikasi sebagai penipu telah menurun. Seperti yang diharapkan, berbagi informasi di seluruh perusahaan dalam suatu industri menghasilkan skema deteksi penipuan yang lebih baik.

Jika terdapat kolaborasi lintas industri, bank dan perusahaan telepon dapat berbagi informasi satu sama lain menggunakan kombinasi pembelajaran gabungan dan inferensi

gabungan, seperti yang dijelaskan dalam Bagian 9.1.2. Hasil kolaborasi lintas industri ditunjukkan pada Gambar 9.4. Dalam kasus khusus ini, semua penipu teridentifikasi dengan benar, dan terdapat persentase yang sangat kecil dari pengguna biasa yang salah ditandai sebagai penipu. Kinerja ini jauh lebih baik dibandingkan dua pendekatan lainnya.

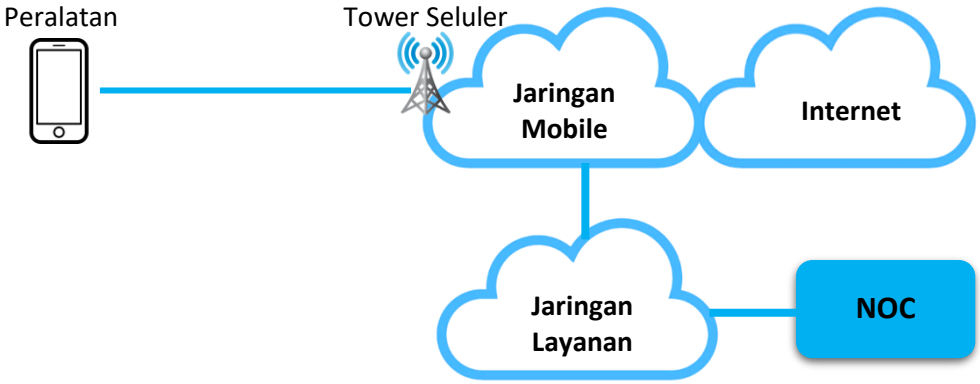
Meskipun kita tidak dapat mengklaim bahwa hasil yang sama baiknya akan diperoleh dalam semua jenis deteksi penipuan, akan cukup adil untuk mengklaim bahwa pembelajaran gabungan antar perusahaan dalam industri yang sama akan lebih baik dalam mendeteksi penipuan jika masing-masing perusahaan mencobanya sendiri. Demikian pula, kolaborasi lintas industri yang menggunakan inferensi gabungan dapat memberikan peningkatan yang signifikan dibandingkan kolaborasi dalam satu industri.

Penipuan keuangan di dunia nyata hadir dalam berbagai bentuk dan dapat diidentifikasi menggunakan banyak pendekatan berbasis AI/ML yang berbeda. Meskipun identifikasi yang sempurna tidak selalu dapat dilakukan, hasil pada bagian ini memberikan penegasan harapan bahwa pembelajaran gabungan dan inferensi gabungan dapat meningkatkan status quo.

9.2 MANAJEMEN JARINGAN FEDERASI

Kolaborasi antar berbagai organisasi dalam lingkungan konsorsium dapat menimbulkan tantangan tersendiri dalam pelaksanaannya. Membuat beberapa organisasi menyetujui tingkat kolaborasi satu sama lain adalah hal yang tidak sepele. Namun, ada banyak penerapan pembelajaran gabungan dalam satu perusahaan. Pada bagian ini, kita akan mengkaji kasus penggunaan pembelajaran gabungan yang diperlukan dalam konteks satu perusahaan, yaitu operator jaringan telekomunikasi. Masalah spesifik yang akan kita bahas adalah pengelolaan jaringan seluler. Kasus penggunaan dikembangkan untuk operator jaringan di Amerika Serikat.

Jaringan seluler menyediakan infrastruktur yang memungkinkan komunikasi telepon seluler modern terjadi dengan lancar. Struktur umum jaringan seluler pada tingkat yang sangat tinggi ditunjukkan pada Gambar 9.5. Lebih detail mengenai infrastruktur jaringan dapat dilihat pada referensi . Dalam jaringan seluler, telepon seluler terhubung ke menara seluler melalui udara sebagai tahap awal pertukaran data dengan server di Internet. Menara seluler yang berbeda saling terhubung melalui jaringan seluler yang dioperasikan oleh operator jaringan seluler. Jaringan seluler menjalankan protokol komunikasi (misalnya protokol 4G atau 5G) yang memungkinkan telepon seluler dan peralatan menara seluler berinteraksi dengan sistem yang menyediakan akses ke Internet. Operator jaringan seluler juga menjalankan jaringan layanan yang digunakan untuk mendukung berbagai fungsi pendukung yang diperlukan untuk operasional jaringan, yang meliputi sistem pendukung penagihan, manajemen, dan operasional. Salah satu fungsi spesifik dalam jaringan layanan adalah Network Operations Center (NOC), yang menampung berbagai sistem yang digunakan untuk memantau, memecahkan masalah, dan memperbaiki masalah apa pun yang mungkin timbul selama pengoperasian jaringan seluler.



Gambar 9.5 Struktur jaringan seluler.

Salah satu tantangan dalam pengelolaan jaringan seluler adalah skalanya yang besar. Dalam lingkungan jaringan seluler pada umumnya, terdapat puluhan ribu menara seluler, sekitar seribu pusat data jaringan yang bertindak sebagai lokasi perantara untuk pemrosesan protokol, dan beberapa lokasi pertukaran tempat mereka berinteraksi dengan Internet. Pusat Operasi Jaringan perlu memantau mesin di beberapa ribu lokasi, memahami jika ada di antara mesin yang mengalami masalah, dan kemudian mengambil tindakan perbaikan. Mengingat banyaknya instrumentasi dalam infrastruktur komunikasi jaringan, setiap kegagalan dapat dideteksi dan dilaporkan ke NOC dengan cukup cepat. Selain itu, kejadian yang terjadi pada menara seluler dapat memberikan indikasi kegagalan yang mungkin terjadi di masa mendatang. Peristiwa ini mencakup entri yang dicatat dalam log sistem peralatan serta pesan yang dipertukarkan antar mesin berbeda yang berkomunikasi dengan peralatan di menara seluler untuk menjalankan jaringan.

