BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

Landasan teori ini mencakup penjelasan mengenai Internet of Things (IoT) sebagai lingkungan yang rentan terhadap serangan Malware, GRU, metode reduksi dimensi PCA, MI, hybrid serta McNemar yang menjadi fokus perbandingan dalam penelitian.

2.1.1 *Internet of Things (IoT)*

IoT merupakan sebuah konsep di mana berbagai perangkat fisik dapat terhubung dan saling berkomunikasi melalui internet (Cahyanto dkk., 2022). Konsep ini tidak hanya mencakup pada perangkat konsumen, tetapi juga mencakup berbagai alat sektor industri, transportasi, pertanian, hingga kesehatan. Selain itu, pemanfaatan IoT memungkinkan proses pengumpulan dan analisis data secara *real time* yang pada akhirnya dapat meningkatkan efisiensi operasional di berbagai bidang (Asghar dkk., 2024)

Arsitektur IoT terdiri dari tiga lapisan utama yaitu perangkat, jaringan, dan aplikasi (Kilichev dkk., 2024). Lapisan perangkat bertugas mengumpulkan data dari lingkungan fisik, lapisan jaringan berfungsi untuk menghubungkan perangkat ke infrastruktur internet, memungkinkan data untuk ditransmisikan dan diterima dengan lancar. Terakhir, lapisan aplikasi bertanggung jawab untuk menganalisis data yang telah dikumpulkan, sehingga dapat memberikan informasi yang bertujuan untuk pengambilan keputusan dan tindakan lebih lanjut. Selain itu perangkat IoT

sering kali memiliki keterbatasan dalam hal sumber daya, seperti kapasitas CPU, RAM, dan daya baterai. Banyak perangkat IoT dirancang untuk beroperasi dalam batasan tertentu, yang membatasi kemampuan mereka untuk menjalankan algoritma keamanan yang rumit (Almazarqi dkk., 2021).

2.1.2 Malware

Malware, singkatan dari *malicious software*, merupakan perangkat lunak yang dirancang secara khusus untuk merusak, mengeksploitasi, atau memperoleh akses tidak sah ke sistem komputer, jaringan, atau perangkat tanpa sepengetahuan pemiliknya. Dalam konteks IoT, ancaman malware sangat signifikan karena banyak perangkat IoT memiliki keterbatasan kemampuan pemrosesan dan memori, sehingga menyulitkan penerapan mekanisme keamanan yang kompleks dan membuat perangkat ini rentan terhadap berbagai jenis malware, termasuk botnet dan ransomware (Pai dkk., 2025). Selain itu, malware IoT modern telah berkembang dengan teknik evasi khusus seperti obfuscation dan adaptasi terhadap lingkungan perangkat yang semakin meningkatkan risiko botnet dan serangan skala besar (Al-Marghilani, 2021).

Serangan malware pada jaringan IoT dapat menyebabkan gangguan serius terhadap ketersediaan layanan jaringan. Salah satu dampak utamanya adalah pemanfaatan perangkat IoT yang terinfeksi sebagai botnet untuk menghasilkan trafik dalam jumlah besar guna melancarkan serangan *Distributed Denial of Service* (DDoS). Dengan jumlah perangkat IoT yang diperkirakan mencapai lebih dari 18 miliar pada tahun 2024, skala serangan DDoS berbasis IoT berpotensi meningkat secara signifikan akibat bertambahnya jumlah perangkat yang dapat dikendalikan,

sehingga dapat membahayakan infrastruktur jaringan global jika tidak diatasi dengan mekanisme keamanan yang memadai (IoT Analytics, 2025).

2.1.3 Network Traffic Analysis

Analisis trafik jaringan pada IoT merupakan komponen utama dalam deteksi intrusi *(Intrusion Detection System / IDS)*, karena banyak perangkat IoT memiliki keterbatasan sumber daya, sehingga tidak memungkinkan untuk dilengkapi dengan sistem keamanan yang berat seperti pada komputer konvensional (Elrawy dkk., 2018). Melalui analisis trafik, pemantauan pola dan identifikasi anomali dapat dilakukan tanpa perlu mengakses payload perangkat, menjadikannya sebagai metode yang ringan dan praktis.

Dalam analisis trafik jaringan, terdapat dua pendekatan utama, yaitu *packet-based* dan *flow-based*. Pendekatan *packet-based* menganalisis setiap paket (header dan payload) secara terpisah, sehingga memberikan detail yang sangat mendalam. Namun, metode ini memerlukan sumber daya komputasi yang tinggi. Di sisi lain, analisis *flow-based* menggabungkan paket-paket dengan karakteristik serupa seperti alamat IP sumber dan tujuan, port, serta protokol menjadi satu flow, kemudian mengekstrak fitur statistik dari flow tersebut, seperti durasi, jumlah paket, dan volume data. Pendekatan ini lebih efisien, lebih scalable, dan lebih menjaga privasi karena hanya menggunakan metadata, bukan payload lengkap (Mondragon dkk., 2025). Selain itu, analisis *flow-based* banyak diterapkan dalam sistem deteksi intrusi berbasis *machine learning* untuk jaringan IoT, karena dapat mendeteksi pola komunikasi yang abnormal tanpa membebani perangkat dengan sumber daya terbatas.

Pendekatan *flow-based* banyak digunakan dalam dataset IDS modern, termasuk IoT, karena mampu merepresentasikan trafik jaringan berskala besar dengan kebutuhan pemrosesan dan penyimpanan yang relatif rendah (Elrawy dkk., 2018). Tantangan utama *flow-based analysis* adalah dimensionalitas tinggi. Sejumlah fitur statistik yang diekstrak bisa bersifat redudan, berkorelasi kuat satu sama lain, atau kurang diskriminatif *(feature noise)*, sehingga menyulitkan proses deteksi (Sarhan dkk., 2022). Oleh karena itu, penerapan teknik reduksi fitur menjadi solusi untuk menjaga performa deteksi sekaligus menurunkan kompleksitas komputasi. Dengan demikian, *network traffic analysis* berbasis flow menyediakan keseimbangan antara efisiensi komputasi dan efektivitas deteksi serangan IoT, terutama jika dikombinasikan dengan teknik reduksi dimensi dan model *machine learning/deep learning.*

2.1.4 Reduksi Dimensi

Reduksi dimensi adalah teknik untuk mengurangi jumlah fitur dalam dataset sambil tetap mempertahankan informasi penting, sehingga model bisa lebih efisien dan tidak mudah overfit. Teknik ini sangat penting ketika menghadapi data berdimensi tinggi, seperti trafik jaringan IoT, di mana banyak fitur bisa *redundant* (Abdulhammed dkk., 2019). Dengan mengurangi dimensi, beban komputasi berkurang, redundansi diminimalkan, dan kemampuan generalisasi model meningkat.

