Faktor Exacta

Vol. 18, No. 2, June 2025, pp. 129~140
eISSN: 2502-339X, pISSN: 1979-276X, DOI: https://doi.org/10.30998/faktorexacta.v18i2.26078

Ketahanan Pembelajaran Mesin terhadap *Adversarial examples*: Metodologi dan Pertahanan

Ade Kurniawan, Ely Aprilia, Achmad Indra Aulia, Amril Mutoi Siregar, Leonard Goeirmanto
Informatika, Institut Teknologi Sains Bandung
Jl. Ganesha Boulevard No.1 Blok A, Kabupaten Bekasi, 17530, Jawa Barat, Indonesia.

Article Info

Article history:

Received Oct 17, 2024
Revised Mar 19, 2025
Accepted Aug 12, 2025

Keywords:

Pembelajaran Mesin
Adversarial examples
Serangan *Adversarial*
Pertahanan Model
Keamanan AI.

ABSTRACT

This paper examines the vulnerability of machine learning models to **adversarial examples**: inputs that are subtly manipulated to deceive a model into making incorrect predictions. Although deep learning has demonstrated remarkable performance across various tasks, the security of these models remains a significant challenge. This study provides a comprehensive review of various methods for generating adversarial examples, a classification of attack techniques, and corresponding defense strategies, including both active and passive approaches. The findings indicate that a combination of several defense techniques is significantly more effective in enhancing model robustness compared to any single approach. This research is expected to provide a foundation for the development of more secure and reliable machine learning models for critical applications.

Corresponding Author:

Ade Kurniawan,
Informatika,
Institut Teknologi Sains Bandung,
Jl. Ganesha Boulevard No.1 Blok A, Kabupaten Bekasi, 17530, Jawa Barat,
Email: ade.k@tsb.ac.id

1. PENDAHULUAN

Di era digital ini, algoritma pembelajaran mesin (*machine learning*) memainkan peran sentral dalam berbagai aplikasi, seperti kendaraan otonom dan sistem penerjemahan bahasa. Perkembangan dalam pembelajaran mendalam, terutama pada jaringan saraf konvolusional (CNN) dan arsitektur jaringan saraf lainnya, telah memperluas penggunaan teknologi ini secara signifikan, memungkinkan penerapannya pada tugas-tugas yang kompleks seperti klasifikasi gambar, deteksi objek, dan pengenalan suara [1][2][3]. Meskipun pencapaian ini luar biasa, masalah keamanan dan ketahanan sistem pembelajaran mesin masih sering kali kurang mendapat perhatian yang memadai.

Salah satu isu keamanan utama dalam pembelajaran mesin adalah kerentanannya terhadap *adversarial examples*. *Adversarial examples* adalah input yang sengaja dimodifikasi untuk mengecoh model pembelajaran mesin sehingga menghasilkan prediksi yang salah, meskipun input tersebut tampak benar di mata manusia. Penelitian awal oleh Szegedy et al. menunjukkan bahwa model jaringan saraf mendalam, meskipun sangat efektif dalam menyelesaikan berbagai tugas, sangat rentan terhadap serangan seperti ini [4]. Keberadaan *adversarial examples* menunjukkan adanya kesenjangan besar antara kemampuan generalisasi model pembelajaran mesin dengan persepsi manusia [5].

Prinsip dasar dari *adversarial examples* serupa dengan ilusi persepsi, di mana model kecerdasan buatan dapat "ditipu" oleh data yang telah dimodifikasi secara khusus. Seperti halnya ilusi optik yang dapat mengungkap bagaimana otak manusia bekerja, *adversarial examples* dapat membantu kita memahami cara kerja model pembelajaran mesin serta menemukan kelemahan-kelemahan dalam model tersebut [6]. Banyak teknik yang digunakan untuk menciptakan *adversarial examples* memanfaatkan sifat linear dari model-model ini, menyebabkan mereka salah dalam mengklasifikasikan input yang hanya dimodifikasi sedikit halus [5][7].

Makalah ini bertujuan memberikan pemahaman komprehensif mengenai *adversarial examples*, mulai dari mekanisme serangan hingga metode pertahanan yang telah dikembangkan. Secara khusus, makalah ini akan membahas:

1) Dasar-dasar pembelajaran mesin: Konsep fundamental pembelajaran mesin yang relevan untuk memahami *adversarial examples*.
2) Mekanisme serangan: Penjelasan mengenai bagaimana *adversarial examples* diciptakan dan mengapa model pembelajaran mesin rentan terhadap serangan ini.
3) Jenis-jenis serangan: Klasifikasi dan penjelasan berbagai teknik serangan, dari yang sederhana hingga yang lebih canggih.
4) Pertahanan: Berbagai metode pertahanan terhadap serangan adversarial, termasuk pendekatan pasif dan aktif.
5) Evaluasi perbandingan: Perbandingan berbagai teknik pertahanan untuk mengidentifikasi kelebihan dan kekurangannya.
6) Tantangan dan masa depan: Tantangan yang masih dihadapi serta arah penelitian masa depan untuk menciptakan sistem pembelajaran mesin yang lebih aman dan andal.
7) Dengan pemahaman yang lebih dalam terhadap ancaman dan mekanisme pertahanan, penelitian ini diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan keamanan dan ketahanan sistem pembelajaran mesin.

Makalah ini membahas pembelajaran mesin dan deep learning, pembuatan serta analisis adversarial examples, klasifikasi serangan dan studi kasus, strategi pertahanan, serta tantangan dan arah pengembangan ke depan. Kesimpulan merangkum temuan utama dan rekomendasi penelitian lanjutan.

2. Dasar-dasar Pembelajaran Mesin

Bagian ini menjelaskan konsep fundamental dalam pembelajaran mesin dan *deep learning*, yang menjadi landasan untuk memahami teknik-teknik canggih dalam analisis data dan prediksi model. Dimulai dengan pengenalan prinsip dasar pembelajaran mesin, bab ini membahas bagaimana mesin dapat belajar dari data untuk membuat keputusan atau prediksi. Selanjutnya, dibahas arsitektur jaringan saraf tiruan, termasuk struktur dan komponen utama yang mendukung pembelajaran mendalam, seperti jaringan saraf konvolusional (CNN) dan jenis arsitektur mendalam lainnya. Selain itu, fungsi aktivasi dan fungsi loss juga dijelaskan, menguraikan peran penting keduanya dalam memengaruhi output model dan menilai kinerja prediksi, yang merupakan bagian integral dari proses pelatihan model. Pada akhirnya, bab ini menguraikan tahapan pelatihan model, mulai dari pemilihan data hingga optimasi dan evaluasi. Pemahaman yang dibangun di bab ini akan memberikan fondasi yang kuat untuk mendalami teknik lebih lanjut serta tantangan dalam pembelajaran mesin dan deep learning.