Untuk membangun model seperti itu, NOC memerlukan akses terhadap informasi kejadian dan kegagalan di berbagai menara seluler. Kegagalan diberitahukan kepada NOC, namun dengan kejadian yang terjadi pada peralatan di sepuluh ribu menara seluler (angka ini untuk operator seluler kecil) dan kemudian dianalisis sepanjang hari untuk memahami korelasinya dengan kegagalan. Indikasinya, NOC perlu mengumpulkan sekitar sepuluh miliar peristiwa setiap hari. Pengumpulan data ini untuk membangun model prediksi kegagalan saja akan menghabiskan sejumlah besar bandwidth dalam jaringan seluler yang mungkin lebih disukai oleh operator jaringan untuk menghasilkan lalu lintas yang menghasilkan pendapatan dari pengguna telepon selulernya.

Daripada menggunakan pendekatan di mana NOC mengumpulkan data dari semua menara sel, akan lebih baik jika membiarkan setiap menara sel melatih model lokalnya sendiri yang dapat mempelajari korelasi antara peristiwa yang terjadi di menara sel dan kemungkinannya. kegagalan di masa depan. Model ini kemudian dapat digabungkan ke seluruh menara sel yang berbeda dan digunakan untuk terus memantau kejadian di menara sel untuk mendapatkan peringatan dini tentang potensi kegagalan.

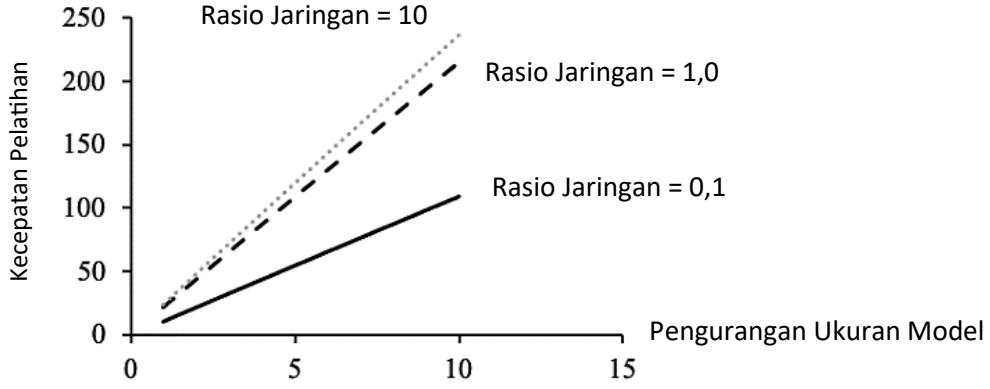
Peristiwa kegagalan jarang terjadi, sehingga setiap menara seluler mungkin harus memantau data selama beberapa hari sebelum mengumpulkan cukup data untuk melatih model lokal. Model lokal kemudian dapat digabungkan secara berkala di NOC. Dalam

lingkungan spesifik ini, semua menara seluler berada di bawah satu domain administratif, sehingga sistem dapat memilih satu cara yang konsisten dalam pengumpulan data peristiwa, model yang dilatih, dan bahkan dapat mengoordinasikan pelatihan sehingga algoritma pelatihan tersinkronisasi seperti yang ada. disebutkan dalam Bab 3 dapat digunakan.

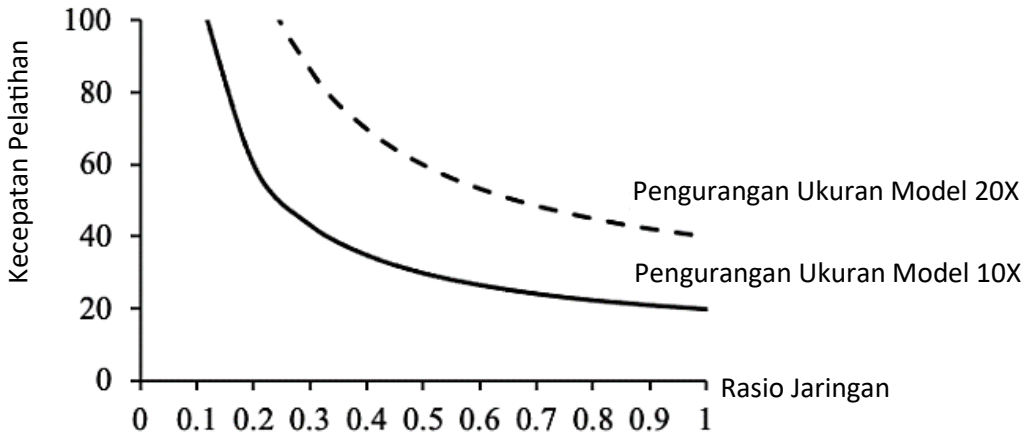
Alasan penggunaan pendekatan pembelajaran gabungan untuk model pelatihan dalam manajemen jaringan adalah penghematan bandwidth yang signifikan yang dapat diperoleh saat membuat model. Dengan menggunakan persamaan untuk kecepatan pembelajaran gabungan, dan fakta bahwa terdapat sejumlah besar menara seluler dalam sistem, perolehan relatif dalam waktu yang dibutuhkan untuk melatih model dapat dihitung menggunakan dua faktor utama, pertama pengurangan dalam ukuran model dan kedua kecepatan relatif jaringan diukur dengan apa yang kita sebut sebagai rasio jaringan. Rasio jaringan berkisar dari nol hingga tak terhingga dan membandingkan kecepatan relatif pelatihan model di edge pada sekumpulan data pelatihan dibandingkan dengan waktu yang dibutuhkan untuk mentransfer data melalui jaringan ke NOC untuk melatih data di lokasi pusat. Jika jaringannya sangat cepat, rasio ini akan mendekati nol. Sebaliknya, jika jaringan relatif lambat, rasio jaringan akan jauh lebih tinggi.

Metrik pengurangan ukuran model mengukur pengurangan ukuran model sebagai rasio ukuran model terhadap ukuran data yang digunakan untuk melatih model. Pengurangan model 0,1 berarti model tersebut berukuran sepersepuluh dari data pelatihan. Pengurangan model sebesar 0,01 berarti model tersebut berukuran seperseratus dari data pelatihan model. Pembelajaran gabungan menghasilkan pelatihan model AI jauh lebih cepat daripada mencoba memindahkan data ke NOC. Kecepatan waktu yang dibutuhkan untuk melatih model ditunjukkan pada Gambar 9.6. Sebuah sistem dengan sepuluh ribu menara seluler dianalisis. Seperti yang ditunjukkan pada gambar, percepatannya linier dengan pengurangan ukuran model, dan kemiringan percepatan ditentukan oleh rasio jaringan.

