# TESIS

# MODEL INTRUSION DETECTION SYSTEM MENGGUNAKAN AUTOENCODER CNN-BILSTM DENGAN HYBRID RESAMPLING BERBASIS ANOMALY SCORE UNTUK PENANGANAN DATA TIDAK SEIMBANG


disusun oleh

**Nama** : **ARYA SATRYA WICAKSANA**
**NIM** : **24.55.1592**
**Konsentrasi** : **Business Intelligence**



**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS AMIKOM YOGYAKARTA**

**YOGYAKARTA**

**2026**

# TESIS

# MODEL INTRUSION DETECTION SYSTEM MENGGUNAKAN AUTOENCODER CNN-BILSTM DENGAN HYBRID RESAMPLING BERBASIS ANOMALY SCORE UNTUK PENANGANAN DATA TIDAK SEIMBANG

# A MODEL FOR AN INTRUSION DETECTION SYSTEM USING A CNN-BILSTM AUTOENCODER WITH HYBRID RESAMPLING BASED ON ANOMALY SCORES FOR HANDLING IMBALANCED DATA

Diajukan untuk memenuhi salah satu syarat mencapai derajat Pascasarjana

Program Studi S2 PJJ Informatika

disusun oleh


**Nama** : **ARYA SATRYA WICAKSANA**
**NIM** : **24.55.1592**
**Konsentrasi** : **Business Intelligence**


**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS AMIKOM YOGYAKARTA**

**YOGYAKARTA**

**2026**

**HALAMAN PERSETUJUAN**

**MODEL INTRUSION DETECTION SYSTEM MENGGUNAKAN AUTOENCODER CNN-BILSTM DENGAN HYBRID RESAMPLING BERBASIS ANOMALY SCORE UNTUK PENANGANAN DATA TIDAK SEIMBANG**

**A MODEL FOR AN INTRUSION DETECTION SYSTEM USING A CNN-BILSTM AUTOENCODER WITH HYBRID RESAMPLING BASED ON ANOMALY SCORES FOR HANDLING IMBALANCED DATA**

yang disusun dan diajukan oleh

**Arya Satrya wicaksana**

**24.55.1592**

telah disetujui oleh Dosen Pembimbing Tesis
pada tanggal 21 Mei 2026

**Dosen Pembimbing,**

**Robert Marco, S.T., M.T., Ph.D.**
**NIK. 190302228**

**HALAMAN PENGESAHAN**

**MODEL INTRUSION DETECTION SYSTEM MENGGUNAKAN AUTOENCODER CNN-BILSTM DENGAN HYBRID RESAMPLING BERBASIS ANOMALY SCORE UNTUK PENANGANAN DATA TIDAK SEIMBANG**

**A MODEL FOR AN INTRUSION DETECTION SYSTEM USING A CNN-BILSTM AUTOENCODER WITH HYBRID RESAMPLING BASED ON ANOMALY SCORES FOR HANDLING IMBALANCED DATA**

yang disusun dan diajukan oleh

**Arya Satrya Wicaksana**

**24.55.1592**

Telah dipertahankan di depan Dewan Penguji
pada tanggal 2 Juni 2026

**Susunan Dewan Penguji**

| Nama Penguji | Tanda Tangan |
|---|---|
| **Dr. Andi Sunyoto, S.Kom., M.Kom.**<br>**NIK. 190302052** | |
| **Dr. Kumara Ari Yuana, S.T., M.T.**<br>**NIK. 190302575** | |
| **Robert Marco, S.T., M.T., Ph.D.**<br>**NIK. 190302228** | |

Tesis ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Magister Komputer
Tanggal 22 Juni 2026

**DEKAN FAKULTAS ILMU KOMPUTER**

**Prof. Dr. Kusrini, M.Kom.**
**NIK. 190302106**

## HALAMAN PERNYATAAN KEASLIAN TESIS

Yang bertandatangan di bawah ini,

**Nama mahasiswa : Arya Satrya Wicaksana**
**NIM : 24.55.1592**

Menyatakan bahwa Tesis dengan judul berikut:

**Model Intrusion Detection System Menggunakan Autoencoder CNN-BiLSTM dengan Hybrid Resampling Berbasis Anomaly Score untuk Penanganan Data Tidak Seimbang**

Dosen Pembimbing : Robert Marco, S.T., M.T., Ph.D.

1. Karya tulis ini adalah benar-benar ASLI dan BELUM PERNAH diajukan untuk mendapatkan gelar akademik, baik di Universitas AMIKOM Yogyakarta maupun di Perguruan Tinggi lainnya.
2. Karya tulis ini merupakan gagasan, rumusan dan penelitian SAYA sendiri, tanpa bantuan pihak lain kecuali arahan dari Dosen Pembimbing.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan disebutkan dalam Daftar Pustaka pada karya tulis ini.
4. Perangkat lunak yang digunakan dalam penelitian ini sepenuhnya menjadi tanggung jawab SAYA, bukan tanggung jawab Universitas AMIKOM Yogyakarta.
5. Pernyataan ini SAYA buat dengan sesungguhnya, apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka SAYA bersedia menerima SANKSI AKADEMIK dengan pencabutan gelar yang sudah diperoleh, serta sanksi lainnya sesuai dengan norma yang berlaku di Perguruan Tinggi.

Yogyakarta, 21 Mei 2026

Yang Menyatakan,

Arya Satrya Wicaksana

# KATA PENGANTAR

Puji syukur dipanjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan tesis yang berjudul "Model Intrusion Detection System Menggunakan Autoencoder CNN-BiLSTM dengan Hybrid Resampling Berbasis Anomaly Score untuk Penanganan Data Tidak Seimbang". Penulisan tesis ini merupakan salah satu syarat untuk memperoleh gelar Magister di Universitas Amikom Yogyakarta.

Penulis menyadari bahwa terselesaikannya tesis ini tidak lepas dari bantuan, bimbingan, dan dukungan berbagai pihak. Oleh karena itu, disampaikan ucapan terima kasih yang sebesar-besarnya kepada:

1. Bapak Robert Marco, S.T., M.T., Ph.D., selaku Dosen Pembimbing yang telah meluangkan waktu, memberikan bimbingan, arahan, dan masukan yang sangat berharga selama proses penyusunan tesis ini.
2. Bapak Dr. Andi Sunyoto, S.Kom., M.Kom., selaku Dosen Penguji I yang telah memberikan koreksi, saran, dan masukan yang membangun dalam penyempurnaan tesis ini.
3. Bapak Dr. Kumara Ari Yuana, S.T., M.T., selaku Dosen Penguji II yang telah memberikan koreksi, saran, dan masukan yang membangun dalam penyempurnaan tesis ini.
4. Orang tua dan keluarga yang senantiasa memberikan doa, dukungan moral, dan semangat tiada henti selama menempuh pendidikan hingga terselesaikannya Tesis ini.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun demi perbaikan di masa mendatang. Semoga tesis ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan, khususnya di bidang keamanan jaringan dan pembelajaran mesin.

Yogyakarta, 2 Juni 2026

Penulis

## DAFTAR ISI

## DAFTAR TABEL

## DAFTAR GAMBAR

## DAFTAR LAMPIRAN

## DAFTAR LAMBANG DAN SINGKATAN

| | |
|---|---|
| $N_{max}$ | Jumlah sampel pada kelas terbesar dalam dataset |
| $N_c$ | Jumlah sampel pada kelas tertentu |
| $T_c$ | Target sampel kelas "c" setelah SMOTE |
| $\theta_{imbalance}$ | Nilai ambang batas (threshold) ketidakseimbangan data |
| $\mathcal{E}$ | Epsilon |
| AE | Autoencoder |
| CNN | Convolutional Neural Network |
| BiLSTM | Bidirectional Long Short-Term Memory |
| DNN | Deep Neural Network |
| DoS | Denial of Service |
| FAR | False Alarm Rate |
| FN | False Negative |
| FP | False Positive |
| FPR | False Positive Rate |
| IDS | Intrusion Detection System |
| IoT | Internet of Things |
| OHE | One-Hot Encoding |
| OVR | One-vs-Rest |
| Probe | Probing (Jenis serangan pemindaian jaringan) |
| R2L | Remote to Local |
| RE | Reconstruction Error |
| RO | Random Oversampling |
| RUS | Random Undersampling |
| SMOTE | Synthetic Minority Over-sampling Technique |
| SOTA | State-of-the-Art |
| TN | True Negative |
| TP | True Positive |
| U2R | User to Root |