Teknik reduksi dimensi dapat dikategorikan menjadi dua jenis utama, diantaranya *feature selection* dan *feature extraction*. *Feature selection* berfungsi untuk memilih subset fitur yang paling relevan dari fitur asli tanpa mengubah

representasinya, sehingga interpretabilitas tetap terjaga. Di sisi lain, *feature extraction* mentransformasikan fitur asli menjadi fitur baru melalui kombinasi linear atau non-linear, yang memungkinkan pengurangan dimensi yang lebih signifikan, meskipun dengan interpretabilitas yang berkurang (Nogales & Benalcázar, 2023). Berdasarkan penggunaan label kelas, teknik reduksi dimensi dapat dibedakan menjadi metode unsupervised, seperti PCA dan autoencoder yang berfokus pada struktur intrinsik data, atau metode supervised, seperti Linear Discriminant Analysis (LDA) dan Mutual Information, yang memanfaatkan label untuk memilih fitur yang paling diskriminatif (Li dkk., 2024) Pemilihan metode yang tepat sangat bergantung pada karakteristik data trafik IoT dan kebutuhan sistem deteksi, apakah memprioritaskan kecepatan pemrosesan (*lightweight*) atau akurasi deteksi pola serangan yang kompleks.

2.1.5 Mutual Information

Mutual Information (MI) merupakan metode seleksi fitur berbasis teori informasi yang mengukur tingkat ketergantungan antara suatu fitur dengan variabel target melalui besarnya informasi yang dibagikan di antara keduanya. Secara matematis, MI didefinisikan sebagai *Kullback–Leibler divergence* antara distribusi probabilitas joint $p(x, y)$ dan hasil perkalian distribusi marginalnya $p(x), p(y)$ yang secara umum dirumuskan sebagai berikut pada persamaan 2.1 (Huang dkk., 2024) :

$$MI(X,Y) = -\iint fX,Y(x,y)\, log\left(\frac{fX,Y(x,y)}{fX(x)\, Y(y)}\right) dxdy \tag{2.1}$$

Berdasarkan persamaan tersebut, MI akan bernilai nol apabila kedua variabel bersifat independen, yang berarti tidak terdapat informasi yang saling dibagikan di antara keduanya. Sebaliknya, semakin besar nilai MI, semakin kuat pula ketergantungan antara variabel Xdan Y. Dalam konteks seleksi fitur, fitur dengan nilai MI tinggi dianggap lebih informatif dan lebih relevan untuk meningkatkan performa prediksi, sedangkan fitur dengan MI rendah cenderung dihilangkan karena berpotensi menambah redundansi dalam data (Yousefi-Azar dkk., 2021).

2.1.6 Principal Component Analysis (PCA)

PCA pertama kali diperkenalkan oleh Karl Pearson pada tahun 1901 dan kemudian diformalkan lebih lanjut oleh Harold Hotelling pada tahun 1933. PCA merupakan teknik reduksi dimensi yang melakukan transformasi linier terhadap data untuk membentuk sejumlah komponen baru yang saling tidak berkorelasi secara linear, yang disebut sebagai komponen utama. Setiap komponen utama merupakan kombinasi linier dari fitur asli dan disusun berdasarkan besarnya variansi data yang dapat dijelaskan (Alaudin Shalih dkk., 2025). Tujuan PCA adalah mengurangi jumlah fitur dengan membentuk representasi baru berdimensi lebih rendah yang tetap mempertahankan sebagian besar informasi penting dari data asli. PCA bekerja dengan memproyeksikan data ke sejumlah komponen utama *(principal components)* yang merupakan kombinasi linier dari variabel awal. Komponen utama tersebut disusun berdasarkan besarnya variansi data yang dapat dijelaskan, sehingga komponen pertama memiliki kontribusi variansi terbesar. (Hasan & Abdulazeez, 2021).

Penerapan PCA pada data *time series* dapat digunakan untuk mengurangi jumlah dimensi input sebelum proses pelatihan model deep learning. Dengan berkurangnya jumlah fitur yang diproses, beban komputasi selama pelatihan model dapat ditekan. Selain itu, PCA berperan dalam merangkum informasi statistik utama dari data melalui komponen utama, sehingga redundansi antar fitur dapat dikurangi dan risiko overfitting dapat diminimalkan. (Gao dkk., 2024).

2.1.7 Hybrid (MI - PCA)

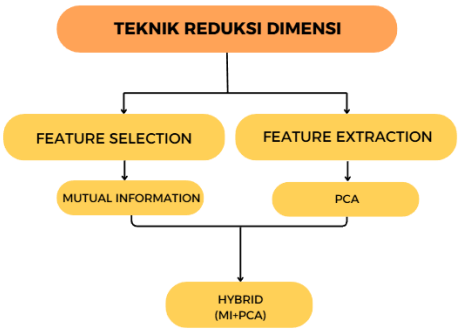
Hybrid (MI - PCA) merupakan pendekatan reduksi dimensi dua tahap yang mengombinasikan metode seleksi fitur dan ekstraksi fitur dalam satu proses berurutan. Pendekatan ini digunakan untuk mengurangi jumlah fitur sekaligus mempertahankan informasi penting yang relevan terhadap proses klasifikasi pada data trafik jaringan IoT yang umumnya memiliki dimensi tinggi serta mengandung fitur redundan.

Pada tahap pertama, Mutual Information (MI) digunakan sebagai metode seleksi fitur untuk mengukur tingkat ketergantungan antara setiap fitur dan label target. Fitur dengan nilai MI tinggi dipertahankan karena memiliki kontribusi informasi yang lebih besar terhadap proses klasifikasi, sedangkan fitur dengan nilai rendah dieliminasi. Namun, seleksi fitur berbasis MI tidak mempertimbangkan korelasi antar fitur terpilih sehingga redundansi masih dapat terjadi.

Pada tahap kedua, Principal Component Analysis (PCA) diterapkan sebagai metode ekstraksi fitur terhadap hasil seleksi MI. PCA melakukan transformasi linier terhadap fitur menjadi sejumlah komponen utama yang saling ortogonal, sehingga korelasi antar fitur dapat dikurangi dan representasi data menjadi lebih ringkas.

Dengan urutan proses MI diikuti PCA, reduksi dimensi dilakukan secara bertahap, yaitu seleksi fitur berdasarkan relevansi terhadap label kemudian transformasi fitur untuk mengurangi redundansi. Pendekatan ini disebut hybrid karena menggabungkan dua teknik reduksi dimensi dengan fungsi berbeda dalam satu alur preprocessing data sebelum proses pelatihan model.