Untuk memahami dampak rasio jaringan, kami memplot kecepatan terhadap rasio jaringan untuk dua kondisi, kondisi pertama ketika rasio jaringan kurang dari 1 (yaitu jaringan relatif cepat dan dapat mentransfer data jauh lebih cepat daripada yang diperlukan untuk melatih model), yang kedua adalah ketika rasio jaringan lebih besar dari 1 (yaitu jaringan relatif lambat dan pelatihan model secara lokal lebih cepat daripada waktu yang diperlukan untuk mentransfer data).



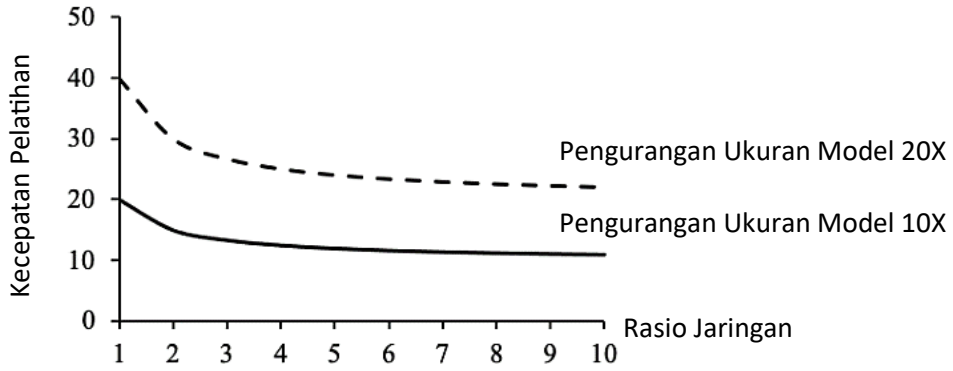
Gambar 9.6 Kinerja relatif pembelajaran gabungan terhadap pendekatan terpusat.



Gambar 9.7 Kinerja relatif pembelajaran gabungan terhadap pendekatan terpusat dengan jaringan cepat.

Gambar 9.7 menunjukkan kecepatan ketika jaringan relatif cepat dibandingkan dengan tugas melatih model. Ketika jaringan sangat cepat, pembelajaran gabungan menunjukkan peningkatan yang signifikan dalam kecepatan penggunaan pembelajaran gabungan. Alasannya adalah peningkatan signifikan dalam paralelisme pembuatan model. Masing-masing menara seluler melatih modelnya pada segmen data secara paralel, yang bertanggung jawab atas percepatannya. Dalam kasus komunikasi jaringan, yang jumlah menara selulernya mencapai puluhan ribu, kecepatannya sangat luar biasa.

Gambar 9.8 menunjukkan kecepatan ketika jaringan relatif lambat dibandingkan dengan tugas melatih model. Dalam hal ini, kecepatan pembelajaran gabungan diatur oleh jumlah data yang perlu ditransfer oleh jaringan. Akibatnya, kecepatan mencapai batas asimtotik yang menunjukkan seberapa kecil model dibandingkan dengan data pelatihan. Karena model cenderung jauh lebih kecil dibandingkan jumlah data yang digunakan untuk melatihnya, hal ini mengakibatkan pendekatan berbasis pembelajaran gabungan bekerja jauh lebih baik dan lebih cepat dibandingkan pendekatan model pembelajaran terpusat.



Gambar 9.8 Kinerja relatif pembelajaran gabungan terhadap pendekatan terpusat dengan jaringan lambat.

9.3 REKOMENDASI KUPON RITEL

Kasus pembelajaran gabungan muncul di industri ritel yang dimotivasi oleh masalah keamanan, bukan masalah kinerja buruk seperti kasus penggunaan yang dijelaskan di Bagian 9.2. Ada beberapa pencurian data konsumen tingkat tinggi dari pengecer dan ritel adalah industri yang memiliki insiden pelanggaran data terbesar di berbagai sektor industri. Sebagian besar pengecer menerapkan kebijakan perlindungan data yang ketat dan langkah-langkah keamanan TI yang kuat, namun meskipun demikian, pelanggaran data yang membahayakan data pengguna terjadi karena kelalaian manusia atau kegagalan dalam prosedur kepatuhan.

Sebuah pengecer besar di Amerika Serikat khawatir bahwa setiap potensi pelanggaran data akan mengakibatkan hilangnya sejumlah besar data konsumen karena semua data disimpan di satu lokasi terpusat. Daripada menyimpan data di lokasi pusat, mereka lebih memilih untuk membiarkan data tetap berada di setiap toko ritel dengan mengikuti kebijakan keamanan yang sama yang diterapkan di gudang data pusat. Data pembelian pelanggan di toko akan dienkripsi dan disimpan secara lokal tanpa membawa informasi tersebut ke lokasi pusat. Jika terjadi pelanggaran data, data yang disusupi hanya akan menjadi milik satu toko, yang akan lebih murah untuk ditangani dan dikelola dibandingkan pelanggaran data di lokasi pusat yang dapat mengungkap informasi tentang pelanggan di seluruh rantai. Keyakinan yang mendasarinya adalah bahwa prosedur pemeliharaan data di rantai ritel cukup kuat sehingga kemungkinan pelanggaran data tidak meningkat melalui distribusi tersebut.

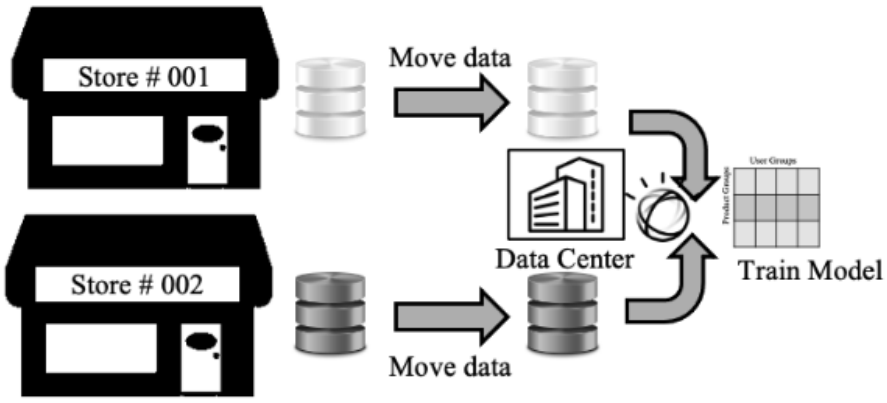
Tantangan dalam mempertahankan data yang didistribusikan dengan cara ini adalah bahwa analisis data dan infrastruktur pembelajaran model AI saat ini untuk rantai ritel dibangun berdasarkan premis bahwa semua data disimpan secara terpusat. Oleh karena itu, mekanisme setara yang memungkinkan model yang sama dibangun dengan data terdistribusi perlu dikembangkan. Meskipun ada beberapa jenis fungsi analisis data untuk rantai ritel yang dapat diimplementasikan menggunakan data terdistribusi, upaya percontohannya adalah untuk mendemonstrasikan perolehan kupon ritel sambil mempertahankan data terdistribusi sebagai contoh untuk jenis analisis lainnya.