## DAFTAR ISTILAH

| | |
|---|---|
| Anomaly Detection | Deteksi serangan berdasarkan penyimpangan dari pola normal jaringan |
| Bottleneck | Ruang dimensi rendah yang berisi ringkasan fitur hasil kompresi data |
| Confusion Matrix | Tabel evaluasi untuk mengukur ketepatan prediksi model (TP, FP, TN, FN) |
| Deep Learning | Cabang machine learning bertingkat banyak untuk mengenali pola data yang kompleks |
| False Negative (FN) | Serangan nyata yang salah terdeteksi sebagai lalu lintas normal (lolos) |
| False Positive (FP) | Lalu lintas normal yang salah terdeteksi sebagai serangan (alarm palsu) |
| Imbalanced Data | Kondisi tidak seimbang antara jumlah data normal (banyak) dan data serangan (sedikit) |
| Imbalance Ratio | Angka perbandingan jumlah data kelas minoritas terhadap kelas mayoritas |
| IDS | Sistem pemantau jaringan untuk mendeteksi aktivitas mencurigakan atau illegal |
| RUS | Cara menyeimbangkan data dengan menghapus sebagian data mayoritas secara acak |
| Specificity | Kemampuan model dalam mengidentifikasi lalu lintas normal secara tepat |
| StandardScaler | Metode menyamakan skala angka agar data memiliki rata-rata 0 dan standar deviasi 1 |
| SMOTE | Teknik menambah data minoritas dengan membuat data tiruan baru secara acak yang mirip |
| Zero-Day Attack | Jenis serangan siber baru yang belum pernah dikenal sebelumnya |

# INTISARI

Ancaman siber yang terus berkembang mendorong kebutuhan akan sistem deteksi intrusi yang mampu mendeteksi serangan secara akurat, khususnya pada kelas serangan yang sangat jarang terjadi. Dataset NSL-KDD yang umum digunakan sebagai *benchmark* IDS memiliki ketidakseimbangan kelas yang ekstrem, dengan kelas U2R hanya mencakup 0,04% dan R2L hanya 0,79% dari total data, sehingga model cenderung gagal mendeteksi serangan pada kelas minoritas meskipun mencapai *accuracy* yang tinggi. Penelitian ini mengusulkan model yang mengintegrasikan Autoencoder, CNN dan BiLSTM dengan mekanisme *Dual Attention* (FocusConv dan TempoNet), dikombinasikan dengan strategi Hybrid Resampling RUS-SMOTE berbasis *Anomaly Score*. *Anomaly score* diperoleh dari *reconstruction error* (RE) median per kelas hasil pelatihan Autoencoder secara *unsupervised*, yang digunakan untuk menentukan intensitas *oversampling* tiap kelas secara adaptif tanpa *hyperparameter* arbitrer. Kelas U2R dengan median RE sebesar 0,091631 dimana 53,9 kali lebih tinggi dari kelas Normal secara otomatis mendapatkan faktor pengali terbesar (14,35×), sehingga *imbalance ratio* berhasil direduksi dari 1.306× menjadi 36,45×. Evaluasi pada 7 skenario ablasi menggunakan data uji NSL-KDD (25.195 sampel) menunjukkan bahwa model final mencapai *accuracy* 99,31%, *Macro Sensitivity* 0,9757, *Macro Specificity* 0,9984, *Recall* R2L 0,9950, dan *Recall* U2R 0,9000 dengan *Score Komposit* 0,9871. Perbandingan dengan *existing methods* pada dataset yang sama menunjukkan peningkatan *Macro Sensitivity* sebesar 15,57% dibandingkan Dangol & Eaman (2025) dan peningkatan *Recall* R2L sebesar 25,19% dibandingkan Ben Said et al. (2023). Hasil ablasi membuktikan bahwa *anomaly score* berkontribusi meningkatkan *Recall* U2R sebesar 20,00% dibandingkan SMOTE rata tanpa *anomaly score*, mengkonfirmasi peran sentralnya sebagai mekanisme yang menjadikan strategi *resampling* lebih efisien dan dapat dipertanggungjawabkan secara ilmiah.

**Kata kunci:** Intrusion Detection System, Autoencoder, CNN-BiLSTM, Anomaly Score, Imbalanced Data, NSL-KDD

***ABSTRACT***

*The ever-evolving landscape of cyber threats drives the need for intrusion detection systems capable of accurately detecting attacks, particularly those that occur very rarely. The NSL-KDD dataset, commonly used as an IDS benchmark, exhibits extreme class imbalance, with the U2R class comprising only 0.04% and the R2L class only 0.79% of the total data, causing models to frequently fail to detect attacks in the minority classes despite achieving high accuracy. This study proposes a model that integrates an Autoencoder, CNN, and BiLSTM with a Dual Attention mechanism (FocusConv and TempoNet), combined with a Hybrid Resampling RUS-SMOTE strategy based on Anomaly Score. The Anomaly Score is derived from the median reconstruction error (RE) per class from the unsupervised training of the Autoencoder, which is used to adaptively determine the intensity of oversampling for each class without arbitrary hyperparameters. The U2R class, with a median RE of 0.091631 which is 53.9 times higher than the Normal class automatically receives the largest multiplier (14.35×), thereby successfully reducing the imbalance ratio from 1,306× to 36.45×. Evaluation across 7 ablation scenarios using the NSL-KDD (25,195 samples) shows that the final model achieves an accuracy of 99.31%, Macro Sensitivity of 0.9757, Macro Specificity of 0.9984, Recall R2L of 0.9950, and Recall U2R of 0.9000 with a Composite Score of 0.9871. Comparison with existing methods on the same dataset shows a 15.57% increase in Macro Sensitivity compared to Dangol & Eaman (2025) and a 25.19% increase in Recall R2L compared to Ben Said et al. (2023). Ablation results demonstrate that the anomaly score contributes to a 20.00% increase in U2R Recall compared to the baseline SMOTE without the anomaly score, confirming its central role as a mechanism that makes the resampling strategy more efficient and scientifically justifiable.*

***Keyword:*** *Intrusion Detection System, Autoencoder, CNN-BiLSTM, Anomaly Score, Imbalanced Data, NSL-KDD*

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Peningkatan intensitas dan kompleksitas ancaman siber menjadi tantangan utama dalam menjaga keamanan jaringan pada era transformasi digital saat ini. Perkembangan teknologi seperti Internet of Things (IoT), cloud computing, dan cyber-physical systems telah memperluas permukaan serangan, sehingga sistem menjadi lebih rentan terhadap berbagai jenis ancaman seperti Distributed Denial of Service (DDoS), serangan eksploitasi, dan intrusi berbasis malware. Sistem keamanan konvensional berbasis signature yang masih banyak digunakan memiliki keterbatasan dalam mendeteksi serangan baru (zero-day attack) karena hanya mampu mengenali pola serangan yang telah diketahui sebelumnya. Kondisi ini mendorong kebutuhan pengembangan Intrusion Detection System (IDS) berbasis machine learning dan deep learning yang mampu mempelajari pola serangan secara adaptif dan otomatis dari data jaringan (Nandanwar & Katarya, 2025; Wu et al., 2025).

Seiring perkembangan deep learning untuk IDS, berbagai arsitektur hybrid telah diuji secara empiris dan menunjukkan superioritas atas model tunggal. Halbouni et al. (2022) membuktikan bahwa arsitektur CNN-LSTM mencapai akurasi 93,78% dengan FAR hanya 6,0% pada UNSW-NB15 mengungguli model tunggal seperti DBN (85,77%) dan SVM (82,42%) karena CNN mampu mengekstraksi fitur spasial lokal secara otomatis sementara LSTM memodelkan

dependensi antar fitur secara sekuensial. Elsayed et al. (2024) dalam studi komparatif mendalam mengkonfirmasi bahwa model hybrid CNN-LSTM secara konsisten mencapai akurasi 98,94% pada NSL-KDD dan menjadi salah satu arsitektur paling kompetitif dalam deteksi intrusi multikelas. Temuan-temuan ini memperkuat posisi CNN sebagai feature extractor dan LSTM sebagai classifier sebagai pendekatan yang telah terbukti secara empiris dalam pengembangan IDS modern (Halbouni et al., 2022; Elsayed et al., 2024).