2.1. Konsep Dasar Pembelajaran Mesin

Pembelajaran mesin (ML) adalah cabang kecerdasan buatan (AI) yang memungkinkan sistem belajar dari data untuk meningkatkan kinerjanya secara otomatis, tanpa perlu intervensi manusia langsung. Algoritma pembelajaran mesin digunakan untuk menganalisis data, menemukan pola, dan membuat keputusan berdasarkan data tersebut. Terdapat tiga pendekatan utama dalam pembelajaran mesin, yaitu *supervised learning*, *unsupervised learning*, dan *reinforcement learning*.

Dalam supervised learning, model dilatih dengan data yang sudah diberi label. Tujuan utamanya adalah menemukan hubungan antara input dan output, sehingga model dapat membuat prediksi yang akurat untuk data baru yang belum pernah dilihat sebelumnya. Beberapa teknik populer dalam pendekatan ini termasuk regresi linier, pohon keputusan, dan SVM [8][9].

Sebaliknya, *unsupervised learning* bekerja dengan data yang tidak memiliki label, dan bertujuan untuk menemukan pola tersembunyi atau struktur dalam data. Algoritma seperti klasterisasi (contoh: K-means) dan pengurangan dimensi (contoh: PCA - *Principal Component Analysis*) sering digunakan dalam pendekatan ini [10].

Sementara itu, *reinforcement learning* melibatkan agen yang belajar mengambil keputusan dengan mencoba berbagai tindakan dan menerima umpan balik dalam bentuk penghargaan (*reward*) atau hukuman (*punishment*). Pendekatan ini banyak digunakan dalam aplikasi yang memerlukan pengambilan keputusan bertahap, seperti permainan dan navigasi robot [11].

2.2. Arsitektur Jaringan Saraf Tiruan

Jaringan saraf tiruan (*artificial neural networks*, ANN) adalah salah satu model paling kuat dalam pembelajaran mesin, terutama dalam konteks pembelajaran mendalam (*deep learning*). Jaringan ini terinspirasi oleh struktur otak manusia dan terdiri dari beberapa lapisan neuron yang saling terhubung. Tiga jenis lapisan utama dalam jaringan saraf tiruan adalah lapisan input, lapisan tersembunyi, dan lapisan output.

Lapisan input menerima data mentah dan mengirimkannya ke lapisan tersembunyi, yang kemudian memproses data tersebut dengan menerapkan fungsi aktivasi untuk menangkap pola non-linear. Lapisan output menghasilkan hasil akhir, berupa prediksi atau klasifikasi [12][13].

Jaringan saraf yang lebih kompleks, seperti jaringan saraf konvolusional (CNN), dirancang khusus untuk memproses data visual, seperti gambar. CNN menggunakan lapisan konvolusi untuk mengekstraksi fitur-fitur penting dari gambar dengan menerapkan filter atau kernel pada input gambar input [14]. Sementara itu, jaringan saraf berulang (RNN) cocok untuk data berurutan, seperti teks atau sinyal waktu, karena memiliki kemampuan untuk mempertahankan informasi dari langkah-langkah sebelumnya [15].

2.3. Fungsi Aktivasi dan Fungsi Loss

Fungsi aktivasi adalah komponen penting dalam jaringan saraf, karena mereka memperkenalkan non-linearitas ke dalam model, memungkinkan model untuk mempelajari hubungan yang lebih kompleks antara input dan output. Beberapa fungsi aktivasi yang populer adalah:

- ReLU (*Rectified Linear Unit*): Fungsi ini menghasilkan output yang sama dengan input jika nilainya lebih besar dari nol, dan nol jika kurang dari nol. ReLU populer karena kemampuannya mengatasi masalah *vanishing gradient* [2].
- Sigmoid: Fungsi ini menghasilkan output dalam rentang 0 hingga 1, dan sering digunakan dalam masalah klasifikasi biner.
- Tanh (*Hyperbolic Tangent*): Serupa dengan Sigmoid, tetapi menghasilkan output dalam rentang -1 hingga 1, yang berguna untuk proses pelatihan karena memberikan gradien yang lebih besar [16].

Fungsi loss mengukur seberapa baik model memprediksi hasil yang benar. Selama pelatihan, fungsi ini digunakan untuk mengoptimalkan model dengan cara meminimalkan nilai loss. Beberapa contoh fungsi loss termasuk:

- *Mean Squared Error* (MSE): Digunakan untuk masalah regresi, mengukur rata-rata kuadrat selisih antara nilai prediksi dan nilai sebenarnya.
- *Cross-Entropy*: Digunakan dalam masalah klasifikasi, mengukur seberapa baik distribusi probabilitas yang diprediksi sesuai dengan distribusi probabilitas yang sebenarnya [17] [18].

2.4. Proses Pelatihan Model

Proses pelatihan model melibatkan beberapa langkah penting untuk meningkatkan kinerja model. Pertama, data biasanya dibagi menjadi tiga bagian: data pelatihan, data validasi, dan data pengujian. Data pelatihan digunakan untuk melatih model, data validasi untuk menyesuaikan *hyperparameter* dan menghindari *overfitting*, sementara data pengujian digunakan untuk mengevaluasi kinerja akhir model.

Selama pelatihan, algoritma optimasi seperti *Gradient Descent* digunakan untuk memperbarui bobot model dengan tujuan meminimalkan fungsi loss. Varian dari *Gradient Descent* seperti *Stochastic Gradient Descent* (SGD) dan Adam memiliki keunggulan tersendiri; Adam, misalnya, menggabungkan keuntungan dari momentum dan penyesuaian learning rate [19] .

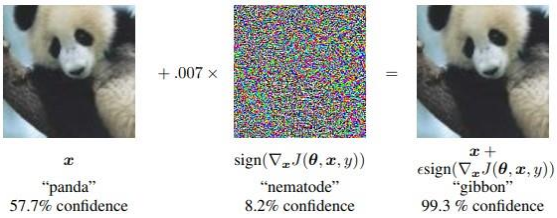
Setelah pelatihan selesai, evaluasi model dilakukan untuk memastikan bahwa model dapat melakukan generalisasi dengan baik pada data baru. Metrik kinerja yang sering digunakan meliputi akurasi, presisi, *recall*, dan F1-score, tergantung pada jenis masalah yang dihadapi. Proses pelatihan dan evaluasi ini bersifat iteratif dan memerlukan penyesuaian *hyperparameter* untuk mencapai hasil yang optimal [9] [20].