Kupon eceran adalah kupon yang diberikan kepada pembeli pada saat mereka memeriksa pembeliannya di meja kasir. Kupon ini memberikan insentif bagi pembeli untuk membeli produk yang ditargetkan pada kunjungan berikutnya. Pembuatan kupon ini biasanya dilakukan melalui sistem rekomendasi top-N yang menentukan item mana yang terbaik untuk direkomendasikan berdasarkan riwayat transaksi pelanggan. Pengecer tempat kami bekerja menggunakan varian algoritma SLIM untuk menghasilkan kuponnya.

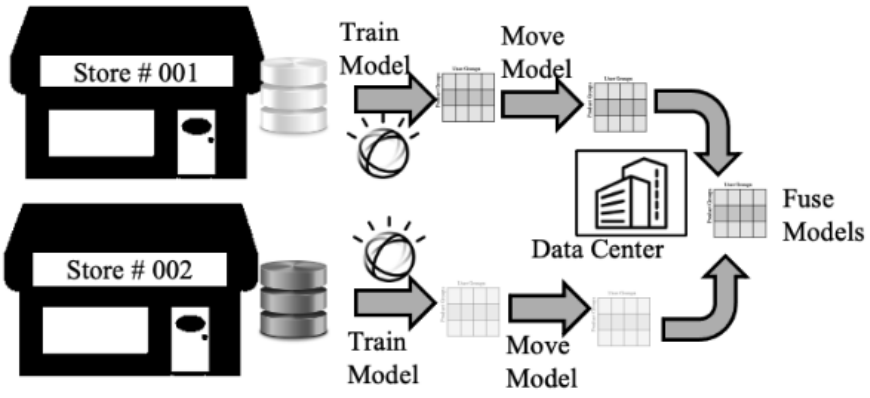
Algoritme kelas SLIM bekerja dengan menghitung kecenderungan komputasi matriks untuk membeli skor dengan baris matriks adalah kelompok pengguna dan kolom produk adalah kelompok produk. Kelompok pengguna dan produk dihitung dengan cara pengelompokan transaksi sehingga pengguna dengan perilaku pembelian yang sama dapat dikelompokkan ke dalam kelompok serupa di sepanjang baris, dan produk dengan pola penjualan serupa dimasukkan ke dalam kelompok serupa di sepanjang kolom. Setelah pengelompokan dilakukan, kecenderungan kelompok pengguna untuk membeli dari suatu kelompok dapat dihitung sebagai matriks. Kupon dihasilkan dengan melihat transaksi

pengguna saat ini, menemukan kelompok produk yang kemungkinan besar akan dibeli oleh pengguna dari matriks yang dihitung, dan kemudian memilih produk dari kelompok yang tidak dibeli oleh pengguna sebagai insentif bagi mereka untuk membeli produk tersebut. di masa depan.

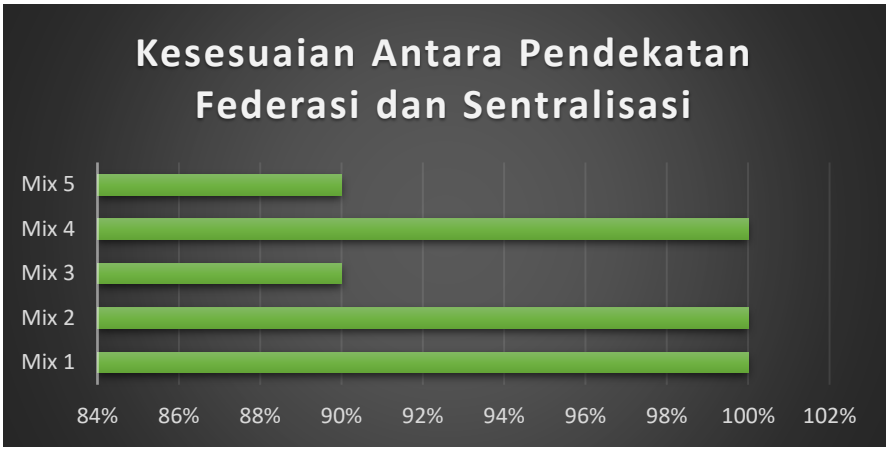
Pendekatan yang ada untuk menghasilkan kupon adalah dengan mengumpulkan data ke lokasi pusat, misalnya. pusat data pengecer, dan melatih model pada data yang dikumpulkan seperti yang ditunjukkan pada Gambar 9.9. Untuk menghasilkan model yang sama dengan cara gabungan seperti yang ditunjukkan pada Gambar 9.10, kita perlu membuat matriks untuk rekomendasi secara lokal dan kemudian menggabungkan matriks-matriks rekomendasi tersebut. Kombinasi matriks merupakan komposisi yang relatif sederhana selama kelompok atau cluster untuk pengguna dan produk konsisten di seluruh toko. Namun, pengelompokan yang dilakukan secara independen pada toko ritel yang berbeda tidak harus konsisten.



Gambar 9.9Pendekatan pembelajaran terpusat untuk menghasilkan model rekomendasi.



Gambar 9.10 Pendekatan pembelajaran gabungan untuk menghasilkan model rekomendasi.



Gambar 9.11 Hasil pembuatan model rekomendasi.

Untuk membuat pengelompokan kelompok produk dan kelompok pengguna konsisten di semua lokasi, kami membagi tugas menghitung matriks rekomendasi dalam dua tahap, yang pertama adalah tahapan di mana kelompok pengguna dan kelompok produk diidentifikasi. Hal ini secara efektif menghasilkan algoritma pengelompokan gabungan. Karena pengelompokan adalah fungsi umum yang dimodelkan oleh jaringan saraf, maka pengelompokan dapat dilakukan dengan menggunakan algoritma yang dijelaskan dalam Bab 3 atau dapat dilakukan dengan algoritma terdistribusi yang dirancang khusus dengan mempertimbangkan pengelompokan. Setelah cluster konsisten di semua situs, setiap situs dapat mempelajari matriksnya, dan matriks gabungan adalah rata-rata tertimbang dari matriks tersebut. Proses ini efisien dan dapat disesuaikan dengan ukuran matriks, jumlah penyimpanan, dan jumlah pengguna.