Namun demikian, LSTM konvensional hanya memproses urutan fitur dalam satu arah (forward), sehingga berpotensi kehilangan konteks relasi antar fitur dari arah sebaliknya. Yu et al. (2023) menunjukkan bahwa penggunaan arsitektur bidirectional pada unit rekuren dalam penelitian tersebut BiGRU menghasilkan akurasi 93,40% pada NSL-KDD dengan kemampuan menangkap pola dari dua arah sekaligus, mengatasi keterbatasan pemrosesan satu arah. Lebih lanjut, Zhang et al. (2025) secara eksplisit membuktikan bahwa arsitektur CNN-BiLSTM tanpa modifikasi tambahan pun telah mencapai akurasi 99,22%, detection rate 98,88%, dan FPR 0,43% pada NSL-KDD menjadikannya salah satu baseline yang kuat di antara model-model SOTA terkini. Selain itu, Hosseini et al. (2025) dalam studi komparatif multi-arsitektur pada NSL-KDD juga menempatkan CNN-BiLSTM dengan akurasi 99,79% sebagai salah satu arsitektur terkuat sebelum penambahan mekanisme attention. Berdasarkan bukti empiris tersebut, penelitian ini memilih CNN-BiLSTM sebagai arsitektur inti karena telah terbukti secara konsisten superior dibandingkan LSTM unidirectional konvensional, dengan kemampuan ekstraksi fitur spasial dari CNN dan pemodelan relasi fitur dua arah dari BiLSTM

yang saling melengkapi dalam konteks klasifikasi intrusi jaringan pada data multifitur NSL-KDD.

Tantangan lain yang signifikan dalam pengembangan IDS adalah ketidakseimbangan distribusi data (imbalanced data). Pada dataset benchmark NSL-KDD, kelas Normal memiliki 43.099 data (53,46%) dan DoS sebanyak 29.393 data (36,46%), sementara kelas minoritas seperti R2L hanya memiliki 637 data (0,79%) dan U2R hanya 33 data (0,04%). Rasio ketimpangan yang ekstrem ini menyebabkan model cenderung bias terhadap kelas mayoritas dan gagal mendeteksi serangan pada kelas minoritas, padahal serangan seperti U2R dan R2L justru memiliki tingkat risiko yang tinggi karena berpotensi memberikan akses tidak sah terhadap sistem (Zhu et al., 2023; Disha & Waheed, 2022). Kegagalan mendeteksi serangan (false negative) dalam konteks IDS memiliki konsekuensi yang jauh lebih serius dibandingkan false alarm (false positive), sehingga metrik sensitivity (recall) menjadi indikator kinerja yang paling kritis dalam evaluasi model IDS. Metode oversampling konvensional seperti SMOTE yang menerapkan jumlah sampel sintetis yang seragam untuk semua kelas minoritas belum cukup adaptif, karena tidak mempertimbangkan tingkat anomali masing-masing kelas (Elreedy et al., 2024). Selain itu, tanpa mekanisme undersampling pada kelas mayoritas, dominasi Normal dan DoS tetap signifikan meskipun kelas minoritas telah di-oversample.

Kegagalan mendeteksi serangan dalam konteks IDS bukan sekadar kesalahan klasifikasi biasa melainkan kegagalan sistem keamanan yang sesungguhnya. Dalam terminologi evaluasi model, kondisi ini disebut False

Negative (FN): serangan nyata yang tidak terdeteksi dan diklasifikasikan sebagai trafik normal oleh model. Konsekuensinya sangat serius sehingga penyerang dapat bergerak bebas di dalam jaringan, mengeksfiltrasi data sensitif, merusak infrastruktur, atau bahkan melakukan eskalasi serangan yang lebih luas tanpa sepengetahuan tim keamanan. Sebaliknya, False Positive (FP) merupakan trafik normal yang salah teridentifikasi sebagai serangan yang hanya mengakibatkan gangguan operasional berupa alarm palsu yang masih dapat ditangani secara manual. Asimetri konsekuensi antara FN dan FP inilah yang menjadikan recall (sensitivity) yang mengukur proporsi serangan nyata yang berhasil terdeteksi, yaitu TP/(TP+FN) sebagai metrik yang paling kritis dalam evaluasi IDS, bukan sekadar accuracy atau precision (Halbouni et al., 2022; Zhu et al., 2023).

Persoalan ini semakin diperparah oleh kondisi imbalanced data pada dataset NSL-KDD, di mana kelas Normal mendominasi 53,46% data. Pada kondisi ini, model yang hanya mengoptimalkan accuracy dapat mencapai nilai mendekati 99% hanya dengan selalu memprediksi trafik sebagai "normal" tanpa mendeteksi satu pun serangan. Ini menunjukkan bahwa accuracy bersifat misleading pada dataset yang tidak seimbang. Recall secara langsung mengukur seberapa baik model menjalankan fungsi utamanya sebagai sistem deteksi yaitu tidak membiarkan serangan lolos tanpa terdeteksi sehingga menjadi indikator performa yang paling relevan dan tidak terdistorsi oleh dominasi kelas mayoritas. Sebagian besar penelitian IDS yang ada masih menjadikan accuracy sebagai metrik primer (Elsayed et al., 2024), sehingga performa pada kelas minoritas seperti U2R dan R2L yang sesungguhnya berbahaya justru tertutupi oleh angka accuracy yang tinggi

namun tidak representatif. Penelitian ini secara eksplisit menjadikan peningkatan recall dan specificity sebagai target utama evaluasi, karena keduanya secara bersama-sama mencerminkan kemampuan model dalam mendeteksi serangan tanpa mengorbankan kemampuan mengenali trafik normal secara berlebihan.

Berdasarkan permasalahan tersebut, penelitian ini mengusulkan model IDS berbasis arsitektur Autoencoder CNN-BiLSTM yang mengintegrasikan dua kontribusi utama secara sinergis. Pertama, arsitektur Autoencoder CNN-BiLSTM dirancang agar Autoencoder berfungsi ganda: sebagai bagian dari pipeline feature extraction sekaligus sebagai pembangkit anomaly score berbasis reconstruction error. Pendekatan ini sejalan dengan temuan Yu et al. (2023) yang menunjukkan bahwa integrasi Autoencoder sebagai feature compressor dalam pipeline deep learning mampu mereduksi dimensionalitas data secara efektif dan meningkatkan akurasi deteksi pada dataset NSL-KDD. Selain itu, Elsayed et al. (2024) juga mencatat bahwa kombinasi CNN dan Autoencoder dalam arsitektur hybrid terbukti menghasilkan deteksi anomali yang lebih andal, di mana reconstruction error dari Autoencoder dimanfaatkan sebagai sinyal detektif terhadap pola data yang menyimpang dari distribusi normal.

Kedua, strategi Hybrid Resampling RUS-SMOTE dirancang secara adaptif, di mana Random Undersampling (RUS) diterapkan pada kelas mayoritas untuk mereduksi dominasi kelas Normal dan DoS, sementara SMOTE diterapkan secara selektif pada kelas minoritas dengan intensitas yang proporsional terhadap anomaly score dari Autoencoder. Hal ini dilandasi oleh temuan Bagui & Li (2021) yang membuktikan bahwa pada dataset IDS yang sangat tidak seimbang seperti KDD99,

kombinasi RUS-SMOTE secara konsisten menghasilkan macro recall tertinggi dibandingkan pendekatan oversampling maupun undersampling tunggal dengan kesimpulan bahwa strategi hybrid lebih efektif dalam meningkatkan kemampuan deteksi kelas minoritas tanpa mengorbankan performa kelas mayoritas secara signifikan. Selain itu, Hossen et al. (2023) juga menegaskan bahwa pendekatan hybrid resampling yang mempertimbangkan karakteristik distribusi tiap kelas menghasilkan keseimbangan yang lebih baik antara sensitivity dan specificity dibandingkan metode oversampling seragam.

Berbeda dengan penelitian sebelumnya yang menerapkan SMOTE dengan fixed threshold atau faktor pengali yang seragam untuk semua kelas minoritas (Elreedy et al., 2024; Mansotra et al., 2023), pendekatan dalam penelitian ini menggunakan reconstruction error median dari Autoencoder sebagai bobot adaptif yang menentukan intensitas SMOTE per kelas sehingga kelas dengan tingkat anomali lebih tinggi, seperti U2R yang memiliki reconstruction error tertinggi (RE median = 0,089), mendapatkan penguatan data yang lebih proporsional. Selain itu, temuan dari penelitian impact of resampling techniques in deep learning-based intrusion detection menunjukkan bahwa strategi hybrid yang menggabungkan oversampling sebelum undersampling menghasilkan performa terbaik pada dataset NSL-KDD, dengan akurasi 81,63% dan recall 0,82 pada kondisi ADASYN + TomekLinks mengkonfirmasi bahwa urutan dan selektivitas teknik resampling berpengaruh signifikan terhadap hasil akhir. Dengan menggunakan dataset benchmark NSL-KDD, penelitian ini mengevaluasi peningkatan performa model secara komprehensif melalui metrik accuracy, sensitivity, dan specificity, dengan

penekanan pada kemampuan model dalam mendeteksi serangan pada kelas minoritas R2L dan U2R secara lebih andal.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian yang menjadi masalah utama dalam penelitian ini adalah:

1. Apakah arsitektur Autoencoder CNN-BiLSTM mampu meningkatkan performa Intrusion Detection System dalam mendeteksi pola serangan jaringan pada dataset NSL-KDD dibandingkan model baseline?
2. Bagaimana pengaruh ketidakseimbangan data pada dataset NSL-KDD terhadap kemampuan model dalam mendeteksi kelas serangan minoritas, khususnya R2L dan U2R?
3. Apakah penerapan Hybrid Resampling RUS-SMOTE berbasis Anomaly Score dari Autoencoder mampu meningkatkan nilai sensitivity dan specificity model Intrusion Detection System pada data tidak seimbang?