3. Mekanisme Serangan Adversarial

Bab ini membahas secara rinci bagaimana *adversarial examples* dibuat, divisualisasikan, dan dianalisis secara matematis dalam konteks pembelajaran mesin dan *deep learning*. Bagian ini sangat penting untuk memahami kerentanan model pembelajaran mesin serta bagaimana serangan *adversarial* dapat dikenali dan dicegah.

3.1. Cara Pembuatan *Adversarial examples*

1) Metode Berbasis Gradien: Teknik ini menggunakan informasi dari gradien fungsi loss terhadap input untuk membuat perubahan kecil yang menipu model. Salah satu metode yang paling terkenal adalah *Fast Gradient Sign Method* (FGSM), yang diperkenalkan oleh Goodfellow et al. FGSM menghitung gradien fungsi loss terhadap input asli dan kemudian menambahkan gangguan kecil ke arah gradien tersebut untuk menghasilkan *adversarial examples* [21]. Meskipun sederhana, metode ini sangat efektif dalam membuat contoh yang dapat menipu model, seperti yang terlihat pada Gambar 1.



Gambar 1. *Adversarial examples* menggunakan FGSM

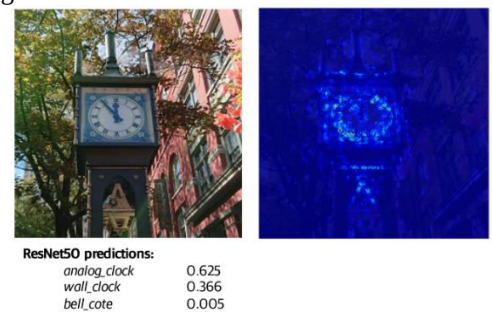
2) Serangan Berbasis Optimasi: Teknik ini memformulasikan pembuatan *adversarial examples* sebagai masalah optimasi, di mana tujuannya adalah memaksimalkan fungsi loss dengan pembatasan bahwa perubahan pada input harus tetap kecil. Salah satu teknik yang terkenal adalah *Projected Gradient Descent* (PGD), di mana optimasi dilakukan secara iteratif dengan memperbarui input ke arah gradien loss dan kemudian memproyeksikannya kembali ke dalam batas gangguan yang diperbolehkan [22]. Meskipun lebih kompleks daripada FGSM, teknik ini sering kali lebih kuat.

3) Serangan *Black-Box*: Serangan ini tidak memerlukan akses langsung terhadap model atau gradiennya. Teknik ini menggunakan pendekatan heuristik atau berbasis kueri untuk membuat *adversarial examples*. Beberapa teknik termasuk serangan berbasis evolusi, di mana populasi contoh diubah dan dievaluasi secara iteratif, serta serangan berbasis kueri yang menggunakan umpan balik dari model untuk memandu modifikasi input [23].

3.2. Visualisasi *Adversarial examples*

Visualisasi *adversarial examples* sangat penting untuk memahami bagaimana modifikasi kecil pada input dapat mempengaruhi prediksi model. Teknik visualisasi ini membantu mengidentifikasi pola-pola atau karakteristik spesifik yang dieksploitasi oleh serangan. Berikut adalah beberapa teknik visualisasi yang umum digunakan:

1) Visualisasi Gangguan (*Perturbation Visualization*): Teknik ini menampilkan perbedaan antara input asli dan *adversarial examples*, sehingga kita dapat melihat seberapa kecil gangguan yang diperlukan untuk menipu model. Misalnya, perbedaan antar-piksel dapat divisualisasikan untuk menunjukkan pola gangguan yang ditambahkan [7].
2) Peta *Saliency* (*Saliency Maps*): Teknik ini menyoroti area input yang paling mempengaruhi keputusan model. Dengan peta *saliency*, kita dapat mengidentifikasi bagian mana dari input yang paling rentan terhadap serangan. Misalnya, dalam klasifikasi gambar, peta *saliency* menunjukkan area yang paling memengaruhi prediksi model [24]. Pada Gambar 2, bagian kiri menunjukkan gambar asli jam di jalan kota, sementara bagian kanan adalah peta *saliency* yang dihasilkan oleh model ResNet50. Area di sekitar wajah jam (terutama bagian tengah) memiliki pengaruh terbesar pada prediksi model, sehingga lebih rentan terhadap serangan *adversarial*.



Gambar 2. Contoh penggunaan *Saliency Map*

3) Reduksi Dimensi: Teknik ini menggunakan metode seperti PCA atau t-SNE untuk memvisualisasikan distribusi data asli dan *adversarial examples* dalam ruang dimensi rendah. Dengan cara ini, kita bisa melihat bagaimana distribusi data berubah setelah gangguan adversarial ditambahkan, sehingga membantu kita memahami penyebaran *adversarial examples* dalam ruang fitur model [26].

3.3. Analisis Matematis di Balik *Adversarial examples*

Analisis matematis membantu memberikan wawasan mendalam tentang mengapa dan bagaimana *adversarial examples* dapat menipu model pembelajaran mesin. Pemahaman ini penting untuk

mengembangkan strategi pertahanan yang efektif. Beberapa konsep kunci dalam analisis matematis *adversarial examples* adalah:

1) Asumsi Linearitas (*Linearization Assumption*): Goodfellow et al. menjelaskan bahwa jaringan saraf cenderung berperilaku hampir linear pada skala lokal. Ini berarti bahwa meskipun jaringan saraf adalah fungsi non-linear yang kompleks, pada skala kecil, model dapat didekati sebagai fungsi linear. Sifat linear ini sering kali dieksploitasi oleh serangan *adversarial* untuk menipu model [5].
2) Batasan Ketahanan (*Robustness Bounds*): Beberapa penelitian telah mengembangkan batas teoretis tentang seberapa besar gangguan yang dapat ditoleransi oleh model tanpa memicu salah klasifikasi. Konsep ini membantu memahami sejauh mana model dapat menahan serangan adversarial. Salah satu pendekatan yang digunakan adalah konsep margin klasifikasi dan radius *robust* [27].
3) Analisis Sensitivitas: Teknik ini mengukur sensitivitas model terhadap perubahan kecil pada input, yang dapat membantu dalam memahami kerentanannya terhadap serangan adversarial. Analisis ini melibatkan gradien dan Hessian untuk melihat bagaimana output model berubah sebagai respons terhadap gangguan kecil pada input [4].