Kombinasi MI dan PCA layak digunakan secara bersamaan karena keduanya memiliki mekanisme reduksi dimensi yang saling melengkapi. MI berperan dalam memilih fitur yang paling relevan terhadap label kelas *(supervised relevance filtering)*, sedangkan PCA berfungsi mereduksi korelasi dan redundansi antar fitur melalui transformasi ortogonal. Dengan demikian, pendekatan hybrid tidak hanya mengurangi fitur yang tidak informatif, tetapi juga menyederhanakan struktur ruang fitur, sehingga diharapkan dapat meningkatkan efisiensi komputasi serta kemampuan generalisasi model.Gambar 2.1 menunjukkan konsep pendekatan hybrid MI–PCA

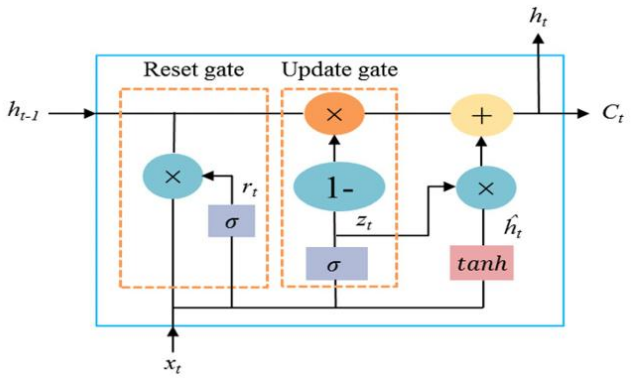


Gambar 2. 1 Hybrid MI+PCA

2.1.8 Gate Recurrent Unit (GRU)

Gated Recurrent Unit (GRU) merupakan salah satu varian arsitektur dari *Recurrent Neural Network* (RNN) yang diperkenalkan oleh Kyunghyun dan

rekannya pada tahun 2014 sebagai modifikasi dan alternatif dari arsitektur *Long Short-Term Memory* (LSTM). GRU merupakan sebuah versi algoritma yang dimodifikasi dari struktur sel LSTM dengan menggabungkan tiga unit gerbang LSTM menjadi dua unit gerbang, yang terdiri dari *update gate* (zt) dan *reset gate* (rt) (Mahjoub et al., 2022). *Reset gate* berfungsi untuk mengatur jumlah informasi yang perlu dilupakan dari masa lalu, sementara *Update gate* bertugas untuk menentukan seberapa banyak informasi terkini yang akan diserap. Dengan kata lain, pintu *reset* mendukung model dalam mengelola memori dengan mengeliminasi informasi yang tidak diperlukan, sedangkan pintu *update* memastikan informasi baru yang krusial dapat diterima dan dipertahankan selama proses pembelajaran (Arfianti dkk., 2021) Gambar 2.1 menunjukkan arsitektur dari GRU yang terdiri dari reset gate dan update gate.



Gambar 2. 2 Aristektur Gated Recurrent Unit (GRU)(Mohsen, 2023)

GRU dirancang untuk mengatasi permasalahan yang sering dialami oleh RNN tradisional, terutama dalam menangani dependensi jangka panjang *(long-term dependencies)* pada data sekuensial. Secara konseptual, GRU merupakan versi lebih sederhana dari LSTM, namun tetap mempertahankan kemampuan untuk mempelajari dan mengelola dependensi jangka panjang dalam data sekuensial

(Mienye et al., 2024). Keunggulan GRU dibandingkan dengan LSTM terletak pada arsitektur yang lebih sederhana, sehingga proses komputasi lebih efisien dan parameter yang digunakan untuk dilatih lebih sedikit. Struktur yang dimiliki GRU ini menjadikannya sebagai pilihan yang lebih optimal untuk dataset yang relatif kecil, namun tetap mempertahankan performa yang sebanding dengan LSTM dalam berbagai tugas pemrosesan data sekuensial (Mienye et al., 2024).

Lebih lanjut, hasil penelitian menunjukkan bahwa pada tugas pembelajaran urutan simbolik, GRU memiliki waktu pelatihan yang lebih cepat dibandingkan LSTM, GRU mampu menyelesaikan proses pelatihan dengan rata-rata waktu 19,72 detik, sedangkan LSTM memerlukan waktu 37,19 detik untuk mencapai tingkat akurasi prediksi yang sebanding, sehingga GRU dinilai dua kali lebih efisien dalam waktu pelatihan (Cahuantzi dkk., 2021). Walaupun studi tersebut berfokus pada data simbolik, temuan ini tetap relevan dalam konteks penelitian ini karena baik data simbolik maupun data trafik jaringan IoT merupakan jenis data sekuensial yang memerlukan pemodelan terhadap pola urutan. Model GRU terdiri dari tiga lapisan utama, diantaranya *input layer, output layer, dan implicit layer.* Input yang masuk ke GRU merupakan data dengan jenis *time series* pada waktu t setelah melalui proses preprocessing (Syifa dkk., 2023). Algoritme yang digunakan dalam sel GRU dapat dijelaskan sebagai berikut:

1. Reset Gate (rt)

   Reset Gate bertujuan untuk menentukan seberapa banyak informasi sebelumnya yang tidak relevan dan harus dihapus. Secara metematis reset gate diformulasikan sebagai :

$$R_t = \sigma\,(W_r\,[H_{t-1}, X_t] + b_r \quad (2.2)$$

Berdasarkan persamaan 2.2 Rt merujuk terhadap seret gate pada langkah waktu t, σ merupakan fungsi aktivasi *sigmoid*, Wr ialah matriks bobot yang digunakan untuk gerbang reset, br adalah nilai bias untuk gerbang reset, H(t−1) menunjukkan keadaan tersembunyi pada langkah waktu sebelumnya, dan Xt adalah input pada langkah waktu t (Lu & Hu, 2023).

2. Update Gate (zt)

Update gate mengontrol seberapa banyak informasi dari *hidden state* sebelumnya yang perlu dipertahankan.

$$Z_t = \sigma\,(W_z\,[H_{t-1}, X_t] + b_z \quad (2.3)$$

Zt merujuk *gate* pembaruan pada langkah waktu t, Wz merupakan matriks bobot yang digunakan untuk gerbang pembaruan, dan bz ialah nilai bias untuk gerbang pembaruan (Lu & Hu, 2023).

3. Candidate Hidden State ($\tilde{h}t$)

Candidate hidden state berperan untuk mengembalikan informasi yang relevan dari masa lalu.

$$\tilde{h}_t = \mathrm{Tan}h\,(W[R_t * H_{t-1}, X_t] + b \quad (2.4)$$

Ht adalah candidate hidden state pada langkah waktu t, dan Tanh adalah fungsi aktivasi tangens hiperbolik (Lu & Hu, 2023).