## 1.3 Batasan Masalah

1. Dataset yang digunakan dibatasi pada dataset benchmark NSL-KDD yang terdiri dari lima kelas klasifikasi, yaitu Normal, DoS, Probe, R2L, dan U2R, sebagai representasi lalu lintas jaringan normal dan beragam jenis serangan.
2. Tahapan preprocessing meliputi pemetaan label ke lima kelas utama,

konversi fitur kategorikal menggunakan One-Hot Encoding, normalisasi menggunakan StandardScaler, serta penanganan ketidakseimbangan data menggunakan metode Hybrid Resampling RUS-SMOTE berbasis anomaly score dari Autoencoder.

3. Arsitektur model yang digunakan adalah Autoencoder CNN-BiLSTM, di mana Autoencoder berperan ganda sebagai pembangkit anomaly score untuk memandu proses resampling sekaligus sebagai bagian dari arsitektur classifier bersama CNN dan BiLSTM.
4. Model dibangun dan diuji menggunakan data dari KDDTrain+ dengan pembagian Train:Validation:Test sebesar 64%:16%:20% secara stratifikasi, tanpa pengujian secara real-time pada jaringan nyata.
5. Evaluasi performa model difokuskan pada metrik accuracy, sensitivity (recall), specificity, precision, F1-score, dan False Positive Rate (FPR) menggunakan confusion matrix, dengan penekanan utama pada sensitivity dan specificity sebagai indikator kinerja deteksi serangan.

## 1.4 Tujuan Penelitian

Tujuan umum dari penelitian ini adalah untuk Membangun model Intrusion Detection System berbasis Autoencoder CNN-BiLSTM dengan Hybrid Resampling RUS-SMOTE untuk meningkatkan performa deteksi serangan jaringan pada kondisi data tidak seimbang, dengan penekanan pada peningkatan nilai accuracy, sensitivity, dan specificity pada dataset NSL-KDD.. Secara khusus, tujuan penelitian ini adalah:

1. Menganalisis dan mengevaluasi performa arsitektur Autoencoder CNN-BiLSTM dalam meningkatkan kemampuan Intrusion Detection System untuk mendeteksi pola serangan jaringan pada dataset NSL-KDD dibandingkan dengan model baseline.
2. Menganalisis pengaruh ketidakseimbangan data pada dataset NSL-KDD terhadap kemampuan model dalam mendeteksi kelas serangan minoritas, khususnya R2L dan U2R.
3. Mengevaluasi efektivitas penerapan Hybrid Resampling RUS-SMOTE berbasis Anomaly Score dari Autoencoder dalam meningkatkan nilai sensitivity dan specificity model Intrusion Detection System pada kondisi data tidak seimbang.

## 1.5 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah :

1. Penelitian ini memberikan kontribusi pada pengembangan ilmu pengetahuan di bidang keamanan jaringan berbasis deep learning, khususnya dalam pemanfaatan arsitektur Autoencoder CNN-BiLSTM pada sistem Intrusion Detection System. Hasil penelitian ini memperkaya pemahaman mengenai peran ganda Autoencoder sebagai feature extractor sekaligus pembangkit anomaly score dalam konteks klasifikasi intrusi jaringan pada data tidak seimbang.
2. Penelitian ini mengusulkan pendekatan Hybrid Resampling RUS-SMOTE berbasis Anomaly Score dari Autoencoder sebagai strategi penanganan

ketidakseimbangan data yang lebih adaptif dibandingkan metode oversampling konvensional. Pendekatan ini dapat menjadi referensi metodologis bagi penelitian selanjutnya yang menghadapi permasalahan serupa pada dataset IDS maupun domain klasifikasi lain dengan distribusi kelas yang tidak seimbang.

3. Penelitian ini mengusulkan pendekatan Hybrid Resampling RUS-SMOTE berbasis Anomaly Score dari Autoencoder sebagai strategi penanganan ketidakseimbangan data yang lebih adaptif dibandingkan metode oversampling konvensional. Pendekatan ini dapat menjadi referensi metodologis bagi penelitian selanjutnya yang menghadapi permasalahan serupa pada dataset IDS maupun domain klasifikasi lain dengan distribusi kelas yang tidak seimbang.

# BAB 2
# TINJAUAN PUSTAKA

## 2.1 Tinjauan Pustaka

Penelitian mengenai Intrusion Detection System (IDS) terus berkembang seiring meningkatnya kompleksitas ancaman siber dan keterbatasan metode keamanan konvensional dalam mendeteksi serangan baru. Dalam kajian yang dilakukan oleh Halbouni et al. (2022), pendekatan machine learning dan deep learning dinyatakan memiliki peran penting dalam meningkatkan kemampuan sistem keamanan siber, khususnya karena model mampu mempelajari pola serangan secara otomatis dari data jaringan. Studi tersebut menegaskan bahwa deep learning lebih unggul dibandingkan pendekatan konvensional dalam mengenali pola serangan yang kompleks, termasuk serangan yang sulit dideteksi oleh sistem berbasis signature. Temuan ini menunjukkan bahwa pengembangan IDS modern semakin mengarah pada pemanfaatan arsitektur deep learning sebagai inti proses deteksi intrusi (Halbouni et al., 2022b).

Pada tingkat yang lebih khusus, penelitian oleh Halbouni et al. (2022) mengusulkan model hybrid CNN-LSTM untuk deteksi intrusi jaringan dan menunjukkan bahwa kombinasi CNN dan LSTM mampu menghasilkan performa yang lebih baik dibandingkan beberapa pendekatan lain pada dataset CIC-IDS2017, UNSW-NB15, dan WSN-DS. Dalam penelitian tersebut, CNN dimanfaatkan untuk mengekstraksi fitur penting dari data jaringan, sedangkan LSTM digunakan untuk mempelajari hubungan pola yang lebih kompleks sehingga akurasi dan detection

rate meningkat serta false alarm rate dapat ditekan. Hasil ini memperlihatkan bahwa integrasi CNN-LSTM merupakan pendekatan yang menjanjikan untuk meningkatkan performa IDS berbasis deep learning (Halbouni et al., 2022a).

Selain CNN-LSTM, penelitian lain juga menunjukkan bahwa arsitektur hybrid mampu meningkatkan performa klasifikasi serangan siber. Yu et al. (2023) mengembangkan metode deteksi intrusi berbasis hybrid improved residual network blocks dan Bidirectional Gated Recurrent Units (BiGRU), yang terbukti mampu meningkatkan akurasi deteksi pada dataset NSL-KDD dan UNSW-NB15. Penelitian ini menegaskan bahwa penggabungan dua arsitektur jaringan saraf dapat mengatasi keterbatasan model tunggal dalam memahami pola data jaringan yang kompleks. Meskipun demikian, pendekatan tersebut menggunakan konfigurasi model yang lebih kompleks, sehingga masih terbuka peluang untuk mengevaluasi arsitektur hybrid lain yang lebih sederhana namun tetap efektif, seperti CNN-LSTM (Yu et al., 2023).

Dari sisi penanganan data tidak seimbang, Zhu et al. (2023) mengemukakan bahwa ketidakseimbangan kelas merupakan salah satu penyebab utama rendahnya kemampuan model dalam mendeteksi serangan pada kelas minoritas. Melalui pendekatan AEC_GAN pada data serangan jaringan, penelitian ini menunjukkan bahwa perbaikan distribusi data mampu meningkatkan akurasi, recall, dan F1-score model, terutama pada kelas yang jumlah sampelnya sangat sedikit. Temuan tersebut memperkuat pandangan bahwa penanganan imbalanced data merupakan tahap penting dalam pengembangan IDS, sebab model yang dilatih pada distribusi data

tidak seimbang cenderung bias terhadap kelas mayoritas dan gagal mendeteksi serangan kritis pada kelas minoritas (Zhu et al., 2023).