Perbandingan rekomendasi yang ditawarkan oleh dua pendekatan berbeda dilakukan dengan menggunakan data yang mewakili transaksi ritel. Sepuluh pola pembelian yang berbeda di antara pengguna yang berbeda disimulasikan di dua toko yang berbeda, dan campuran pola yang berbeda di toko dijalankan melalui kedua jenis sistem rekomendasi kupon, satu di mana matriks rekomendasi kupon dipelajari secara terpusat, dan satu di yang dipelajari secara gabungan. Hasilnya ditunjukkan pada Gambar 9.11. Kecocokan keseluruhan menunjukkan apakah produk dalam kelompok yang sama direkomendasikan oleh kedua algoritme. Kecocokan keseluruhan sebesar 94% menunjukkan bahwa algoritme tersebut secara efektif konsisten satu sama lain, dan mempertahankan data yang terdistribusi tidak memengaruhi kemampuan untuk membangun model AI.

9.4 RINGKASAN

Pada bagian ini, kami telah memeriksa secara singkat beberapa kasus penggunaan pembelajaran gabungan yang diterapkan pada kasus penggunaan bisnis. Meskipun ketiga hal ini merupakan contoh kasus penggunaan, pembelajaran gabungan dan inferensi gabungan dapat digunakan dalam banyak situasi lain yang analog dengan situasi yang dijelaskan di atas. Ketiga kasus penggunaan tersebut menunjukkan keragaman penerapan pembelajaran gabungan. Ketika digunakan bersama dengan inferensi dan pembelajaran, AI gabungan dapat

memberikan solusi yang layak untuk beberapa tantangan lain dalam lingkungan bisnis modern.

DAFTAR PUSTAKA

Al Badawi, B. Veeravalli, J. Lin, N. Xiao, M. Kazuaki, and A. K. M. Mi, "Multi-GPU design and performance evaluation of homomorphic encryption on GPU clusters," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 2, pp. 379–391, 2020.

Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, "Federated learning: A survey on enabling technologies, protocols, and applications," *IEEE Access*, vol. 8, pp. 140699–140725, 2020.

B. Gu, Z. Dang, X. Li, and H. Huang, "Federated doubly stochastic ker-nel learning for vertically partitioned data," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*,pp. 2483–2493, 2020.

B. Kaliski, "A survey of encryption standards," IEEE Micro, vol. 13, no. 6, pp. 74–81, 1993.

B. Steffen, F. Howar, and M. Merten, "Introduction to active automata learning from a practical perspective," in *International School on Formal Methods forthe Design of Computer, Communication and Software Systems*, pp. 256–296,Springer, 2011.

Bar-Noy, G. Cirincione, R. Govindan, S. Krishnamurthy, T. LaPorta, P. Mo- hapatra, M. Neely, and A. Yener, "Quality of information aware networking for tactical military networks," in *IEEE International Conference on PervasiveComputing and Communications*, 2011.

Bashir, Mastering Blockchain: Distributed ledger technology, decentraliza- tion, and smart contracts explained. Packt Publishing Ltd, 2018.

C. Bisdikian, L. M. Kaplan, M. B. Srivastava, D. J. Thornley, D. Verma, and C. Brodie, D. George, C.-M. Karat, J. Karat, J. Lobo, M. Beigi, X. Wang, C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer, 2018.

C. Dwork, "Differential privacy: A survey of results," in *International Con- ference on Theory and Applications of Models of Computation*, pp. 1–19, Springer, 2008.

C. Ezeh, C. O. Izugbara, C. W. Kabiru, S. Fonn, K. Kahn, L. Manderson,S. Undieh, A. Omigbodun, and M. Thorogood, "Building capacity for public and population health research in Africa: the consortium for advancedresearch training in Africa (CARTA) model," *Global Health Action*, vol. 3, no. 1, p. 5693, 2010.

C. Feng and D. Michie, "Machine learning of rules and trees," *Machine Learn-ing, Neural and Statistical Classification*, pp. 50–83, 1994.

C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceed- ings of the Forty-first Annual ACM Symposium on Theory of Computing, pp. 169–178, 2009.

C. Grosan and A. Abraham, "Rule-based expert systems," in *Intelligent Sys- tems*, pp. 149–185, 2011.

C. Kiddon, J. Konečnỳ, S. Mazzocchi, H. B. McMahan, *et al.*, "Towards Fed- erated Learning at Scale: System Design," *arXiv preprint arXiv:1902.01046*,2019.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Pro-cessing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

D. A. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometrics*, vol. 741, 2009.

D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointlylearning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

D. C. Verma and P. Verma, *Techniques for Surviving the Mobile Data Explo-sion*. John Wiley & Sons, 2014.

D. C. Verma, "Service level agreements on IP networks," Proceedings of the IEEE, vol. 92, no. 9, pp. 1382–1388, 2004.

D. C. Verma, "Simplifying network administration using policy-based man- agement," *IEEE Network*, vol. 16, no. 2, pp. 20–26, 2002.

D. C. Verma, E. Bertino, A. Russo, S. Calo, and A. Singla, "Policy based ensembles for multi domain operations," in *Artificial Intelligence and Ma- chine Learning for Multi-Domain Operations Applications II*, vol. 11413,

D. C. Verma, G. White, S. Julier, S. Pasteris, S. Chakraborty, and G. Cirin- cione, "Approaches to address the data skew problem in federated learning," in *Artificial Intelligence and Machine Learning for Multi-Domain OperationsApplications*, vol. 11006, p. 110061I, International Society for Optics and Photonics, 2019.

D. C. Verma, *Policy-based Networking: Architecture and Algorithms*. New Riders Publishing, 2000.

D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic Regres-sion*. Springer, 2002.

D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A Friendly Federated Learning Research Framework," *arXiv e-prints*,pp. arXiv–2007, 2020.

D. Niyato, and C. Miao, "Federated learning in mobile edge networks: Acomprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," D. Paraschakis, B. J. Nilsson, and J. Hollander, "Comparative evaluation of top-n recommenders in e-commerce: An industrial perspective," in *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 1024–1031, IEEE, 2015.

D. Ramage, A. Segal, and K. Seth, "Practical secure aggregation for federatedlearning on user-held data," *arXiv preprint arXiv:1611.04482*, 2016.

D. Roberts, G. Lock, and D. C. Verma, "Holistan: A futuristic scenario for international coalition operations," in *2007 International Conference on In- tegration of Knowledge Intensive Multi-Agent Systems*, pp. 423–427, IEEE, 2007.

D. Turgut and L. Boloni, "Value of information and cost of privacy in the internet of things," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 62– 66, 2017.