4. Jenis-Jenis Serangan *Adversarial*

Bab ini menguraikan berbagai jenis serangan *adversarial* yang dapat dilakukan terhadap model pembelajaran mesin, mengklasifikasikannya berdasarkan target serangan dan teknik yang digunakan, serta memberikan studi kasus untuk menggambarkan implementasi dan dampak dari serangan-serangan tersebut

4.1 Klasifikasi Serangan Berdasarkan Target

Serangan *adversarial* dapat dibedakan berdasarkan target yang ingin dicapai oleh penyerang. Dua kategori utama dalam klasifikasi ini adalah:

1) Serangan Terarah (*Targeted Attacks*): Serangan terarah bertujuan untuk memaksa model memberikan prediksi yang spesifik. Contohnya, sebuah *adversarial examples* bisa dirancang untuk membuat model klasifikasi gambar mengenali gambar anjing sebagai kucing. Serangan ini membutuhkan penyesuaian yang tepat agar model salah klasifikasi ke target yang diinginkan. Keberhasilan serangan ini menunjukkan bahwa model sangat rentan terhadap perubahan kecil pada input yang dirancang khusus untuk menipu [28].
2) Serangan Tidak Terarah (*Untargeted Attacks*): Tujuan dari serangan ini adalah membuat model memberikan prediksi yang salah tanpa memedulikan kategori spesifik mana yang dipilih oleh model. Misalnya, membuat model mengklasifikasikan gambar anjing sebagai objek apa pun selain anjing. Serangan ini biasanya lebih mudah dilakukan dibandingkan serangan terarah karena hanya bertujuan untuk menyebabkan kesalahan tanpa menentukan hasil prediksi yang salah tersebut [5].

4.2 Klasifikasi Serangan Berdasarkan Teknik

Serangan *adversarial* juga dapat diklasifikasikan berdasarkan teknik yang digunakan untuk menghasilkan *adversarial examples*. Beberapa teknik utama yang sering digunakan meliputi:

1) FGSM adalah salah satu teknik yang digunakan untuk menghasilkan *adversarial examples* dengan cepat dan efisien. Metode ini memanfaatkan gradien dari fungsi loss terhadap input untuk menambahkan gangguan kecil pada input tersebut. FGSM mengeksploitasi sifat linear dari model pembelajaran mesin untuk menciptakan gangguan yang efektif dan sering digunakan dalam penelitian sebagai metode dasar untuk mengevaluasi kerentanan model terhadap serangan *adversarial* [29].
   Metode FGSM bekerja dengan menghitung gradien dari fungsi loss $\mathcal{L}$ terhadap input x, kemudian membuat gangguan kecil dengan mengalikan gradien ini dengan konstanta kecil ϵ dan mengambil tanda (*sign*) dari vektor gradien tersebut. Secara matematis, FGSM dapat dirumuskan sebagai berikut:

$$x' = x + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(x, y)) \quad (1)$$

di mana:

- x adalah input asli.
- x' adalah input yang telah ditambahkan gangguan (*adversarial examples*).
- ϵ adalah konstanta kecil yang mengontrol besarnya gangguan.
- $\nabla_x \mathcal{L}(x, y)$ adalah gradien dari fungsi loss $\mathcal{L}$ terhadap input x.
- sign}(·) adalah fungsi tanda yang menghasilkan +1 atau -1 tergantung pada tanda dari elemen-elemen dalam vektor gradien.

Penjelasan Notasi

- Input x: Data yang diberikan ke model untuk menghasilkan prediksi. Misalnya, pada model klasifikasi gambar, x adalah gambar asli yang akan diklasifikasikan.
- Fungsi Loss $\mathcal{L}$: Fungsi yang mengukur seberapa baik atau buruk prediksi model dibandingkan dengan label sebenarnya y. Fungsi loss umum meliputi *cross-entropy loss* untuk klasifikasi.

- Gradien $\nabla_x \mathcal{L}(x, y)$: Vektor yang menunjukkan arah dan kecepatan perubahan fungsi loss terhadap perubahan input x. Gradien ini dihitung menggunakan algoritma *backpropagation* dalam jaringan saraf dalam.
- Konstanta ϵ: Nilai kecil yang mengontrol seberapa besar gangguan yang ditambahkan ke input asli. Nilai ϵ harus cukup kecil untuk memastikan gangguan tidak terdeteksi oleh manusia, tetapi cukup besar untuk menipu model pembelajaran mesin.
- Fungsi Tanda sign}(.): Fungsi yang menghasilkan +1 atau -1 untuk setiap elemen dalam vektor gradien, tergantung pada apakah elemen tersebut positif atau negatif.

Dalam praktiknya, FGSM mengasumsikan bahwa gangguan yang dihasilkan akan memperbesar nilai loss dan membuat model salah mengklasifikasikan input yang telah dimodifikasi. Meskipun tidak ada jaminan bahwa setiap peningkatan nilai loss akan mengakibatkan salah klasifikasi, arah ini dianggap masuk akal karena nilai loss untuk instance yang salah klasifikasi secara definisi lebih besar.

2) *Projected Gradient Descent* (PGD) adalah teknik yang merupakan perpanjangan dari FGSM. Teknik ini melibatkan beberapa iterasi untuk memperbarui input berdasarkan gradien loss. PGD melakukan optimasi berulang kali dan memastikan gangguan tetap dalam batas yang diizinkan dengan memproyeksikannya kembali ke dalam ruang gangguan yang diperbolehkan setelah setiap iterasi. PGD dikenal sebagai salah satu metode paling kuat untuk menghasilkan *adversarial examples* [22].

PGD bekerja dengan mengaplikasikan FGSM secara iteratif dalam batas supremum-norm pada total gangguan input. *Adversarial examples* yang dihasilkan oleh PGD dapat dirumuskan sebagai berikut:

$$x'_0 = x \tag{2}$$

$$x'_{k+1} = \text{Clip}_{x,\epsilon}\{x'_k + \alpha \cdot \text{sign}(\nabla_x \mathcal{L}(x'_k, y))\} \tag{3}$$

Di mana:

- x'_0 adalah input awal (input asli)
- x'_k adalah input yang diperbarui pada iterasi ke-k
- α adalah ukuran langkah per iterasi
- $\nabla_x \mathcal{L}(x'_k, y)$ adalah gradien dari fungsi loss $\mathcal{L}$ terhadap input x'_k
- $\text{Clip}_{x,\epsilon}(\cdot)$ adalah operator clipping yang memastikan gangguan berada dalam batas ϵ-neighborhood dari input asli x

Operator clipping $\text{Clip}_{x,\epsilon}(\cdot)$ didefinisikan sebagai:

$$\text{Clip}_{x,\epsilon}(x') = min(max(x, x' - \epsilon), x + \epsilon) \tag{4}$$

Ini memastikan bahwa setiap fitur input (misalnya, piksel) pada koordinat tertentu tetap dalam ϵ-*neighborhood* dari instance asli x, serta dalam ruang input yang layak (misalnya, [0, 255] untuk nilai intensitas 8-bit).