Dari sisi penanganan imbalanced data, Zhu et al. (2023) mengemukakan bahwa ketidakseimbangan kelas merupakan salah satu penyebab utama rendahnya kemampuan model dalam mendeteksi serangan pada kelas minoritas. Melalui pendekatan AEC-GAN pada data serangan jaringan, penelitian ini menunjukkan bahwa perbaikan distribusi data mampu meningkatkan recall dan F1-score model, terutama pada kelas yang jumlah sampelnya sangat sedikit. Sejalan dengan hal ini, Bagui & Li (2021) melakukan studi komparatif secara sistematis terhadap berbagai teknik resampling pada dataset IDS yang sangat tidak seimbang, termasuk KDD99, dan menyimpulkan bahwa kombinasi Random Undersampling dan SMOTE (RU-SMOTE) secara konsisten menghasilkan macro recall tertinggi dibandingkan pendekatan oversampling maupun undersampling tunggal. Temuan ini memperkuat pandangan bahwa strategi hybrid yang menggabungkan kedua pendekatan lebih efektif dalam meningkatkan kemampuan deteksi kelas minoritas tanpa mengorbankan performa kelas mayoritas secara signifikan.

Penelitian lain oleh Hnamte et al. (2023) mengembangkan model dua tahap LSTM-AE untuk deteksi intrusi jaringan dan menunjukkan bahwa pendekatan hybrid yang menggabungkan Autoencoder dan LSTM mampu memberikan hasil yang baik dalam klasifikasi serangan multikelas. Namun, model tersebut lebih berorientasi pada kombinasi Autoencoder dan LSTM untuk reduksi noise, tanpa memanfaatkan CNN sebagai feature extractor dan tanpa mengintegrasikan

reconstruction error dari Autoencoder sebagai sinyal adaptif dalam proses resampling.

Penelitian lain oleh Hnamte et al. (2023) mengembangkan model dua tahap LSTM-AE untuk deteksi intrusi jaringan dan menunjukkan bahwa pendekatan hybrid yang menggabungkan Autoencoder dan LSTM mampu memberikan hasil yang baik dalam klasifikasi serangan multikelas. Namun, model tersebut lebih berorientasi pada kombinasi Autoencoder dan LSTM untuk reduksi noise, tanpa memanfaatkan CNN sebagai feature extractor dan tanpa mengintegrasikan reconstruction error dari Autoencoder sebagai sinyal adaptif dalam proses resampling.

## 2.2 Keaslian Penelitian

Tabel 2.1 Matriks literatur review dan posisi penelitian
Model Intrusion Detection System Menggunakan Autoencoder CNN-BiLSTM
dengan Hybrid Resampling Berbasis Anomaly Score untuk Penanganan Data Tidak Seimbang

| No | Judul Penelitian | Nama Peneliti, Tahun, Index | Metode Penelitian | Dataset | Hasil | Keunggulan dan Kelemahan | Perbandingan |
|---|---|---|---|---|---|---|---|
| 1 | Network Intrusion Detection Method Based on Hybrid Improved Residual Network Blocks and Bidirectional Gated Recurrent Units | H. Yu et al., IEEE Access, 2023 | ResNet + BiGRU + Autoencoder | NSL-KDD dan UNSW-NB15 | Akurasi 93,40% pada NSL-KDD dan 93,26% pada UNSW-NB15 dengan recall tinggi | Arsitektur bidirectional terbukti efektif; Autoencoder digunakan sebagai feature compressor. Namun Autoencoder tidak dimanfaatkan sebagai pembangkit anomaly score untuk memandu resampling | Penelitian ini mengintegrasikan Autoencoder untuk dua peran sekaligus: feature extraction dan pembangkit anomaly score berbasis reconstruction error untuk Hybrid RUS-SMOTE adaptif, berbeda dengan penelitian tersebut yang menggunakan ResNet-BiGRU-AE tanpa adaptive resampling |
| 2 | A Comparative Study of Using Deep Learning Algorithms in Network Intrusion Detection | S. Elsayed et al., IEEE Access, 2024 | DNN, CNN, RNN, LSTM, GRU, CNN-LSTM | NSL-KDD | Multiclass NSL-KDD: LSTM terbaik (ACC 99,39%, Recall 99,39%), CNN-LSTM ACC 97,96%, Recall 97,79%. Deteksi U2R masih menjadi tantangan di semua model | Studi komparatif komprehensif 6 arsitektur di NSL-KDD; menegaskan pentingnya pemilihan arsitektur berbasis dataset. Namun tidak ada penanganan imbalanced data dan tidak | Penelitian ini mengembangkan AE-CNN-BiLSTM dengan Hybrid RUS-SMOTE berbasis Anomaly Score untuk mengatasi kelemahan CNN-LSTM konvensional yang diidentifikasi Elsayed et al., khususnya dalam deteksi kelas U2R dan R2L yang |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | menggunakan BiLSTM atau Autoencoder | masih lemah tanpa penanganan imbalanced data |
| 3 | Impact of Resampling Techniques in Deep Learning Based Intrusion Detection: A Comparative Study on NSL-KDD and UNSW-NB15 | N. Dangol & A. Eaman, Procedia Computer Science, 2025 | CNN-BLSTM + ADASYN, RUS, TomekLinks, OSS | NSL-KDD dan UNSW-NB15 | Hasil terbaik NSL-KDD: ACC 81,63%, Recall 0,82 (ADASYN+TomekLinks). Akurasi U2R hanya 25,37% pada strategi terbaik | Membuktikan bahwa hybrid resampling lebih baik dari single strategy; CNN-BLSTM arsitektur yang sama diuji secara sistematis. Namun intensitas resampling bersifat fixed dan tidak berbasis karakteristik anomali kelas | Penelitian ini menggunakan arsitektur yang sama (CNN-BiLSTM) namun menggantikan strategi resampling fixed dengan Hybrid RUS-SMOTE berbasis Anomaly Score dari Autoencoder, yang diharapkan mengoptimalkan potensi arsitektur CNN-BiLSTM yang sesungguhnya pada NSL-KDD |
| 4 | Hybrid Intrusion Detection System Optimization Using Enhanced Deep Learning Approach | Anon. et al., SCEECS, 2026 | RF + XGBoost + BiLSTM + Logistic Regression (Stacked Ensemble) | NSL-KDD dan CICIDS-2017 | Macro-F1 87,2% pada NSL-KDD (ensemble); BiLSTM saja 83,7%. Macro-F1 98,0% pada CICIDS-2017 | Pendekatan ensemble memanfaatkan kekuatan beberapa model secara bersamaan. Namun performa pada NSL-KDD masih terbatas, tidak menggunakan Autoencoder, dan tidak ada strategi resampling adaptif | Penelitian ini menggunakan arsitektur tunggal AE-CNN-BiLSTM yang lebih terfokus dengan Hybrid RUS-SMOTE berbasis Anomaly Score, yang diharapkan melampaui performa ensemble pada NSL-KDD dengan strategi penanganan imbalanced data yang lebih adaptif |
| 5 | CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System | A. Halboumi et al., IEEE Access, 2022 | CNN-LSTM | CIC-IDS2017, UNSW-NB15 dan WSN-DS | ACC hingga 99,64% pada CIC-IDS2017, FAR 6,0% pada UNSW-NB15. Detection rate dan FAR lebih baik dari SVM, DBN, MLP | Performa tinggi dan stabil; membuktikan CNN-LSTM sebagai arsitektur hybrid yang kuat. Namun belum fokus pada penanganan imbalanced data dan tidak | Penelitian ini mengembangkan CNN-LSTM ke CNN-BiLSTM dengan penambahan Autoencoder dan Hybrid RUS-SMOTE berbasis Anomaly Score, mengisi |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | menggunakan BiLSTM atau Autoencoder | celah penanganan imbalanced data yang belum ditangani Halbouni et al. |
| 6 | Resampling Imbalanced Data for Network Intrusion Detection Datasets | S. Bagui & K. Li, Journal of Big Data, 2021 | RU, RO, RURO, RU-SMOTE, RU-ADASYN pada KDD99, UNSW-NB15, NB17 dengan ANN | KDD99, UNSW-NB15 dan NB17 | Pada KDD99: RU-SMOTE dan RURO secara konsisten menghasilkan macro recall tertinggi dibandingkan semua strategi lain | Studi sistematis komparatif yang komprehensif; membuktikan superioritas RU-SMOTE untuk dataset IDS sangat tidak seimbang. Namun tidak menggunakan deep learning modern dan tidak ada mekanisme adaptif berbasis anomaly score | Penelitian ini mengadopsi justifikasi RU-SMOTE dari Bagui & Li sebagai landasan ilmiah pemilihan strategi resampling hybrid, dengan peningkatan berupa penentuan intensitas SMOTE secara adaptif berbasis anomaly score dari Autoencoder |
| 7 | AEC-GAN: Unbalanced Data Processing Decision-Making in Network Attacks Based on ACGAN and Machine Learning | N. Zhu et al., IEEE Access, 2023 | GAN + ML | UNSW-NB15 | Recall meningkat signifikan hingga >0,90 pada beberapa kelas dengan peningkatan F1-score | Sangat efektif untuk imbalance; pendekatan generatif mampu menghasilkan data sintetis berkualitas tinggi. Namun pelatihan GAN kompleks, mahal secara komputasi, dan tidak menggunakan arsitektur CNN-BiLSTM | Penelitian ini menggunakan SMOTE yang lebih efisien secara komputasi dengan panduan adaptif dari Anomaly Score Autoencoder, berbeda dengan GAN yang lebih kompleks dalam proses generasi data sintetis |
| 8 | A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE | V. Hnamte et al., IEEE Access, 2023 | LSTM + Autoencoder | CIC-IDS2018 | Akurasi 99.10% dan F1-score tinggi dalam deteksi intrusi multikelas | Hybrid AE-LSTM efektif dalam reduksi noise dan meningkatkan klasifikasi. Namun tidak memanfaatkan CNN sebagai feature extractor dan tidak mengintegrasikan | Penelitian ini menggunakan CNN sebagai feature extractor dan AE sebagai pembangkit anomaly score sekaligus bagian arsitektur, berbeda dengan LSTM-AE yang tidak memanfaatkan rekonstruksi error secara |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | | reconstruction error AE untuk adaptive resampling | adaptif dalam proses resampling |
| 9 | Network Intrusion Detection with Two-Phased Hybrid Ensemble Learning and Automatic Feature Selection | A. K. Mananayaka et al., IEEE Access, 2023 | Ensemble + Feature Selection (THE-AFS) | NSL-KDD dan AWID | Detection rate hingga 94% dengan FPR rendah (~0,005) | Akurasi tinggi dengan feature selection; ensemble meningkatkan robustness. Namun kompleks, berisiko kehilangan fitur penting, dan tidak ada penanganan imbalanced data | Penelitian ini menggunakan CNN untuk ekstraksi fitur otomatis tanpa feature selection manual, dengan penanganan imbalanced data melalui Hybrid RUS-SMOTE berbasis Anomaly Score yang tidak ada pada penelitian tersebut |
| 10 | Early Detection of the Advanced Persistent Threat Attack | J. H. Joloudari et al., Journal of Information Security and Applications, 2021 | Deep Learning (6-layer NN) | NSL-KDD | ACC 98,85%, FPR 1,13%, AUC 99,9% pada NSL-KDD | Efektif untuk APT detection dengan akurasi tinggi. Namun tidak menggunakan arsitektur hybrid dan tidak ada penanganan imbalanced data secara eksplisit | Penelitian ini menggunakan arsitektur hybrid AE-CNN-BiLSTM dengan Hybrid RUS-SMOTE berbasis Anomaly Score, menawarkan pendekatan yang lebih komprehensif dan adaptif dibandingkan model deep learning tunggal |