D. Verma, G. de Mel, and G. Pearson, "VoI for complex AI based solutions in coalition environments," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006, p. 1100609, InternationalSociety for Optics and Photonics, 2019.

D. Verma, S. Calo, E. Bertino, A. Russo, and G. White, "Policy based Ensem-bles for applying ML on big data," in *2019 IEEE International Conference onBig Data (Big Data)*, pp. 4038–4044, IEEE, 2019.

D. Verma, S. Calo, S. Witherspoon, E. Bertino, A. A. Jabal, A. Swami, G. Cir-incione, S. Julier, G. White, G. de Mel, *et al.*, "Federated Learning for CoalitionOperations," in *AAAI Fall Symposium Series: Artificial Intelligence in Gov- ernment and Public Sector, Arlington, Virginia, USA*, AAAI, 2019.

D. Verma, S. Calo, S. Witherspoon, I. Manotas, E. Bertino, A. A. Jabal,
D. Verma, S. Calo, S. Witherspoon, I. Manotas, E. Bertino, A. M. A. Jabal,

D. Verma, S. Julier, and G. Cirincione, "Federated AI for building ai solutionsacross multiple agencies," in *AAAI Fall Symposium Series: Artificial Intel- ligence in Government and Public Sector, Arlington, Virginia, USA*, AAAI, 2018.

D. Verma, S. Julier, and G. Cirincione, "Federated AI for building AI so- lutions across multiple agencies," in *AAAI FSS-18: Artificial Intelligence in Government and Public Sector, Arlington, VA, USA*, 2018.

D. W. Cheung, V. T. Ng, A. W. Fu, and Y. Fu, "Efficient mining of association rules in distributed databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, pp. 911–922, 1996.

E. A. Lile, "Client/Server architecture: A brief overview," *Journal of SystemsManagement*, vol. 44, no. 12, p. 26, 1993.

E. Bagdasaryan, A. Veit, Y. Hua, D. Estrin, and V. Shmatikov, "How to back-door federated learning," in *International Conference on Artificial Intelligenceand Statistics*, pp. 2938–2948, PMLR, 2020.

E. Bakopoulou, B. Tillman, and A. Markopoulou, "A federated learning ap- proach for mobile packet classification," *arXiv preprint arXiv:1907.13113*, 2019.

E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "Scalable density-based distributedclustering," in *European Conference on Principles of Data Mining and Knowl- edge Discovery*, pp. 231–244, Springer, 2004.

E. Marseille, *The Rapid Growth of Data Breaches in Today's Society*. PhD thesis, Utica College, 2020.

E. Milanov, "The RSA algorithm," RSA Laboratories, pp. 1–11, 2009.

F. A. Khan, S. Butt, S. A. Khan, L. Bölöni, and D. Turgut, "Value of informationbased data retrieval in uwsns," *Sensors*, vol. 18, no. 10, p. 3414, 2018.

F. Hendrikx, K. Bubendorfer, and R. Chard, "Reputation systems: A survey and taxonomy," *Journal of Parallel and Distributed Computing*, vol. 75, pp. 184–197, 2015.

F. Le, K. Leung, K. Poularakis, L. Tassiulas, and Y. Paul, "Extracting Inter- pretable Rules from Deep Models across Coalitions," in *Proceedings of the Annual Fall Meeting of DAIS ITA*, 2020.

F. Sattler, K.-R. Müller, and W. Samek, "Clustered federated learning: Model- agnostic distributed multitask optimization under privacy constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

G. A. Seber and A. J. Lee, *Linear Regression Analysis*, vol. 329. John Wiley& Sons, 2012.

G. Cirincione and D. Verma, “Federated Machine Learning for Multi-DomainOperations at the tactical edge,” in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006, p. 1100606, Interna-tional Society for Optics and Photonics, 2019.

G. Cirincione, A. Swami, G. Pearson, and G. de Mel, “Self-generating policies for machine learning in coalition environments,” in *Policy-Based AutonomicData Governance*, pp. 42–65, Springer, 2019.

G. de Mel, A. Swami, G. Cirincione, and G. Pearson, “Managing training data from untrusted partners using self-generating policies,” in *Artificial In- telligence and Machine Learning for Multi-Domain Operations Applications*, vol. 11006, p. 110060P, International Society for Optics and Photonics, 2019.

G. Pearson and T. Pham, “The challenge of sensor information processing and delivery within network and information science research,” in *Defense Transformation and Net-Centric Systems*, vol. 6981, p. 698105, InternationalSociety for Optics and Photonics, 2008.

G. Pearson, D. Verma, and G. de Mel, “Value of information: Quantificationand application to coalition machine learning,” in *Policy-Based Autonomic Data Governance*, pp. 21–41, Springer, 2019.

H. Abdi and D. Valentin, “Multiple correspondence analysis,” *Encyclopedia of Measurement and Statistics*, vol. 2, no. 4, pp. 651–657, 2007.

H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisci-plinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, *et al.*, “Communication-efficient learning of deep networks from decentralized data,” *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

H. Jin, L. Su, D. C. K. Nahrstedt, and J. Xu, “Quality of information aware incentive mechanisms for mobile crowd sensing systems,” in *Proceedings of theACM International Symposium on Mobile Ad-Hoc Networking and Computing*, pp. 167–176, 2015.

H. Kargupta, W. Huang, K. Sivakumar, and E. Johnson, “Distributed clusteringusing collective principal component analysis,” *Knowledge and Information Systems*, vol. 3, no. 4, pp. 422–448, 2001.

H. Lu, M.-J. Li, T. He, S. Wang, V. Narayanan, and K. S. Chan, “Robust coreset construction for distributed machine learning,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, pp. 2400–2417, 2020.

H. Ludwig, N. Baracaldo, G. Thomas, Y. Zhou, A. Anwar, S. Rajamoni,

Hitaj, G. Ateniese, and F. Perez-Cruz, “Deep models under the GAN: information leakage from collaborative deep learning,” in *Proceedings of the2017 ACM SIGSAC Conference on Computer and Communications Security*,pp. 603–618, 2017.

J. D. Moffett and M. S. Sloman, “Policy conflict analysis in distributed systemmanagement,” *Journal of Organizational Computing and Electronic Com- merce*, vol. 4, no. 1, pp. 1–22, 1994.

J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

J. G. Kemeny and J. L. Snell, *Markov Chains*. Springer-Verlag, New York, 1976.

J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Ba- con, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, 2016.

J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, “Poisoning attack in federatedlearning using generative adversarial nets,” in *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communica- tions/13th IEEE International Conference on Big Data Science and Engineer-ing (TrustCom/BigDataSE)*, pp. 374–380, IEEE, 2019.