Penjelasan Notasi:

- Input x: Data asli yang diberikan ke model untuk menghasilkan prediksi.
- Fungsi Loss $\mathcal{L}(x, y)$: Fungsi yang mengukur seberapa baik atau buruk prediksi model dibandingkan dengan label sebenarnya y. Contoh umum fungsi loss adalah *cross-entropy loss.*
- Gradien $\nabla_x \mathcal{L}(x, y)$: Vektor yang menunjukkan arah dan kecepatan perubahan fungsi loss terhadap perubahan input x. Gradien ini dihitung menggunakan algoritma *backpropagation* dalam jaringan saraf dalam.
- Ukuran Langkah α: Nilai yang mengontrol seberapa besar perubahan yang dibuat pada setiap iterasi.
- Operator Clipping $\text{Clip}_{x,\epsilon}(\cdot)$: Operator yang memastikan bahwa gangguan tetap dalam batas yang diizinkan. Ini dilakukan dengan membatasi nilai gangguan dalam ϵ-*neighborhood* dari input asli x dan memastikan bahwa nilai gangguan berada dalam rentang input yang valid (misalnya, [0, 255] untuk gambar 8-bit).

PGD dikenal karena efektivitasnya dalam menghasilkan *adversarial examples* yang kuat dan sering digunakan untuk menguji ketahanan berbagai model pembelajaran mesin, termasuk model klasifikasi dan segmentasi semantik. *Adversarial examples* yang dihasilkan oleh PGD telah digunakan untuk menyerang model-model seperti *Fully Convolutional Networks* (FCN) untuk segmentasi semantik.

PGD tidak hanya efektif dalam menghasilkan *adversarial examples* yang dapat mengecoh model pembelajaran mesin dalam lingkungan digital, tetapi juga dalam kondisi dunia nyata, seperti yang ditunjukkan oleh penggunaan *adversarial examples* yang dicetak pada kertas. Dengan menggunakan PGD, peneliti dan praktisi dapat mengidentifikasi kelemahan model pembelajaran mesin mereka dan mengembangkan metode pertahanan yang lebih kuat untuk melindungi model dari serangan tersebut.

3) *Black-Box Attacks*: Serangan ini tidak memerlukan akses langsung ke model atau gradiennya. Metode ini menggunakan pendekatan heuristik atau berbasis *query* untuk menghasilkan *adversarial examples*. Contoh teknik ini adalah serangan berbasis evolusi dan serangan berbasis query yang menggunakan umpan balik dari model untuk mengarahkan modifikasi input. *Black-box attacks* menunjukkan bahwa model dapat rentan bahkan ketika penyerang tidak memiliki informasi lengkap tentang model tersebut [23].

1) Carlini & Wagner (C&W) *Attacks* [7]: C&W menggunakan optimasi yang lebih kompleks untuk menghasilkan gangguan yang sulit dideteksi oleh metode pertahanan biasa. Serangan ini dikenal sangat efektif dan sering digunakan sebagai tolok ukur dalam menguji ketahanan model terhadap serangan *adversarial*. Dengan menggunakan optimasi yang canggih, penyerang dapat menghasilkan gangguan yang sangat efisien dan hampir tidak terlihat.

C&W *attacks* memperkenalkan keluarga serangan untuk menemukan gangguan *adversarial* yang meminimalkan berbagai metrik kesamaan: L_0, L_2, dan L_∞. Inti dari serangan ini adalah mengubah strategi optimasi terbatas umum menjadi fungsi loss yang dipilih secara empiris dalam formulasi optimasi tak terbatas.

Fungsi objektif utama yang digunakan dalam serangan ini adalah:

$$L(x', t) = max\left(0, Z(x')_t - \max_{i \neq t} Z(x')_i + \kappa\right) \quad (5)$$

di mana:

- $Z(x')_i$ adalah komponen ke-i dari logit classifier.
- t adalah label target.
- κ adalah parameter yang mencerminkan margin kepercayaan minimum yang di-inginkan untuk *adversarial examples*.

Fungsi loss ini meminimalkan jarak nilai logit antara kelas target t dan kelas kedua yang paling mungkin. Jika t memiliki nilai logit tertinggi, maka perbedaan nilai logit akan negatif, dan optimasi akan berhenti ketika perbedaan logit antara t dan kelas kedua melampaui κ. Jika t tidak memiliki nilai logit tertinggi, maka meminimalkan $L(x', t)$ akan mendekatkan jarak antara logit kelas tertinggi dan logit kelas target.

- Serangan L_2, C&W

Formulasi serangan adalah sebagai berikut:

$$\min_w \left\|\frac{1}{2}(\tanh(w) + 1) - x\right\|_2^2 + c \cdot L\left(\frac{1}{2}(\tanh(w) + 1), t\right) \quad (6)$$

di mana:

w adalah variabel perubahan.

$\frac{1}{2}(\tanh(w) + 1)$ untuk memastikan bahwa (x') berada dalam batas [0,1].

c adalah parameter yang dipilih melalui prosedur optimasi eksternal untuk menyeimbangkan kedua komponen fungsi objektif.

- Serangan L_0, C&W

Serangan L_0, lebih kompleks karena metrik jarak yang terkait tidak dapat didiferensiasikan. Strategi iteratif digunakan untuk secara bertahap menghilangkan fitur input yang tidak signifikan sehingga misclassification dapat dicapai dengan mengganggu sesedikit mungkin nilai input.

Serangan L_∞. C&W

Formulasi optimasi untuk varian L_∞. adalah:

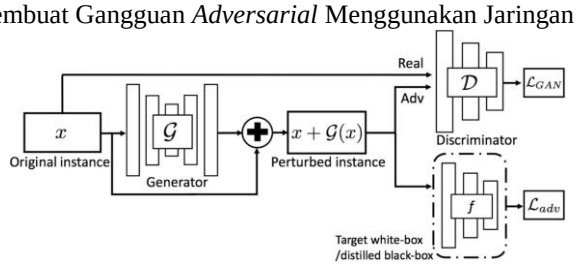
$$\min_r \| r \|_\infty \text{ dengan syarat } f(x + r) = t \quad (7)$$

Parameter τ diinisialisasi ke 1, dan dikurangi setiap iterasi sebesar faktor 0.9 jika $\|r(i)\| < \tau$ untuk semua i, hingga tidak ditemukan *adversarial examples*.