## 2.3 Landasan Teori

1. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) merupakan sistem keamanan siber yang berfungsi untuk mendeteksi aktivitas jaringan yang mencurigakan atau tidak sah dalam suatu sistem informasi. IDS bekerja dengan cara menganalisis lalu lintas jaringan dan membandingkannya dengan pola normal maupun pola serangan yang telah diketahui. Secara umum, IDS dibagi menjadi dua pendekatan utama, yaitu signature-based detection dan anomaly-based detection. Pendekatan signature-based hanya mampu mendeteksi serangan yang telah dikenal sebelumnya, sedangkan anomaly-based memiliki kemampuan untuk mendeteksi serangan baru berdasarkan penyimpangan pola data.

Seiring meningkatnya kompleksitas serangan siber, pendekatan berbasis machine learning dan deep learning semakin banyak digunakan karena mampu mempelajari pola serangan secara otomatis dan adaptif. Model deep learning seperti CNN dan LSTM terbukti mampu meningkatkan kemampuan deteksi serta menurunkan tingkat kesalahan klasifikasi dibandingkan metode konvensional (Halbouni et al., 2022; Yu et al., 2023).

2. Random Undersampling (RUS)

Random Undersampling (RUS) merupakan teknik penyeimbangan data dengan cara mengurangi jumlah sampel pada kelas mayoritas secara acak. Teknik ini bekerja dengan memilih secara acak sejumlah sampel dari kelas

mayoritas dan menghapusnya dari dataset pelatihan, sehingga rasio ketimpangan antar kelas berkurang. Keunggulan RUS terletak pada efisiensinya secara komputasi dan kemampuannya mereduksi dominasi kelas mayoritas yang berlebihan.

Bagui & Li (2021) membuktikan bahwa kombinasi RUS dengan SMOTE (RU-SMOTE) secara konsisten menghasilkan macro recall tertinggi dibandingkan teknik oversampling atau undersampling tunggal pada dataset IDS yang sangat tidak seimbang seperti KDD99. Dalam penelitian ini, RUS diterapkan pada kelas mayoritas (Normal dan DoS) dengan faktor RUS_FACTOR = 0,60 untuk mereduksi dominasinya tanpa menghilangkan informasi penting, sebelum SMOTE diterapkan secara adaptif pada kelas minoritas.

3. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE merupakan metode oversampling yang digunakan untuk mengatasi ketidakseimbangan data dengan menghasilkan data sintetis pada kelas minoritas. Metode ini bekerja dengan melakukan interpolasi antara satu sampel minoritas dengan tetangga terdekatnya dalam ruang fitur, sehingga dapat meningkatkan representasi kelas minoritas tanpa sekadar menduplikasi data.

Pendekatan ini sangat penting dalam konteks IDS, karena dataset seperti NSL-KDD memiliki distribusi data yang tidak seimbang, di mana kelas seperti U2R dan R2L memiliki jumlah sampel yang sangat sedikit. Dengan

menggunakan SMOTE, model dapat belajar lebih baik terhadap pola serangan minoritas, sehingga meningkatkan nilai recall (sensitivity) tanpa mengorbankan specificity secara signifikan (Elreedy et al., 2024).

4. One-Hot Encoding (OHE)

One-Hot Encoding (OHE) merupakan teknik transformasi data kategorikal menjadi representasi numerik yang dapat diproses oleh algoritma machine learning dan deep learning. Pada metode ini, setiap kategori diubah menjadi vektor biner, di mana hanya satu elemen bernilai 1 dan sisanya bernilai 0.

Dalam dataset IDS seperti NSL-KDD, terdapat fitur kategorikal seperti protocol_type, service, dan flag yang tidak dapat langsung diproses oleh model. Oleh karena itu, digunakan One-Hot Encoding untuk mengubah fitur tersebut menjadi bentuk numerik tanpa memberikan bobot urutan tertentu pada kategori. Pendekatan ini penting untuk menghindari bias model terhadap nilai kategori tertentu serta meningkatkan kualitas representasi data dalam proses pelatihan model.

5. StandardScaler

StandardScaler merupakan metode normalisasi data yang digunakan untuk mengubah distribusi fitur numerik agar memiliki nilai rata-rata (mean) sebesar 0 dan standar deviasi sebesar 1. Proses ini bertujuan untuk menyamakan skala antar fitur sehingga model tidak bias terhadap fitur dengan nilai yang lebih besar.

Normalisasi sangat penting dalam model deep learning seperti CNN dan

LSTM, karena perbedaan skala fitur dapat mempengaruhi proses pembelajaran dan memperlambat konvergensi model. Dengan menggunakan StandardScaler, setiap fitur akan memiliki distribusi yang seragam sehingga meningkatkan stabilitas dan performa model dalam proses pelatihan.

6. Deep Learning

Deep learning merupakan cabang dari machine learning yang menggunakan jaringan saraf tiruan berlapis (neural networks) untuk mempelajari representasi data secara otomatis. Berbeda dengan metode tradisional, deep learning mampu mengekstraksi fitur secara langsung dari data mentah tanpa memerlukan proses feature engineering manual. Dalam bidang keamanan jaringan, deep learning telah banyak digunakan untuk meningkatkan performa Intrusion Detection System (IDS), karena mampu mengenali pola serangan yang kompleks dan tidak linear (Halbouni et al., 2022).