Journal of Artificial Intelligence Research, vol. 11, pp. 169–198, 1999.

K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, K. D. Harris and A. General, “California data breach report,” *Retrieved August*,vol. 7, p. 2016, 2016.

K. E. Schaefer, J. Oh, D. Aksaray, and D. Barber, “Integrating Context into Artificial Intelligence: Research from the Robotics Collaborative Technology Alliance,” *AI Magazine*, vol. 40, no. 3, pp. 28–40, 2019.

K. Grueneberg, S. Calo, P. Dewan, D. Verma, and T. O’Gorman, “A Policy-based Approach for Measuring Data Quality,” in *2019 IEEE International Conference on Big Data (Big Data)*, pp. 4025–4031, IEEE, 2019.

K. Sarpatwar, R. Vaculin, H. Min, G. Su, T. Heath, G. Ganapavarapu, and D. Dillenberger, “Towards enabling trusted artificial intelligence via blockchain,” in Policy-Based Autonomic Data Governance, pp. 137–153, Springer, 2019.

L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Com-puter Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

L. Guan, “An incremental updating algorithm of attribute reduction set in de-cision tables,” in *Sixth International Conference on Fuzzy Systems and Knowl-edge Discovery*, vol. 2, pp. 421–425, IEEE, 2009.

L. He, L.-d. WU, and Y.-c. CAI, “Survey of clustering algorithms in data mining,” *Application Research of Computers*, vol. 1, pp. 10–13, 2007.

L. Karadsheh, “Applying security policies and service level agreement to IaaSservice model to enhance security and transition,” *Computers & Security*, vol. 31, no. 3, pp. 315–326, 2012.

L. Rabiner and B. Juang, “An introduction to Hidden Markov Models,” *IEEEASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Pro- cessing*. Prentice Hall Press, 2010.

L. Rokach and O. Maimon, “Decision trees,” in *Data Mining and Knowledge Discovery Handbook*, pp. 165–192, Springer, 2005.

L. Rokach and O. Maimon, “Top-down induction of decision trees classifiers-a survey,” *IEEE*

Transactions on Systems, Man, and Cybernetics, Part C (Appli-cations and Reviews), vol. 35, no. 4, pp. 476–487, 2005.

L. Wu and R. Buyya, "Service Level Agreement (SLA) in utility computing systems," in *Performance and Dependability in Sservice Computing: Concepts, Techniques and Research directions*, pp. 1–25, IGI Global, 2012.

M. A. Nielsen, *Neural Networks and Deep Learning*, vol. 2018. Determination Press San Francisco, CA, 2015.

M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*,pp. 1322–1333, 2015.

M. J. Wolf, K. W. Miller, and F. S. Grodzinsky, "Why we should have seen thatcoming: comments on microsoft's tay "experiment," and wider implications,"*The ORBIT Journal*, vol. 1, no. 2, pp. 1–12, 2017.

M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, Elsevier, 1989.

M. Nixon and A. Aguado, *Feature Extraction and Image Processing for Com- puter Vision*. Academic Press, 2019.

M. Ramsay, O. Sankoh, as members of the AWI-Gen study, and the H3Africa Consortium, "African partnerships through the H3Africa Consor- tium bring a genomic dimension to longitudinal population studies on the continent," 2016.

M. V. Wickerhauser, *Adapted Wavelet Analysis: From Theory to Software*. CRC Press, 1996.

M.-F. F. Balcan, S. Ehrlich, and Y. Liang, "Distributed k-means and k-median clustering on general topologies," *Advances in Neural Information ProcessingSystems*, vol. 26, pp. 1995–2003, 2013.

M.-q. Hong, P.-Y. Wang, and W.-B. Zhao, "Homomorphic encryption scheme based on elliptic curve cryptography for privacy protection of cloud com- puting," in 2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Perfor- mance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), pp. 152–157, IEEE, 2016.

N. Bhagoji, S. Chakraborty, P. Mittal, and S. Calo, "Analyzing federated learning through an adversarial lens," in *International Conference on Machine Learning*, pp. 634–643, PMLR, 2019.

N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks onspeech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, IEEE, 2018.

N. Shan and W. Ziarko, "An incremental learning algorithm for constructingdecision rules," in *Rough Sets, Fuzzy Sets and Knowledge Discovery*, pp. 326–334, Springer, 1994.

Nedic and D. P. Bertsekas, "Incremental subgradient methods for non- differentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

O. Bachem, M. Lucic, and A. Krause, "Practical coreset constructions for machine learning,"

arXiv preprint arXiv:1703.06476, 2017.

O. Hall, N. Chawla, K. W. Bowyer, *et al.*, "Combining decision trees learned in parallel," in *Knowledge Discovery and Data Mining Workshop on Dis- tributed Data Mining*, pp. 10–15, 1998.

p. 114130A, International Society for Optics and Photonics, 2020.

P. E. Utgoff, "Incremental Induction of Decision Trees," *Machine Learning*, vol. 4, no. 2, pp. 161–186, 1989.

P. Paillier, "Public-key cryptosystems based on composite degree residuos- ity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, 1999.

P. R. Smart and K. P. Sycara, "Collective Sensemaking and Military Coali- tions," *IEEE Intelligent Systems*, vol. 28, no. 1, pp. 50–56, 2012.

P. Richhariya and P. K. Singh, "A survey on financial fraud detection method- ologies," *International Journal of Computer Applications*, vol. 45, no. 22,pp. 15–22, 2012.

P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*,2017.

Plachkinova and C. Maurer, "Security breach at target," *Journal of Infor-mation Systems Education*, vol. 29, no. 1, pp. 11–20, 2018.

Quinlan, "Generating production rules from decision trees," in *Proceedings of the 10th International Joint Conference on Artificial Intelligence*, pp. 304– 307, Morgan Kaufmann Publishers Inc., 1987.

R. Cramer, I. B. Damgard, , and J. B. Nielsen, *Secure Multiparty Computation*. Cambridge University Press, 2015.

R. I. Young, "Building principles for a quality of information specification for sensor information," in *2009 12th International Conference on Information Fusion*, pp. 1370–1377, IEEE, 2009.

R. Kohavi, "The power of decision tables," in *European Conference on Ma- chine Learning*, pp. 174–189, Springer, 1995.

R. R. Henning, "Security service level agreements: quantifiable security for theenterprise?," in *Proceedings of the 1999 Workshop on New Security Paradigms*,pp. 54–60, 1999.