4. Final Hidden State (ht)

Final hidden state berperan untuk menyimpan hasil akhir dari pemrosesan informasi pada setiap langkah waktu t. Nilai ini dihasilkan dari

kombinasi antara keadaan tersembunyi sebelumnya dan candidate hidden state, di mana penggabungan ini diatur oleh update gate (Lu & Hu, 2023) .

$$H_t = (1 - Z_t) * [H_{t-1} + Z_t] * \tilde{h}_t \quad (2.5)$$

2.1.9 Statistical Validation

Merupakan proses untuk memastikan bahwa perbedaan performa antar model dalam penelitian *machine learning* benar-benar signifikan secara statistik dan bukan hasil dari kebetulan atau variasi acak pada data. Dalam evaluasi model, metrik standar seperti akurasi sering digunakan untuk mengukur kinerja. Namun, perbedaan nilai rata-rata metrik tersebut belum dapat dijadikan bukti bahwa suatu model secara konsisten lebih unggul. Hal ini disebabkan karena selisih performa antar model dapat saja merupakan *statistical anomaly*, yaitu perbedaan yang muncul secara acak dan tidak mencerminkan keunggulan nyata (Hamarashid, 2021).

2.1.9.1 McNemar Test

McNemar Test adalah uji statistik yang penting untuk membandingkan dua hasil yang berpasangan dan berskala nominal atau biner. Uji ini diperkenalkan oleh Quinn McNemar pada tahun 1947 hal ini sejalan dengan (Rainio dkk., 2024) yang menekankan bahwa uji ini merupakan metode yang kuat dan direkomendasikan untuk membandingkan model klasifikasi modern.

McNemar Test mengevaluasi kinerja dua model dengan menganalisis jumlah kasus ketidaksepakatan (*discordant pairs*), yaitu kondisi ketika satu model menghasilkan prediksi benar sedangkan model lain salah. Sebaliknya, kasus

kesepakatan, baik ketika kedua model sama-sama benar maupun sama-sama salah, tidak digunakan dalam perhitungan karena tidak memberikan informasi mengenai perbedaan performa antar model. Hasil prediksi kedua model direpresentasikan dalam bentuk tabel kontingensi 2 × 2 sebagaimana ditunjukkan pada Gambar 2.3

	Model B Benar	Model B Salah
Model A Benar	N_{00}	N_{01}
Model A Salah	N_{10}	N_{11}

Gambar 2. 3 Kontingensi Model Klasifikasi untuk McNemar Test

Tujuan utama uji McNemar adalah menguji apakah terdapat perbedaan signifikan antara performa dua model klasifikasi yang diuji pada dataset yang sama. Hipotesis nol (H_0) menyatakan bahwa kedua model memiliki tingkat kesalahan yang sama, yang secara formal dinyatakan sebagai probabilitas ketidaksepakatan berlawanan bernilai sama, yaitu $P(n_{01}) = P(n_{10})$. Sebaliknya, hipotesis alternatif (H_1) menyatakan bahwa kedua model memiliki performa yang berbeda secara statistik, yaitu $P(n_{01}) \neq P(n_{10})$. Untuk menguji hipotesis tersebut, Mcnemar test menggunakan statistik X^2 dengan distribusi chi-square berderajat kebebasan 1 ($df = 1$). Bentuk umum statistik uji (tanpa koreksi) adalah:

$$\chi^2 = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}} \tag{2.6}$$

Uji ini sesuai digunakan apabila jumlah pasangan ketidaksepakatan cukup besar sehingga asumsi aproksimasi chi-square terpenuhi. Apabila jumlah pasangan ketidaksepakatan relatif kecil (misalnya kurang dari 25), digunakan versi dengan koreksi kontinuitas Yates, yaitu:

$$\chi^2 = \frac{(|\, n_{01} - n_{10} \,| - 1)^2}{n_{01} + n_{10}} \tag{2.7}$$

Nilai χ^2yang diperoleh kemudian dibandingkan dengan nilai kritis dari distribusi chi-square (atau dihitung p-value) untuk memutuskan apakah menolak H_0. Jika p-value lebih kecil dari tingkat signifikansi α(misalnya 0,05), maka disimpulkan bahwa terdapat perbedaan performa yang signifikan antara kedua model.

Penggunaan McNemar didasarkan pada kesesuaiannya dengan penelitian ini, yaitu membandingkan dua model klasifikasi pada dataset yang sama dengan keluaran prediksi bersifat biner. Metode ini secara khusus dirancang untuk data berpasangan dan telah digunakan dalam studi empiris perbandingan model machine learning dan deep learning untuk menguji signifikansi perbedaan performa classifier (Tian dkk., 2025). Oleh karena itu, McNemar dinilai sebagai pendekatan statistik yang paling sesuai dibandingkan metode lain yang umumnya digunakan untuk data kontinu atau tidak berpasangan.

2.1.10 Matriks Evaluasi

Evaluasi model klasifikasi membutuhkan metrik yang lebih informatif dibandingkan dengan akurasi keseluruhan. Confusion Matrix menjadi dasar untuk menghitung berbagai metrik evaluasi yang spesifik, karena memberikan gambaran visual yang menyeluruh mengenai kinerja model dengan membandingkan label yang sebenarnya dengan label yang diprediksi.

2.1.10.1 Confusion Matrix

Confusion Matrix merupakan instrumen dasar dalam evaluasi model klasifikasi yang membandingkan prediksi model dengan label aktual melalui sebuah tabel kontingensi, sehingga memungkinkan identifikasi jumlah prediksi

benar maupun salah secara detail. Pada klasifikasi biner, *confusion matrix* terdiri atas empat komponen: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). (De Diego dkk., 2022). Dengan demikian, *confusion matrix* tidak hanya memberikan ukuran akurasi keseluruhan tetapi juga memungkinkan analisis mendalam terhadap jenis kesalahan yang dibuat model, serta mendasari perhitungan metrik lain seperti presisi, recall, dan F1-score (Zeng, 2025).

		AKTUAL	
		Positif	**Negatif**
PREDIKSI	**Positif**	True Positif (TP)	False Positif (FP)
	Negatif	False Negatif (FN)	True Negaatif (TN)

Gambar 2. 4 Confusion Matrix

Tabel confusion matrix pada gambar 2.4 menunjukkan bagaimana prediksi model dibandingkan dengan label aktual dalam struktur 2×2 dengan rincian sebagai berikut :

True Positif (TP) : Jumlah sampel positif yang diprediksi dengan benar sebagai positif.

True Negatif (TN) : Jumlah sampel negatif yang diprediksi dengan benar sebagai negatif.