7. Hybrid Deep Learning

Hybrid deep learning merupakan pendekatan yang menggabungkan dua atau lebih arsitektur jaringan saraf untuk memanfaatkan keunggulan masing-masing model. Pendekatan ini banyak digunakan dalam berbagai bidang, termasuk deteksi intrusi jaringan, karena mampu meningkatkan performa klasifikasi dengan menggabungkan kemampuan ekstraksi fitur dan pemodelan pola data. Dalam IDS, hybrid model seperti CNN-LSTM digunakan untuk menangkap pola spasial dan hubungan antar fitur secara

bersamaan, sehingga meningkatkan akurasi deteksi dan mengurangi kesalahan klasifikasi.

8. Autoencoder

Autoencoder merupakan arsitektur neural network berbasis unsupervised learning yang terdiri dari dua komponen utama: encoder dan decoder. Encoder mengompresi data input menjadi representasi berdimensi lebih rendah (bottleneck atau latent space), sedangkan decoder merekonstruksi kembali data dari representasi tersebut mendekati data aslinya. Fungsi loss yang digunakan adalah reconstruction error (RE), yaitu selisih antara data input asli dengan hasil rekonstruksinya.

Dalam konteks IDS, Autoencoder memiliki dua peran yang saling melengkapi. Pertama, sebagai feature compressor yang menghasilkan representasi fitur berkualitas tinggi dan tereduksi dimensinya sebelum masuk ke classifier. Yu et al. (2023) membuktikan bahwa integrasi Autoencoder sebagai feature compressor dalam pipeline deep learning mampu mereduksi dimensionalitas data secara efektif dan meningkatkan akurasi deteksi pada dataset NSL-KDD. Kedua, sebagai anomaly scoring module yaitu Autoencoder yang dilatih secara unsupervised akan mempelajari distribusi representasi data yang dominan, sehingga data yang polanya menyimpang dari distribusi tersebut akan menghasilkan reconstruction error (RE) yang lebih tinggi. Elsayed et al. (2024) mengkonfirmasi bahwa RE dari Autoencoder dalam arsitektur hybrid dapat dimanfaatkan sebagai sinyal detektif terhadap pola data yang menyimpang

dari distribusi normal, memperkuat validitas RE sebagai ukuran anomali yang andal.

Peran kedua Autoencoder sebagai anomaly scoring module inilah yang menjadi landasan penggunaannya dalam penelitian ini untuk memandu proses SMOTE secara adaptif. Nilai RE median per kelas digunakan sebagai proksi kuantitatif tingkat anomali kelas dengan RE median tertinggi berarti polanya paling asing bagi model, sehingga secara logis memerlukan penguatan data sintetis terbesar agar classifier dapat mempelajari batas keputusannya secara efektif. Zhu et al. (2023) membuktikan melalui pendekatan AEC-GAN bahwa representasi Autoencoder mampu menangkap karakteristik distribusi kelas minoritas pada dataset IDS yang sangat tidak seimbang dan informasi tersebut dapat digunakan untuk menghasilkan data sintetis yang lebih representatif. Hnamte et al. (2023) juga menunjukkan bahwa arsitektur hybrid yang mengintegrasikan Autoencoder dengan model sekuensial mampu meningkatkan performa deteksi pada kelas minoritas, mengkonfirmasi relevansi Autoencoder dalam pipeline penanganan imbalanced data untuk IDS. Pendekatan penggunaan RE Autoencoder sebagai panduan SMOTE ini merupakan pengembangan dari justifikasi Bagui & Li (2021) yang membuktikan superioritas strategi RU-SMOTE untuk IDS, namun menggantinya dari faktor seragam menjadi faktor adaptif berbasis anomaly score sehingga kelas yang lebih anomali secara otomatis mendapatkan penguatan resampling yang lebih besar tanpa hyperparameter arbitrer.

9. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan arsitektur deep learning yang dirancang untuk mengekstraksi fitur secara otomatis melalui operasi konvolusi. CNN bekerja dengan menggunakan filter (kernel) untuk menangkap pola lokal dalam data, sehingga mampu mengidentifikasi fitur penting tanpa memerlukan proses seleksi fitur manual.

Dalam konteks IDS, CNN digunakan untuk mengekstraksi pola dari data tabular jaringan, seperti hubungan antar atribut pada dataset NSL-KDD. Proses ini membantu mengurangi kompleksitas fitur sekaligus meningkatkan kualitas representasi data sebelum dilakukan klasifikasi. Dengan demikian, CNN berperan sebagai feature extractor yang sangat penting dalam arsitektur CNN-LSTM (Halbouni et al., 2022a).

10. Bidirectional Long Short-Term Memory (BiLSTM)

Long Short-Term Memory (LSTM) merupakan pengembangan dari Recurrent Neural Network (RNN) yang dirancang untuk mengatasi masalah vanishing gradient dalam pemrosesan data sekuensial. LSTM memiliki mekanisme gates (input gate, forget gate, dan output gate) yang memungkinkan model menyimpan dan mengatur informasi secara selektif dalam jangka panjang.

Bidirectional LSTM (BiLSTM) merupakan pengembangan dari LSTM konvensional yang memproses data dalam dua arah secara bersamaan (forward dan backward) sehingga setiap langkah waktu memiliki informasi

konteks dari kedua arah. Dalam konteks IDS pada data tabular multifitur seperti NSL-KDD, BiLSTM mampu menangkap relasi antar fitur dari dua arah secara simultan, menghasilkan representasi yang lebih kaya dibandingkan LSTM unidirectional. Yu et al. (2023) membuktikan bahwa arsitektur bidirectional pada unit rekuren menghasilkan akurasi 93,40% pada NSL-KDD, mengungguli LSTM konvensional. Elsayed et al. (2024) juga mengkonfirmasi bahwa LSTM mencapai recall tertinggi (99,39%) dalam klasifikasi multikelas NSL-KDD, memperkuat relevansinya sebagai komponen classifier pada arsitektur hybrid.

11. Arsitektur AE-CNN-BiLSTM

Arsitektur hybrid AE-CNN-BiLSTM merupakan kombinasi sinergis dari tiga komponen deep learning yang masing-masing memiliki peran berbeda namun saling melengkapi. Autoencoder berfungsi sebagai feature compressor yang mengompresi 122 fitur hasil preprocessing menjadi representasi bottleneck yang lebih kompak, sekaligus menghasilkan anomaly score berbasis reconstruction error untuk memandu resampling adaptif. CNN kemudian mengekstraksi pola lokal dari representasi bottleneck melalui dua lapisan Conv1D, BatchNormalization, MaxPooling, dan Dropout. BiLSTM selanjutnya memproses keluaran CNN dari dua arah untuk menangkap relasi antar fitur secara komprehensif sebelum diteruskan ke lapisan Dense dengan aktivasi softmax untuk klasifikasi lima kelas.

Pendekatan hybrid ini dilandasi oleh bukti empiris bahwa kombinasi CNN dan LSTM secara konsisten mengungguli model tunggal pada dataset IDS

(Halbouni et al., 2022; Elsayed et al., 2024), sementara penggunaan komponen bidirectional dan Autoencoder menambahkan lapisan representasi yang lebih kaya untuk mendeteksi pola serangan yang kompleks, khususnya pada kelas minoritas seperti U2R dan R2L.

Dalam sistem IDS, kombinasi ini memungkinkan model untuk menangkap pola kompleks dalam data jaringan, sehingga mampu meningkatkan nilai accuracy, recall (sensitivity), dan specificity. Penelitian sebelumnya menunjukkan bahwa CNN-LSTM memiliki performa yang lebih baik dibandingkan model tunggal, baik CNN maupun LSTM, terutama dalam mendeteksi serangan pada data tidak seimbang (Halbouni et al., 2022; Yu et al., 2023).

12. Imbalanced Data pada IDS

Imbalanced data merupakan kondisi di mana distribusi jumlah data antar kelas tidak seimbang. Pada dataset IDS seperti NSL-KDD, kelas mayoritas seperti normal dan DoS memiliki jumlah data yang jauh lebih besar dibandingkan kelas minoritas seperti U2R dan R2L. Hal ini menyebabkan model cenderung bias terhadap kelas mayoritas dan mengabaikan pola pada kelas minoritas. Permasalahan ini sangat krusial karena kegagalan mendeteksi serangan (false negative) memiliki dampak yang lebih besar dibandingkan kesalahan deteksi lainnya. Oleh karena itu, diperlukan metode seperti SMOTE untuk meningkatkan representasi data minoritas agar model dapat belajar secara lebih optimal (Hussen et al., 2023; Zhu et al., 2023).