R. Rathmell, "A Coalition Force Scenario'Binni-Gateway to the Golden Bowl of Africa'," in *Proceedings of the International Workshop on Knowledge-basedPlanning for Coalition Forces (ed. Tate, A.)*, pp. 115–125, 1999.

R. Yegireddi and R. K. Kumar, "A survey on conventional encryption algo- rithms of Cryptography," in 2016 International Conference on ICT in Business Industry & Government (ICTBIG), pp. 1–4, IEEE, 2016.

S. Calo, D. Verma, A. Schaeffer-Filho, *et al.*, "The coalition policy manage- ment portal for policy authoring, verification, and deployment," in *2008 IEEEWorkshop on Policies for Distributed Systems and Networks*, pp. 247–249, IEEE, 2008.

S. Chakraborty, C. Liu, and D. Verma, "Secure model fusion for distributed Learning using partial homomorphic encryption," in *Policies for Autonomic Data Governance at*

ESORICS, 2018.

S. E. Schaeffer, “Graph clustering,” *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.

S. KIKUCHI, Y. KANNA, and Y. ISOZAKI, “CIM Simplified Policy Lan- guage (CIM-SPL) CIM Simplified Policy Language (CIM-SPL), 2009,” *IEICE Transactions on Information and Systems*, vol. 95, no. 11, pp. 2634–2650, 2012.

S. Legg, M. Hutter, *et al.*, “A collection of definitions of intelligence,” in *Frontiers in Artificial Intelligence and Applications*, vol. 157, p. 17, IOS Press, 2007.

S. Ruping, “Incremental learning with support vector machines,” in *Proceed-ings 2001 IEEE International Conference on Data Mining*, pp. 641–642, IEEE, 2001.

S. S. Roy, F. Turan, K. Jarvinen, F. Vercauteren, and I. Verbauwhede, “FPGA- based high-performance parallel architecture for homomorphic computing on encrypted data,” in 2019 IEEE International Symposium on High Performance Computer Architecture (HPCA), pp. 387–398, IEEE, 2019.

S. Suthaharan, “Support vector machine,” in *Machine Learning Models and Algorithms for Big Data Classification*, pp. 207–235, Springer, 2016.

S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan,“When edge meets learning: Adaptive control for resource-constrained dis- tributed machine learning,” in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 63–71, IEEE, 2018.

S. Wolford, *The Politics of Military Coalitions*. Cambridge University Press,2015.

S.-L. Documentation, “Compare the effect of different scalers on data with outliers,” Nov 2020.

T. Bray (Ed.), “The JavaScript Object Notation (JSON) Data Interchange For-mat.” RFC 8259 (Internet Standard), Dec. 2017.

T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, *et al.*, “Extensible markup language (XML) 1.0,” 2000.

T. ElGamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4,pp. 469–472, 1985.

T. Hofmann, B. Schölkopf, and A. J. Smola, “Kernel methods in machine learning,” *The Annals of Statistics*, pp. 1171–1220, 2008.

T. Murata, “Petri Nets: Properties, Analysis and Applications,” *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

T. Nelson, *Mergers and Acquisitions from A to Z*. Amacom, 2018.

T. Pham, G. Cirincione, A. Swami, G. Pearson, and C. Williams, “Distributed Analytics and Information Sciences,” in *International Conference on Informa-tion Fusion*, pp. 245–252, IEEE, 2015.

T. Pham, G. H. Cirincione, D. Verma, and G. Pearson, “Intelligence, Surveil- lance and Reconnaissance fusion for Coalition Operations,” in *International Conference on Information Fusion*, pp. 1–8, IEEE, 2008.

T. Velte, A. Velte, and R. Elsenpeter, *Cloud Computing, A Practical Approach*. McGraw-Hill, Inc.,

2009.

V. G. da Silva, M. Kirikova, and G. Alksnis, “Containers for virtualization: An overview,” *Applied Computer Systems*, vol. 23, no. 1, pp. 21–27, 2018.

V. Mnih, N. Heess, A. Graves, *et al.*, “Recurrent models of visual attention,”in *Advances in Neural Information Processing Systems*, pp. 2204–2212, 2014.

V. Pimentel and B. G. Nickerson, “Communicating and displaying real-time data with websocket,” *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, 2012.

V. Sachidananda, A. Khelil, and N. Suri, “Quality of information in wirelesssensor networks: A survey,” in *Proceeding of the International Conference onInformation Quality*, 2010.

W. D. Nothwang, M. J. McCourt, R. M. Robinson, S. A. Burden, and J. W. Curtis, “The human should be part of the control loop?,” in *2016 Resilience Week (RWS)*, pp. 214–220, IEEE, 2016.

W. Han and C. Lei, “A survey on policy languages in network and security management,” *Computer Networks*, vol. 56, no. 1, pp. 477–489, 2012.

W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

W. J. Scheirer, L. P. Jain, and T. E. Boult, “Probability models for open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*,vol. 36, no. 11, pp. 2317–2324, 2014.

W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang,

X. Ning and G. Karypis, “SLIM: Sparse linear methods for top-n recom- mender systems,” in *2011 IEEE 11th International Conference on Data Min-ing*, pp. 497–506, IEEE, 2011.

Y. Kang, Y. Liu, and T. Chen, “FedMVT: Semi-supervised vertical federated learning with multiView training,” *IJCAI Workshop on Federated Learning for User Privacy and Data Confidentiality*, 2020.

Y. Liang, M.-F. F. Balcan, V. Kanchanapally, and D. Woodruff, “Improved distributed principal component analysis,” in *Advances in Neural Information Processing Systems*, pp. 3113–3121, 2014.

Y. Ong, J. Radhakrishnan, A. Verma, M. Sinn, *et al.*, “IBM federated learning: an enterprise framework white paper v0. 1,” *arXiv preprint arXiv:2007.10987*,2020.

Y. Song, T. Liu, T. Wei, X. Wang, Z. Tao, and M. Chen, “FDA3: federated defense against adversarial attacks for cloud-based iiot applications,” *IEEE Transactions on Industrial Informatics*, 2020.

Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “Fully homomorphic en- cryption without bootstrapping,” ACM Transactions on Computation Theory (TOCT), vol. 6, no. 3, pp. 1–36, 2014.

Z. Chen, P. Tian, W. Liao, and W. Yu, “Zero Knowledge Clustering Based Adversarial Mitigation in Heterogeneous Federated Learning,” *IEEE Transac-tions on Network Science and Engineering*, 2020.

Kecerdasan Buatan Gabungan pada Sistem Operasi Bisnis

Dr. Agus Wibowo, M.Kom, M.Si, MM.

BIO DATA PENULIS

Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
JL. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-623-8120-83-3 (PDF)
9 786238 120833