False Positif (FP) : Jumlah sampel negatif yang salah diprediksi sebagai positif (Kesalahan Tipe I).

False Negatif (FN) : Jumlah sampel positif yang salah diprediksi sebagai negatif (Kesalahan Tipe II).

Pola angka pada sel diagonal maupun non-diagonal menunjukkan seberapa banyak model membuat prediksi benar dan salah, serta apakah kesalahan lebih sering berupa *false positives* atau *false negatives* (Sathyanarayanan, 2024). Pola ini menjadi dasar evaluasi untuk memutuskan langkah perbaikan, seperti mengatur threshold atau menerapkan teknik resampling untuk menangani ketidakseimbangan kelas (Carvalho dkk., 2025). Jika klasifikasi melibatkan lebih dari dua kelas, matriks diperluas menjadi n×n sehingga analisis misclassification antarkelas dapat dilakukan.

2.1.10.2 Akurasi

Akurasi adalah metrik evaluasi mendasar yang mengukur proporsi total prediksi yang benar dibandingkan seluruh prediksi yang dilakukan oleh model klasifikasi. Secara matematis, akurasi dirumuskan sebagai berikut :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.8}$$

Metrik ini memberikan gambaran umum seberapa sering model membuat prediksi yang tepat, sehingga banyak digunakan sebagai metrik awal dalam penelitian klasifikasi. Namun, ketika distribusi kelas dalam dataset tidak seimbang, akurasi bisa menyesatkan karena model yang hanya memprediksi kelas mayoritas saja dapat memperoleh nilai akurasi tinggi tanpa benar-benar mampu mengidentifikasi kasus pada kelas minoritas (Chen dkk., 2024).

2.1.10.3 Presisi

Presisi *(precision)* mengukur proporsi prediksi positif yang benar dari seluruh prediksi positif yang dihasilkan model. Secara matematis, presisi dapat dirumuskan sebagai berikut :

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

Dengan demikian, presisi menunjukkan seberapa terpercaya prediksi positif model apabila presisi tinggi, maka dari semua *instance* yang diprediksi positif, sebagian besar benar-benar positif.

2.1.10.4 Recall

Recall adalah metrik yang mengukur proporsi dari semua kasus aktual positif yang berhasil dikenali dengan benar oleh model. Recall dapat dihitung dengan rumus:

$$Recall = \frac{TP}{TP + FN} \tag{2.10}$$

Recall memberikan gambaran tentang seberapa "lengkap" model dalam menangkap *instance-instance* positif sesungguhnya semakin tinggi nilai *recall*, semakin sedikit kasus positif yang terlewat (FN rendah).

2.1.10.5 F1-Score

F1-Score adalah metrik evaluasi yang menggabungkan dua aspek penting dari performa model klasifikasi yaitu *precision* dan *recall* ke dalam satu nilai tunggal, dengan menggunakan rata-rata harmonik dari keduanya. Secara matematis, F1-Score dinyatakan sebagai:

$$F1 = 2\,x\,\frac{Precision\ x\ Recall}{Precision + Recall} \quad (2.11)$$

Hal ini membuat F1-Score hanya mencapai nilai tinggi jika baik precision maupun recall sama-sama baik; jika salah satu rendah, maka nilai F1-Score otomatis turun.

2.1.10.5 Specificity

Specificity adalah metrik evaluasi yang mengukur proporsi dari semua kasus aktual negatif yang berhasil diklasifikasi dengan benar sebagai negatif oleh model. Secara matematis specificity dinyatakan sebagai berikut :

$$Specificity = \frac{TN}{TN + FP} \quad (2.\ 12)$$

Specificity mengukur kemampuan model dalam mengidentifikasi dengan benar kasus negatif, yang berarti menghindari terjadinya "kesalahan positif." Namun, *specificity* hanya mencerminkan kinerja pada kelas negatif, sehingga tidak memberikan informasi mengenai seberapa efektif model dalam mengenali kelas positif.

2.2 Penelitian Terkait

Berbagai penelitian telah mengusulkan teknik reduksi dimensi untuk mengatasi permasalahan redundansi dan kompleksitas fitur. Pendekatan yang umum digunakan seperti Mutual Information (MI), Principal Component Analysis (PCA), serta pendekatan berbasis autoencoder. Selain itu, pengembangan arsitektur *deep learning* seperti CNN, LSTM, dan GRU juga banyak digunakan untuk meningkatkan kemampuan deteksi serangan pada lingkungan IoT. Tabel 2.1 merangkum penelitian-penelitian terdahulu yang relevan dengan topik reduksi dimensi dan pengembangan model IDS pada lingkungan IoT.

Tabel 2. 1 *State-of-the-Art* Penelitian

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
1	(Talukder dkk., 2025)	A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction	Dataset IDS pada WSN dan IoT sering mengalami masalah class imbalance dan high dimensionality yang menyebabkan overfitting, bias terhadap majority class, dan peningkatan kompleksitas komputasi.	Mengusulkan model hybrid yang mengintegrasikan: (1) KMeans-SMOTE (KMS) untuk data balancing, (2) PCA untuk reduksi dimensi menggunakan Kaiser Criterion, (3) Random Forest Classifier (RFC) sebagai model utama. Dataset: WSN-DS dan TON-IoT.	WSN-DS: Akurasi **99.94%**, F1-score **99.94%**. TON-IoT: Akurasi **99.97%**, F1-score **99.97%**. Model hybrid (KMS+PCA+RFC) mengungguli metode STL dan GAN dalam balancing dan efisiensi komputasi.
2	(Shaikh dkk., 2025)	A deep Reinforcement learning-based robust	IoMT memiliki keterbatasan sumber daya dan rentan	Mengusulkan HCLR-IDS yang mengintegrasikan: (1) Enhanced	Akurasi klasifikasi biner: **99.58%**. Akurasi multi-class (18