C&W attacks secara empiris menunjukkan bahwa metode mereka lebih unggul dibandingkan serangan lain saat diuji pada dataset MNIST, CIFAR10, dan ImageNet. Serangan L_0,

dibandingkan dengan JSMA, varian $L_{2,}$ dibandingkan dengan DeepFool, dan metode L_∞ dibandingkan dengan FGSM dan BIM. Hasilnya menunjukkan bahwa serangan C&W consistently outperforming dalam hal distorsi rata-rata dan tingkat keberhasilan serangan. Mereka juga menunjukkan bahwa serangan mereka berhasil melewati defensive distillation dengan tingkat keberhasilan 100%, sementara *adversarial examples* tetap mirip dengan input asli menurut metrik L_0, $L_{2,}$ dan L_∞.

4) AdvGAN [30]: Membuat Gangguan *Adversarial* Menggunakan Jaringan Saraf Tiruan.



Gambar 1. Ilustrasi dari AdvGAN [30]

AdvGAN adalah metode yang menggunakan jaringan saraf tiruan untuk menghasilkan gangguan *adversarial*, mirip dengan *pertubation adversarial transformation networks* (P-ATN) [31].
AdvGAN seperti di ilsutrasikan pada gambar 1, generator G menghasilkan gangguan r yang, ketika ditambahkan ke input asli x, menghasilkan *adversarial examples* x'. Perbedaan utama antara AdvGAN dan P-ATN adalah AdvGAN menambahkan jaringan diskriminator D dan melatih generator secara *adversarial* dalam kerangka kerja Generative *Adversarial* Network (GAN).
Fungsi objektif untuk generator dalam AdvGAN terdiri dari beberapa komponen, termasuk loss klasifikasi $\mathcal{L}_{\text{adv}}$, loss GAN $\mathcal{L}_{\text{GAN}}$, dan loss hinge $\mathcal{L}_{\text{hinge}}$:

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \alpha\mathcal{L}_{\text{GAN}} + \beta\mathcal{L}_{\text{hinge}} \quad (8)$$

di mana:

- $\mathcal{L}_{\text{adv}}$ adalah loss klasifikas
- $\mathcal{L}_{\text{GAN}}$ adalah loss GAN.
- $\mathcal{L}_{\text{hinge}}$ adalah loss hinge.
- α adalah konstanta yang mengontrol pembobotan dari masing-masing komponen loss.

Komponen Fungsi Objektif
1. Loss Klasifikasi:

$$\mathcal{L}_{\text{adv}} = \mathbb{E}_x\left[\mathcal{L}_f(x + G(x), t)\right] \quad (9)$$

Di sini, $\mathcal{L}_f$ adalah fungsi loss klasifikasi yang digunakan oleh model f, seperti cross-entropy loss, dan *t* adalah label salah sasaran.
2. Loss GAN $\mathcal{L}_{\text{GAN}}$:

$$\mathcal{L} = \mathcal{L}_{\text{adv}} + \alpha\mathcal{L}_{\text{GAN}} + \beta\mathcal{L}_{\text{hinge}} \quad (10)$$

Loss ini mengukur seberapa baik diskriminator *D* dapat membedakan antara data asli *x* dan *adversarial examples* $x + G(x)$.
3. Loss Hinge $\mathcal{L}_{\text{hinge}}$

$$\mathcal{L}_{\text{hinge}} = \mathbb{E}_x[max(0, \| G(x) \|_2 - c)] \quad (11)$$

Loss ini memastikan bahwa gangguan $r = G(x)$ tetap dalam batas yang diizinkan, di mana c adalah konstanta yang mewakili batas untuk loss hinge.
Metode AdvGAN terbukti lebih efektif daripada FGSM dalam menghasilkan contoh-contoh adversarial dan C&W saat diuji terhadap model yang dilatih dengan pertahanan *adversarial*. AdvGAN mampu menghasilkan *adversarial examples* yang tidak dapat dibedakan secara persepsi dari gambar asli oleh manusia, sehingga membuatnya lebih sulit untuk dideteksi.
Namun, perlu dicatat bahwa dalam eksperimen awal, hanya serangan FGSM yang dimasukkan dalam setup pelatihan *adversarial*. Oleh karena itu, meskipun AdvGAN berhasil menemukan *adversarial* yang mengatasi metode pertahanan FGSM dan C&W, hasil ini mungkin tidak selalu berlaku untuk setup lain yang melibatkan metode serangan dan pertahanan *adversarial* yang berbeda.

4.3 Studi Kasus Serangan Adversarial

Untuk memahami dampak nyata dari serangan *adversarial*, bab ini juga menyajikan beberapa studi kasus dari serangan yang telah dilakukan di dunia nyata:

1) Serangan pada Sistem Klasifikasi Gambar: Salah satu studi kasus yang terkenal adalah serangan terhadap sistem klasifikasi gambar seperti yang digunakan dalam kompetisi ImageNet. Dalam penelitian yang dilakukan oleh Szegedy et al., ditemukan bahwa menambahkan gangguan yang hampir tidak terlihat pada gambar dapat membuat model salah klasifikasi dengan akurasi tinggi. Studi ini menunjukkan bahwa bahkan gangguan kecil yang tidak terlihat oleh manusia dapat menipu model secara efektif [4].
2) Serangan pada Kendaraan Otonom: Serangan adversarial juga telah diuji pada sistem pengenalan gambar di kendaraan otonom. Misalnya, gangguan kecil yang ditambahkan pada gambar rambu lalu lintas dapat membuat sistem menginterpretasikan rambu tersebut secara salah, yang bisa mengakibatkan kecelakaan serius. Penelitian ini menyoroti potensi risiko keamanan yang signifikan dalam aplikasi kritis seperti kendaraan otonom [32].
3) Serangan pada Sistem Pengenalan Wajah: Sistem pengenalan wajah yang digunakan dalam keamanan dan autentikasi juga rentan terhadap serangan adversarial. Penelitian menunjukkan bahwa dengan menambahkan gangguan kecil pada gambar wajah, sistem pengenalan bisa dibuat untuk salah mengenali identitas seseorang. Hal ini menunjukkan bahwa serangan adversarial dapat mengancam keamanan sistem yang bergantung pada pengenalan biometrik [33]

Bab 4 ini memberikan wawasan mendalam tentang berbagai jenis serangan *adversarial* yang dapat menargetkan model pembelajaran mesin. Dengan memahami klasifikasi dan teknik yang digunakan dalam serangan-serangan ini, kita dapat lebih siap dalam mengembangkan metode pertahanan yang efektif untuk melindungi sistem pembelajaran mesin dari ancaman-ancaman tersebut

5. Pertahanan Terhadap Serangan Adversarial

Bab ini membahas berbagai pendekatan untuk melindungi model pembelajaran mesin dari serangan adversarial. Pendekatan pertahanan dibagi menjadi dua kategori utama: teknik pertahanan pasif dan aktif. Selain itu, evaluasi komprehensif dari berbagai teknik ini juga disajikan untuk memberikan gambaran mengenai efektivitas dan keterbatasan masing-masing metode.