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
		Intrusion Detection System for securing IoMT Healthcare Networks	terhadap serangan dinamis. IDS tradisional tidak mampu menangkap pola spasial dan temporal secara bersamaan serta sulit beradaptasi terhadap serangan baru. Tingginya false positive dan ketidakmampuan menangani evolving attacks menjadi tantangan utama.	Mutual Information Feature Selection (MIFS), (2) Hybrid CNN-LSTM untuk ekstraksi fitur spasial & temporal, (3) Reinforcement Learning (DQN + PPO) untuk pengambilan keputusan adaptif. Dataset: CICIoMT2024 (3.2 juta data, 18 jenis serangan).	kelas): **77.73%**. Model menunjukkan peningkatan robustness dan kemampuan adaptasi terhadap serangan dinamis dibanding IDS tradisional.
3	(Albalwy & Almohaimeed, 2025)	Advancing Artificial Intelligence of Things Security: Integrating Feature Selection and Deep Learning for Real-Time Intrusion Detection	Sistem IDS pada lingkungan IoT menghadapi tantangan tingginya dimensi data (83 fitur), keterbatasan resource perangkat IoT, serta kesulitan menjaga keseimbangan antara akurasi deteksi dan efisiensi komputasi. Penggunaan seluruh fitur menyebabkan noise, overfitting, dan penurunan performa model deep learning.	Mengintegrasikan lima metode feature selection (CFS, Pearson, Gain Ratio, Information Gain, Symmetrical Uncertainty) dengan PCA, kemudian diuji menggunakan tiga model DL (ANN, DNN, TabNet) pada dataset RT-IoT2022. Dilakukan tiga skenario eksperimen: (1) DL tanpa reduksi dimensi, (2) DL + feature selection, (3) DL + feature selection + PCA.	Tanpa seleksi fitur, akurasi model hanya berkisar 92% (ANN) dan 94% (TabNet). Kombinasi terbaik adalah Pearson + PCA + ANN yang mencapai akurasi 99.7%. Integrasi seleksi fitur dan PCA terbukti signifikan meningkatkan efisiensi dan akurasi.
4	(Sheheryar & Sharma, 2025)	Ensemble Feature Engineering and Deep Learning for Botnet Attacks Detection in the Internet of Things	IDS pada jaringan IoT menghadapi tantangan besar berupa traffic yang kompleks dan heterogen, jumlah fitur sangat banyak, serta adanya fitur redundant yang menurunkan akurasi model deep learning. Model DL	Mengusulkan Proficient Ensemble Feature Engineering untuk feature selection dengan kombinasi: (1) Genetic Algorithm approach, (2) filter methods: Mutual Information, LASSO Regularization, Forward–Backward Search, (3) Merger	Performa model DL setelah seleksi fitur sangat tinggi: Precision 97.88–98.99%, Recall 99.10–99.95%, Accuracy 98.05–99.21%, F1-score 98.45–99.82%. Metode ensemble feature engineering mencapai Precision 98.26%, Recall 99.68%,

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
			murni sering mengalami overfitting dan tidak efisien untuk real-time intrusion detection.	Approach untuk mengatasi redundancy, (4) Recursive Feature Removal (wrapper) untuk memilih top 9 fitur terbaik. Validasi dilakukan dengan deep learning models: CNN, RNN, LSTM, GRU pada dataset IoT-Botnet 2020.	Accuracy 98.49%, F1 99.00%, AUC-ROC 82.37%, Specificity 98.38%. Hasil menunjukkan metode sangat robust untuk mendeteksi traffic benign vs malicious botnet IoT.
5	(Sagu dkk., 2025)	Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm	Metode IDS tradisional sulit mendeteksi serangan IoT yang kompleks karena karakteristik data bersifat spasial dan temporal, volume besar, serta variasi pola serangan. Model konvensional tidak mampu menangkap kedua pola tersebut secara simultan dan memiliki tingkat error (FPR/FNR) yang masih tinggi.	Mengusulkan model hybrid CNN-GRU untuk menangkap fitur spasial (CNN) dan temporal (GRU), dengan optimisasi hyperparameter menggunakan algoritma Self-Upgraded Cat and Mouse Optimization (SUCMO). Dievaluasi pada dataset UNSW-NB15 dan BoT-IoT untuk menguji generalisasi model.	Pada dataset BoT-IoT, model mencapai akurasi 98.71% dan F1-score 98.58%. Pada UNSW-NB15, model menunjukkan akurasi, precision, recall (sensitivity), specificity, dan F1-score yang tinggi dengan nilai false positive rate (FPR) dan false negative rate (FNR) rendah. Secara keseluruhan, model outperform metode konvensional dan beberapa pendekatan sebelumnya.
6	(G. & K., 2025)	A GRU-based approach for botnet detection using deep learning technique	Serangan botnet pada lingkungan IoT sulit dideteksi karena trafik bersifat sekuensial dan kompleks (dataset UNSW-NB15). Model klasik (RF, DT, KNN) kurang mampu menangkap dependensi temporal.	Mengembangkan IDS berbasis GRU + SelectKBest (Chi-Square) untuk seleksi fitur. Dibandingkan dengan RF, DT, KNN, dan LSTM. Fokus pada efisiensi komputasi untuk resource-constrained IoT.	Akurasi: 97,38% Precision: 98,10% Recall: 97,12% F1-score: 97,61% (lebih tinggi dari LSTM 97,23%) GRU mengungguli RF, DT, KNN, dan LSTM dalam semua metrik evaluasi. Lebih ringan dan efisien dibanding LSTM.

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
7	(Archana & Aneetha, 2024)	LSTM-MI: Revolutionizing Intrusion Detection Through Adaptive Learning and Mutual Information Analysis	Data jaringan memiliki dimensi tinggi (high-dimensional data) sehingga meningkatkan kompleksitas komputasi dan risiko overfitting pada model deep learning. Selain itu, belum ada evaluasi komparatif yang sistematis antara PCA dan Mutual Information (MI) dalam konteks LSTM untuk IDS.	Mengembangkan dua pendekatan preprocessing sebelum LSTM: (1) PCA untuk dimensionality reduction, dan (2) Mutual Information (MI) untuk feature selection. Dilakukan perbandingan sistematis performa LSTM-PCA dan LSTM-MI pada dataset CSE-CIC-IDS2018.	(1) PCA untuk dimensionality reduction, dan (2) Mutual Information (MI) untuk feature selection. Dilakukan perbandingan sistematis performa LSTM-PCA dan LSTM-MI pada dataset CSE-CIC-IDS2018. LSTM-PCA → Akurasi 97,73%. LSTM-MI → Akurasi 98,21%, Sensitivity 98,80%, Specificity 96,30%, AUC 0,984. MI terbukti lebih efektif dibanding PCA dalam meningkatkan performa LSTM untuk IDS.
8	(Balhareth & Ilyas, 2024)	Optimized Intrusion Detection for IoMT Networks with Tree-Based Machine Learning and Filter-Based Feature Selection	Jaringan IoMT (Internet of Medical Things) sangat sensitif terhadap serangan siber. Tantangan utama: (1) kebutuhan deteksi cepat dengan latency rendah, (2) keterbatasan resource perangkat medis, (3) banyak fitur tidak relevan pada dataset medis yang memperlambat training, (4) risiko kesalahan deteksi yang berdampak	1. Tree-Based Machine Learning: Decision Tree, Random Forest, XGBoost, CatBoost. 2. Filter-Based Feature Selection: Mutual Information (MI) + XGBoost Feature Importance + metode Intersection untuk memilih fitur paling relevan. Dataset: CICIDS2017. Fokus pada peningkatan akurasi sekaligus efisiensi komputasi pada lingkungan IoMT.	Akurasi: **98.79%** False Alarm Rate: **0.007** Precision, Recall, F1-score tinggi dan stabil. Sistem menunjukkan efisiensi komputasi yang baik dan mampu menjaga performa deteksi pada jaringan IoMT.

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
			langsung pada keselamatan pasien.		
9	(Shoab & Alsbatin, 2024)	GRU Enabled Intrusion Detection System for IoT Environment with Swarm Optimization and Gaussian Random Forest Classification	Lingkungan IoT menghasilkan trafik sekuensial time-series yang kompleks. Model konvensional gagal menangkap dependensi temporal. Selain itu, pemilihan hyperparameter yang tidak optimal menyebabkan overfitting dan performa tidak stabil.	Pendekatan tiga tahap (Extract–Select–Classify): 1. Feature Extraction: GRU + Autoencoder untuk menangkap pola temporal trafik jaringan. 2. Feature Selection: Variational Cuckoo Search + Particle Swarm Optimization (PSO). 3. Classification: Hybrid Gaussian Naïve Bayes + Random Forest. Dataset: NSL-KDD.	Akurasi: 99.82% F1-score: 99.45% Detection Rate: 99.8% Waktu komputasi: 32.56 jam (2–3× lebih cepat dibanding baseline). Sistem robust pada klasifikasi multi-kelas (DoS, Probe, R2L, U2R). GRU terbukti efektif dalam menangkap dependensi temporal trafik IoT.
10	(John dkk., 2024)	Enhanced Intrusion Detection Model Based on PCA and Variable Ensemble Machine Learning Algorithm	IDS sering mengalami akurasi rendah dan false alarm tinggi akibat dimensi fitur besar dan class imbalance. PCA membantu reduksi dimensi, tapi gagal menangkap hubungan non-linear antar fitur sehingga masih menyebabkan error deteksi.	Mengusulkan model hybrid berbasis PCA + Variable Ensemble ML, dengan integrasi: (1) PCA+AdaBoost untuk mengatasi kelemahan PCA lewat exponential loss, (2) PCA+LogitBoost untuk multiclass lewat logistic loss, (3) PCA+Random Forest untuk mengurangi overfitting dengan bagging. Dataset: WSN-DS, NSL-KDD, UNSW-N15.	PCA+RF mencapai 100% akurasi pada semua dataset. PCA+Bagging: 99.8% (WSN-DS), 100% (NSL-KDD), 93.4% (UNSW-N15). PCA+AdaBoost lebih rendah: 92.3% (WSN-DS), 89.0% (NSL-KDD), 67.9% (UNSW-N15). Model terbaik: PCA+RF.
11	(Alalhareth & Hong, 2023)	An Improved Mutual Information Feature	IDS pada IoMT menghadapi tantangan utama berupa high-	1. Metode Baru (LRGU-MIFS): Mengusulkan algoritma *Logistic*	1. Reduksi Fitur Efektif: LRGU-MIFS berhasil mengidentifikasi

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
		Selection Technique for Intrusion Detection Systems in the Internet of Medical Things.	dimensional data dan fitur tidak relevan yang menurunkan akurasi serta meningkatkan beban komputasi. Mutual Information standar belum optimal karena redundansi antar fitur.	*Redundancy Coefficient Gradual Upweighting Mutual Information Feature Selection*. 2. Pendekatan Non-Linear: Mengganti penghitungan redundansi linear dengan Fungsi Logistik (Logistic Function) untuk menangkap hubungan non-linear antar fitur secara lebih akurat. 3. Mekanisme Seleksi: Menghitung skor redundansi fitur kandidat secara individual berdasarkan kurva logistik, bukan sekadar rata-rata linear. 4. Validasi Eksperimen: Diuji pada dataset WUSTL-EHMS-2020 (spesifik IoMT) dan CICIDS2017 menggunakan 5 model klasifikasi: Decision Tree (DT), Random Forest (RF), SVM, Naïve Bayes (NB), dan k-NN.	subset fitur yang jauh lebih kompak (sedikit) dibanding metode MI tradisional (MIFS, mRMR, JMIM) namun tetap mempertahankan informasi penting. 2. Akurasi Tinggi: Model Random Forest (RF) dan Decision Tree (DT) dengan fitur hasil seleksi LRGU mencapai akurasi tertinggi (rata-rata di atas 99%) dibanding SVM, NB, dan k-NN.
12	(Sazzed & Ullah, 2023)	Enhancing Efficiency and Privacy in Memory-Based Malware Classification through Feature Selection.	Malware classification berbasis memory dump menghadapi masalah: fitur sangat banyak, komputasi mahal, serta risiko privasi karena data sensitif ikut diproses. Feature selection	Mengusulkan pendekatan feature selection berbasis filter: Chi-Square, ANOVA, dan Mutual Information (MI). Model diuji pada 3 level klasifikasi: benign vs malware, malware type, dan	MI + Random Forest paling konsisten. Dengan hanya 25–50% fitur, performa tetap sama atau lebih baik dibanding 100% fitur. F1 malware detection mendekati 0.99–1.0. Untuk malware family classification: Trojan family

No	Peneliti/Tahun	Judul	Permasalahan	Solusi yang diusulkan	Hasil Penelitian
			masih jarang dieksplorasi di domain malware memory analysis.	malware family. Dataset: MalMemAnalysis-2022.	F1=0.75, Spyware F1=0.66, Ransomware F1=0.57. Feature reduction meningkatkan efisiensi dan privasi.
13	(Al-Sarem dkk., 2022)	An Aggregated Mutual Information Based Feature Selection with Machine Learning Methods for Enhancing IoT Botnet Attack Detection	Deteksi botnet IoT seperti Mirai dan Bashlite sulit karena dataset N-BaIoT sangat besar dan imbalance. Banyak fitur tidak relevan memperlambat training serta menurunkan generalisasi model.	Mengusulkan pendekatan Aggregated Mutual Information Feature Selection dikombinasikan dengan ML classifiers. Dataset diseimbangkan dengan under-sampling. Training/testing split 80/20. Fokus pada botnet detection biner benign vs attacked.	Metode MI aggregated meningkatkan akurasi dan efisiensi komputasi. Model lebih cepat dan lebih akurat dibanding tanpa seleksi fitur. Cocok untuk deteksi botnet skala besar pada IoT.
14	(Almaiah dkk., 2022)	Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels	SVM efektif untuk IDS, tapi performanya sangat tergantung pada pemilihan kernel. Dataset IDS juga berdimensi tinggi sehingga butuh reduksi fitur agar lebih efisien dan cepat.	Menginvestigasi kombinasi PCA + SVM dengan berbagai kernel: Linear, Polynomial, Sigmoid, dan RBF untuk memilih kernel terbaik. Dataset: KDD Cup'99 dan UNSW-NB15.	Kernel terbaik adalah Gaussian RBF. Akurasi: 99.11% (KDD Cup'99), 93.94% (UNSW-NB15). Sensitivitas dan F1 juga tertinggi pada RBF dibanding kernel lain.