5.1. Teknik Pertahanan Pasif

Teknik pertahanan pasif berfokus pada penguatan model agar lebih tahan terhadap serangan adversarial tanpa secara eksplisit mendeteksi serangan tersebut. Beberapa metode yang termasuk dalam kategori ini antara lain:

1) Regularisasi *Adversarial* (*Adversarial Training*): Metode ini melibatkan pelatihan model menggunakan *adversarial examples* yang dihasilkan selama pelatihan. Dengan memasukkan contoh-contoh ini ke dalam set pelatihan, model dapat belajar untuk mengenali dan mengatasi gangguan *adversarial*. Penelitian oleh Goodfellow et al. menunjukkan bahwa pelatihan dengan regularisasi *adversarial* secara signifikan meningkatkan ketahanan model terhadap serangan *adversarial* [21]. Namun, teknik ini memerlukan sumber daya komputasi yang lebih besar karena intensitas proses pelatihannya.
2) Pertahanan Berbasis Penghalusan (*Input Smoothing*): Teknik ini menggunakan metode seperti Gaussian blur atau median filter untuk menghaluskan input sebelum diproses oleh model. Tujuannya adalah untuk mengurangi dampak gangguan kecil yang ditambahkan pada input. Namun, efektivitas metode ini terbatas, karena serangan yang lebih canggih dapat menyesuaikan diri untuk melewati penghalusan tersebut [34].
3) Pertahanan Berbasis Ensembel (*Ensemble Methods*): Metode ini melibatkan penggunaan beberapa model yang berbeda untuk membuat prediksi. Dengan menggabungkan prediksi dari berbagai model, gangguan adversarial yang ditargetkan pada satu model dapat diminimalkan. Teknik ensembel telah terbukti efektif dalam meningkatkan ketahanan terhadap serangan adversarial, terutama ketika model-model yang digunakan memiliki arsitektur dan pelatihan yang berbeda [35].
4) Regularisasi Non-linearitas: Metode ini bertujuan menambah non-linearitas pada model melalui modifikasi arsitektur atau penambahan regularisasi. Dengan meningkatkan kompleksitas model, serangan adversarial yang memanfaatkan linearitas model menjadi kurang efektif [36].

5.2. Teknik Pertahanan Aktif

Teknik pertahanan aktif secara langsung mendeteksi dan mengurangi dampak serangan *adversarial* saat serangan tersebut terjadi. Beberapa metode dalam kategori ini antara lain:

1) Deteksi Anomali (*Anomaly Detection*): Teknik ini menggunakan model tambahan yang dilatih untuk mengenali input yang mencurigakan atau tidak sesuai. Input yang dideteksi sebagai anomali dapat

ditolak atau diproses dengan cara tertentu. Efektivitas metode ini bergantung pada kemampuan model deteksi dalam mengenali pola-pola yang menunjukkan adanya gangguan *adversarial*. Salah satu pendekatan yang populer adalah penggunaan *autoencoder* untuk mendeteksi input yang berbeda dari data pelatihan normal [37].

2) Transformasi Input (*Input Transformation*): Teknik ini mengubah input sebelum diproses oleh model utama. Contoh teknik ini termasuk penggunaan model generatif mendalam untuk merekonstruksi input sebelum diklasifikasikan. Proses transformasi ini membantu menghilangkan gangguan adversarial sebelum input mencapai model utama, sehingga meningkatkan ketahanan terhadap serangan [38].
3) Pembelajaran Adversarial Adaptif (Adaptive *Adversarial Learning*): Pendekatan ini melibatkan pembelajaran dinamis, di mana model secara berkelanjutan belajar mengenali dan menanggapi gangguan adversarial. Teknik ini melibatkan penyesuaian parameter model atau penggunaan sub-model yang dirancang khusus untuk mendeteksi dan menangani input yang mencurigakan. Misalnya, model dapat dilatih berulang kali dengan memasukkan contoh baru yang dihasilkan selama proses pelatihan [39].

5.3. Evaluasi Perbandingan Teknik Pertahanan

Bagian ini mengevaluasi dan membandingkan berbagai teknik pertahanan berdasarkan beberapa kriteria, untuk menentukan kelebihan dan kekurangan masing-masing pendekatan:

1) Efektivitas terhadap Berbagai Jenis Serangan: Evaluasi ini mengukur seberapa baik setiap teknik pertahanan dalam menangkal berbagai jenis serangan *adversarial*, baik serangan terarah maupun tidak terarah, serta serangan berbasis *white-box* dan *black-box*. Penelitian menunjukkan bahwa tidak ada satu teknik pun yang sepenuhnya efektif melawan semua jenis serangan, sehingga kombinasi beberapa teknik sering diperlukan untuk mencapai hasil yang optimal [22].
2) Biaya Komputasi: Evaluasi ini mempertimbangkan sumber daya komputasi yang diperlukan untuk menerapkan setiap teknik pertahanan. Beberapa metode, seperti adversarial training, memerlukan biaya komputasi yang tinggi karena model harus dilatih ulang dengan *adversarial examples*. Sebaliknya, teknik seperti deteksi anomali mungkin memerlukan tambahan komputasi saat prediksi, tetapi lebih efisien dalam jangka panjang [40].
3) Keberlanjutan dan Skalabilitas: Evaluasi ini melihat sejauh mana teknik pertahanan dapat diterapkan dalam skala besar dan dalam berbagai aplikasi. Teknik yang memerlukan penyesuaian spesifik untuk setiap jenis serangan mungkin tidak praktis untuk implementasi skala besar. Studi menunjukkan bahwa metode berbasis ensembel lebih mudah diskalakan karena fleksibilitasnya dalam menggabungkan berbagai model [23].

Dengan memahami dan mengevaluasi berbagai teknik pertahanan ini, peneliti dan praktisi dapat memilih metode yang paling sesuai untuk kebutuhan mereka, serta mengembangkan teknik baru yang lebih efektif dalam menghadapi ancaman yang terus berkembang.

6. Tantangan dan Arah Pengembangan Masa Depan

Bab kesimpulan ini merangkum temuan utama yang telah diuraikan dalam penelitian, menjelaskan implikasi praktis dari hasil yang diperoleh, serta mengidentifikasi arah penelitian lebih lanjut yang dapat membantu memperkuat ketahanan sistem pembelajaran mesin terhadap *adversarial examples*.