Berdasarkan Tabel 2.1, penelitian terdahulu umumnya dapat dikelompokkan ke dalam dua fokus utama, yaitu pengembangan teknik reduksi dimensi untuk meningkatkan efisiensi dan kualitas fitur, serta pengembangan arsitektur deep learning untuk meningkatkan performa klasifikasi. Beberapa penelitian memanfaatkan Mutual Information (MI) sebagai metode seleksi fitur untuk memilih atribut yang paling relevan terhadap kelas, sementara penelitian lainnya menggunakan Principal Component Analysis (PCA) untuk mereduksi dimensi data melalui transformasi linier berbasis variansi. Terdapat pula penelitian yang mengombinasikan MI dan PCA dalam skema hybrid guna memanfaatkan keunggulan keduanya.

Meskipun demikian, sebagian penelitian yang mengimplementasikan teknik reduksi dimensi masih berfokus pada penggunaan classifier konvensional seperti Decision Tree, SVM, atau Random Forest, serta dievaluasi pada satu dataset tertentu. Salah satu penelitian yang melaporkan kombinasi MI dan PCA dalam skema hybrid adalah (Kaushik dkk., 2022) Namun, evaluasi yang dilakukan dalam penelitian tersebut masih difokuskan pada model Machine Learning dan belum mengkaji efektivitas kombinasi MI-PCA pada arsitektur deep learning berbasis sekuensial seperti GRU maupun dalam skenario multi-dataset.

Selain itu, penelitian deep learning lainnya cenderung berfokus pada peningkatan akurasi model tanpa melakukan analisis komparatif yang sistematis terhadap pengaruh penggunaan MI, PCA, dan Hybrid MI-PCA dalam skenario multi-dataset. Dengan

demikian, masih terdapat celah penelitian dalam analisis komparatif yang terstruktur antara baseline, MI, PCA, dan Hybrid terhadap performa GRU pada beberapa dataset IoT dengan karakteristik yang berbeda.

Untuk memperjelas posisi penelitian serta memetakan kombinasi pendekatan yang telah digunakan sebelumnya, diperlukan penyajian dalam bentuk matriks penelitian. Tabel 2.2 menyajikan matriks penelitian dari studi-studi terdahulu yang relevan dengan topik ini

Tabel 2. 2 Matriks penelitian

No	Peneliti/ Tahun	Judul	Ruang Lingkup																
			Algoritma											Teknik Reduksi Dimensi					
			GRU	LSTM	CNN	RNN	ANN	RANDOM FOREST	DECISION TREE	SVM	NAÏVE BAYES	XGBOOST	ENSEMBLE	MI	PCA	PEARSON	Chi-Square	ANOVA	GA/PSO
1	(Talukder dkk., 2025)	A hybrid machine learning model for intrusion detection in wireless sensor networks leveraging data balancing and dimensionality reduction						✓							✓				
2	(Shaikh dkk., 2025)	A deep Reinforcement learning-based robust Intrusion Detection System for securing IoMT Healthcare Networks		✓	✓									✓					
3	(Albalwy & Almohaimeed, 2025)	Advancing Artificial Intelligence of Things Security: Integrating Feature Selection and Deep Learning for Real-Time Intrusion Detection					✓								✓	✓			
4	(Sheheryar & Sharma, 2025)	Ensemble Feature Engineering and Deep Learning for Botnet Attacks Detection in the Internet of Things	✓	✓	✓									✓					
5	(Sagu dkk., 2025)	Advances to IoT security using a GRU-CNN deep learning model trained on SUCMO algorithm	✓		✓														
6	(G. & K., 2025)	A GRU-based approach for botnet detection using deep learning technique	✓														✓		
7	(Archana & Aneetha, 2024)	LSTM-MI: Revolutionizing Intrusion Detection Through Adaptive Learning and Mutual Information Analysis		✓										✓	✓				

No	Peneliti/ Tahun	Judul	Ruang Lingkup																
			Algoritma											Teknik Reduksi Dimensi					
			GRU	LSTM	CNN	RNN	ANN	RANDOM FOREST	DECISION TREE	SVM	NAÏVE BAYES	XGBOOST	ENSEMBLE	MI	PCA	PEARSON	Chi-Square	ANOVA	GA/PSO
8	(Balhareth & Ilyas, 2024)	Optimized Intrusion Detection for IoMT Networks with Tree-Based Machine Learning and Filter-Based Feature Selection						✓	✓			✓		✓					
9	(Shoab & Alsbatin, 2024)	GRU Enabled Intrusion Detection System for IoT Environment with Swarm Optimization and Gaussian Random Forest Classification	✓					✓			✓								✓
10	(John dkk., 2024)	Enhanced Intrusion Detection Model Based on PCA and Variable Ensemble Machine Learning Algorithm						✓							✓				
11	(Alalhareth & Hong, 2023)	An Improved Mutual Information Feature Selection Technique for Intrusion Detection Systems in the Internet of Medical Things.		✓				✓	✓	✓				✓					
12	(Sazzed & Ullah, 2023)	Enhancing Efficiency and Privacy in Memory-Based Malware Classification through Feature Selection.						✓						✓				✓	
13	(Al-Sarem dkk., 2022)	An Aggregated Mutual Information Based Feature Selection with Machine Learning Methods for Enhancing IoT Botnet Attack Detection										✓		✓					
14	(Almaiah dkk., 2022)	Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels								✓					✓				

Berdasarkan matriks penelitian yang terdapat pada Tabel 2.2, terlihat sebagian besar penelitian sebelumnya hanya membandingkan beberapa kombinasi teknik reduksi dimensi atau hanya menguji satu jenis dataset. Tidak ada studi yang secara sistematis membandingkan keempat skenario sekaligus (baseline, MI, PCA, dan Hybrid) pada model GRU dengan menggunakan beberapa dataset IoT. Oleh karena itu, penelitian ini mengevaluasi dampak dari setiap teknik reduksi dimensi pada model GRU. Tujuan penelitian ini adalah untuk memberikan pemahaman yang lebih lengkap mengenai efektivitas kombinasi MI dan PCA dalam mendeteksi malware IoT.