Penelitian ini telah mengungkap berbagai aspek penting terkait *adversarial examples* dalam konteks pembelajaran mesin, mulai dari pemahaman dasar konsep hingga eksplorasi teknik serangan dan pertahanan. Beberapa temuan utama dari penelitian ini meliputi:

1) Kerentanan Model Pembelajaran Mesin: Model pembelajaran mesin, terutama yang menggunakan arsitektur jaringan saraf dalam, terbukti rentan terhadap serangan adversarial yang mengeksploitasi kelemahan spesifik model tersebut.
2) Variasi Teknik Serangan: Beragam teknik serangan adversarial telah diklasifikasikan berdasarkan target dan metode yang digunakan. Hal ini menunjukkan bahwa tidak ada model yang sepenuhnya aman dari semua jenis serangan.
3) Efektivitas Teknik Pertahanan: Hasil evaluasi menunjukkan bahwa menggabungkan teknik pertahanan pasif dan aktif mampu memberikan perlindungan yang lebih baik dibandingkan hanya menggunakan satu pendekatan.

Penelitian ini juga membawa beberapa implikasi praktis yang penting untuk diimplementasikan dalam aplikasi dunia nyata, di antaranya:

1) Peningkatan Keamanan Sistem AI: Dengan memahami cara *adversarial examples* menipu model pembelajaran mesin, pengembang dapat merancang sistem yang lebih tangguh terhadap serangan,

sehingga meningkatkan keamanan aplikasi AI di berbagai bidang, seperti kendaraan otonom, pengenalan wajah, dan keamanan siber.

2) Pengembangan Kebijakan Keamanan: Pengetahuan tentang kerentanan model dan teknik pertahanan terhadap serangan *adversarial* dapat membantu organisasi dalam merancang kebijakan dan protokol keamanan yang lebih kuat guna melindungi aset digital mereka.
3) Desain Model yang Lebih Tahan: Penelitian ini mendorong pengembangan model pembelajaran mesin yang lebih kuat melalui pendekatan seperti regularisasi *adversarial* dan pembelajaran adaptif, yang dapat mengurangi risiko kesalahan klasifikasi akibat gangguan *adversarial*.

Meskipun penelitian ini telah memberikan kontribusi yang berarti, masih ada beberapa bidang yang memerlukan penelitian lebih lanjut, antara lain:

1) Pengembangan Teknik Pertahanan Baru: Diperlukan penelitian lanjutan untuk mengembangkan teknik pertahanan yang lebih efektif dan efisien, terutama dalam menghadapi serangan *adversarial* yang semakin kompleks dan sulit dideteksi.
2) Studi Jangka Panjang tentang Ketahanan Model: Penelitian longitudinal dapat memberikan wawasan tentang bagaimana model pembelajaran mesin merespons serangan *adversarial* dalam jangka waktu yang lama, serta membantu mengidentifikasi strategi pertahanan yang paling efektif.
3) Implikasi Etis dan Regulasi: Penelitian tambahan diperlukan untuk mengeksplorasi implikasi etis dari penggunaan dan pertahanan terhadap *adversarial examples*, terutama dalam hal privasi dan keamanan data. Selain itu, penting juga untuk mengembangkan regulasi yang mengatur penggunaan teknik ini dalam aplikasi yang bersifat kritis.

DAFTAR PUSTAKA

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012.

[3] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[4] C. Szegedy *et al.*, "Intriguing properties of neural networks," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, Dec. 2013, [Online]. Available: http://arxiv.org/abs/1312.6199

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–11, 2015.

[6] D. Eagleman, *The brain: The story of you*. Canongate Books, 2015.

[7] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2017, pp. 39–57. doi: 10.1109/SP.2017.49.

[8] T. Mitchell, "Introduction to machine learning," *Mach. Learn.*, vol. 7, pp. 2–5, 1997.

[9] B. CM, "Pattern recognition and machine learning." Springer, New York, 2010.

[10] J. MacQueen and others, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, 1967, pp. 281–297.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[12] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. pp. 436–444, May 2015. doi: 10.1038/nature14539.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv Prepr. arXiv1409.1556*, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[17] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization 3rd International Conference on Learning Representations," in *ICLR 2015-Conference Track Proceedings*, 2015.

[18] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, 2005, pp. 524–531.

[19] S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4–5, pp. 185–196, 1993.

[20] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of*

COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers, 2010, pp. 177–186.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–11, 2015.

[22] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–28, Jun. 2017, [Online]. Available: http://arxiv.org/abs/1706.06083

[23] H.-Y. Chen *et al.*, "Improving Adversarial Robustness via Guided Complement Entropy," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2019, pp. 4880–4888. doi: 10.1109/ICCV.2019.00498.

[24] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *2nd Int. Conf. Learn. Represent. ICLR 2014 - Work. Track Proc.*, pp. 1–8, Dec. 2013, [Online]. Available: http://arxiv.org/abs/1312.6034

[25] K. Warr, *Strengthening Deep Neural Networks*, First Edit. O'Reilly Media, Inc., 2019.

[26] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, 2008, [Online]. Available: https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

[27] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvari, "Learning with a Strong Adversary," no. 2014, pp. 1–12, 2015, [Online]. Available: http://arxiv.org/abs/1511.03034

[28] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2016, pp. 582–597. doi: 10.1109/SP.2016.41.

[29] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *5th Int. Conf. Learn. Represent. ICLR 2017 - Work. Track Proc.*, no. c, pp. 1–14, Jul. 2016, [Online]. Available: http://arxiv.org/abs/1607.02533

[30] C. Xiao, B. Li, J. Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *IJCAI Int. Jt. Conf. Artif. Intell.*, vol. 2018-July, pp. 3905–3911, 2018, doi: 10.24963/ijcai.2018/543.

[31] S. Baluja and I. Fischer, "Learning to attack: Adversarial transformation networks," *32nd AAAI Conf. Artif. Intell. AAAI 2018*, no. 1, pp. 2687–2695, 2018.

[32] M. Bojarski *et al.*, "End to end learning for self-driving cars," *arXiv Prepr. arXiv1604.07316*, 2016.

[33] A. K. Jain and S. Z. Li, *Handbook of face recognition*, vol. 1. Springer, 2011.

[34] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 9, pp. 2805–2824, Sep. 2019, doi: 10.1109/TNNLS.2018.2886017.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[36] M. Hein and M. Andriushchenko, "Formal guarantees on the robustness of a classifier against adversarial manipulation," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[37] L. Beggel, M. Pfeiffer, and B. Bischl, "Robust anomaly detection in images using adversarial autoencoders," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16--20, 2019, Proceedings, Part I*, 2020, pp. 206–222.

[38] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense Against Adversarial Attacks Using High-Level Representation Guided Denoiser," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 1778–1787. doi: 10.1109/CVPR.2018.00191.

[39] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 1, pp. 436–448, Feb. 2018, [Online]. Available: http://arxiv.org/abs/1802.00420

[40] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 2017.