# BAB 3
# METODE PENELITIAN

## 3.1 Jenis, Sifat, dan Pendekatan Penelitian

Penelitian ini termasuk dalam kategori penelitian terapan dengan pendekatan kuantitatif. Fokus utama penelitian adalah membangun dan menguji efektivitas model Intrusion Detection System (IDS) berbasis arsitektur Autoencoder CNN-BiLSTM dengan strategi Hybrid Resampling RUS-SMOTE berbasis Anomaly Score untuk menangani ketidakseimbangan data pada dataset NSL-KDD. Sifat penelitian ini adalah eksperimental, karena melibatkan proses perancangan, implementasi, dan evaluasi model secara sistematis dalam lingkungan simulasi komputasi.

Pendekatan kuantitatif digunakan untuk mengukur performa model secara numerik dan objektif menggunakan berbagai metrik evaluasi, yaitu accuracy, precision, recall (sensitivity), specificity, F1-score, dan false positive rate (FPR). Dengan pendekatan tersebut, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan model IDS yang tidak hanya memiliki akurasi tinggi, tetapi juga mampu meningkatkan recall (sensitivity) dan specificity dalam mendeteksi serangan pada kondisi data tidak seimbang, khususnya pada kelas minoritas R2L dan U2R.

## 3.2 Metode Pengumpulan Data

Sumber data yang digunakan dalam penelitian ini bersifat sekunder, yaitu dataset NSL-KDD yang tersedia secara terbuka dan telah banyak digunakan dalam

penelitian keamanan jaringan. NSL-KDD merupakan penyempurnaan dari dataset KDD'99 yang terdiri dari rekaman lalu lintas jaringan yang telah diklasifikasikan ke dalam kategori normal dan serangan, serta mencakup 41 fitur input yang merepresentasikan karakteristik koneksi jaringan. Dataset ini memiliki lima kelas utama, yaitu normal, Denial of Service (DoS), Probe, Remote to Local (R2L), dan User to Root (U2R). Proses pengumpulan data dilakukan dengan mengunduh dataset dari platform Kaggle pada tautan https://www.kaggle.com, kemudian menyesuaikannya dengan kebutuhan penelitian.

### 3.3 Metode Analsis Data

Metode analisis data dalam penelitian ini dilakukan melalui lima tahapan utama yang saling berkesinambungan, yaitu preprocessing data, pembangkitan anomaly score dari Autoencoder, penanganan ketidakseimbangan data menggunakan Hybrid Resampling RUS-SMOTE berbasis anomaly score, pemodelan AE-CNN-BiLSTM, serta evaluasi performa model. Tahapan ini dirancang secara sistematis untuk memastikan bahwa data yang digunakan telah siap diproses oleh model serta menghasilkan evaluasi yang objektif dan komprehensif, dengan penekanan pada kemampuan model mendeteksi serangan pada kelas minoritas yang selama ini terabaikan dalam pendekatan konvensional.

Tahap pertama adalah preprocessing data, yang bertujuan menyiapkan dataset NSL-KDD agar sesuai dengan kebutuhan model deep learning. Proses ini diawali dengan mapping label serangan ke dalam lima kelas utama, yaitu Normal, DoS, Probe, R2L, dan U2R. Selanjutnya dilakukan pemisahan antara fitur input dan target, diikuti konversi fitur kategorikal (protocol_type, service, dan flag)

menggunakan metode One-Hot Encoding untuk menghindari bias ordinal antar kategori. Setelah proses encoding, jumlah fitur bertambah dari 41 menjadi 122 fitur. Fitur numerik kemudian dinormalisasi menggunakan StandardScaler agar memiliki distribusi dengan rata-rata nol dan standar deviasi satu, sehingga meningkatkan stabilitas dan kecepatan konvergensi selama pelatihan model. Setelah preprocessing selesai, dataset dibagi menjadi tiga bagian menggunakan teknik stratified split dengan proporsi 64% data latih (80.622 sampel), 16% data validasi (20.156 sampel), dan 20% data uji (25.195 sampel). Teknik stratifikasi digunakan untuk memastikan distribusi kelas tetap konsisten di ketiga subset sehingga hasil evaluasi lebih representatif. Penting untuk dicatat bahwa seluruh proses resampling hanya diterapkan pada data latih dan tidak menyentuh data validasi maupun data uji, untuk menjaga integritas evaluasi. Sebelum dimasukkan ke dalam model, data diubah ke dalam format pseudo-sequence tiga dimensi dengan ukuran (n_samples, N_FEATURES, 1) agar dapat diproses oleh lapisan Conv1D dalam arsitektur AE-CNN-BiLSTM.

Tahap kedua adalah pembangkitan anomaly score menggunakan Autoencoder. Sebelum proses resampling dilakukan, Autoencoder dilatih terlebih dahulu pada data latih untuk mempelajari distribusi pola dari seluruh kelas. Autoencoder dirancang dengan dua lapisan Conv1D encoder masing-masing dengan 128 filter dan 64 filter, kernel size 3, aktivasi ReLU yang diikuti BatchNormalization, MaxPooling1D, dan Dropout 0,2 pada setiap lapisan. Keluaran encoder kemudian dikompres melalui lapisan Bottleneck Dense berdimensi N_FEATURES dengan aktivasi linear yang berfungsi sebagai

representasi denoising dari fitur input. Setelah pelatihan, reconstruction error (RE) dihitung per sampel sebagai Mean Squared Error antara input asli dan output rekonstruksi, kemudian RE median per kelas digunakan sebagai anomaly score yang mencerminkan seberapa besar penyimpangan distribusi tiap kelas dari pola yang dipelajari model. Hasil perhitungan menunjukkan bahwa kelas Normal memiliki RE median terendah (0,000945), sementara kelas U2R memiliki RE median tertinggi (0,089361), diikuti DoS (0,007686), Probe (0,005296), dan R2L (0,004676). Nilai RE yang sangat tinggi pada kelas U2R mengkonfirmasi bahwa pola serangan U2R paling jauh menyimpang dari distribusi normal, sehingga kelas ini memerlukan penguatan resampling yang paling besar.

Tahap ketiga adalah penanganan ketidakseimbangan data menggunakan strategi Hybrid Resampling RUS-SMOTE berbasis anomaly score. Berbeda dengan pendekatan SMOTE konvensional yang menerapkan oversampling secara seragam pada semua kelas minoritas, penelitian ini menggunakan anomaly score dari Autoencoder untuk menentukan intensitas oversampling secara adaptif per kelas. Pertama, kelas kandidat SMOTE diseleksi berdasarkan imbalance ratio sehingga hanya kelas dengan rasio lebih besar dari 5× terhadap kelas terbesar yang mendapatkan perlakuan SMOTE. Berdasarkan kriteria ini, kelas Probe, R2L, dan U2R terpilih sebagai kandidat oversampling. Faktor pengali SMOTE per kelas kemudian dihitung secara otomatis menggunakan formula adaptif yang mengintegrasikan dua komponen: komponen kuantitas berbasis logaritma natural dari imbalance ratio, dan komponen kualitas anomali berbasis anomaly score yang telah dinormalisasi. Formula ini bekerja tanpa batas atas yang arbitrer, sehingga

setiap kelas mendapatkan intensitas oversampling yang sepenuhnya ditentukan oleh karakteristik intrinsiknya sendiri baik dari sisi ketidakseimbangan numerik maupun tingkat anomali polanya. Hasilnya, kelas U2R mendapatkan faktor pengali terbesar sebesar 14,35× karena memiliki anomaly score tertinggi (RE median = 0,091631), sehingga jumlahnya meningkat dari 33 menjadi 473 data. Kelas R2L mendapatkan faktor 4,23× (637 → 2.729 data), dan Probe mendapatkan faktor 1,93× (7.460 → 14.275 data). Berbeda dari pendekatan SMOTE konvensional yang menetapkan faktor seragam dengan hyperparameter tetap, pendekatan ini menjamin bahwa kelas dengan tingkat anomali paling tinggi seperti U2R yang polanya 53,9× lebih menyimpang dari kelas Normal sehingga secara otomatis mendapatkan penguatan data yang paling besar dan proporsional. Kelas R2L mendapatkan faktor 4,23× (637 → 2.693 data), dan Probe mendapatkan faktor 1,93× (7.460 → 14.379 data). Justifikasi ilmiah pemilihan strategi hybrid ini dilandasi oleh temuan Bagui & Li (2021) yang membuktikan bahwa kombinasi RU-SMOTE secara konsisten menghasilkan macro recall tertinggi pada dataset IDS yang sangat tidak seimbang, serta temuan Dangol & Eaman (2025) yang mengkonfirmasi bahwa strategi resampling yang tidak mempertimbangkan karakteristik anomali tiap kelas menghasilkan performa yang terbatas pada NSL-KDD. Setelah SMOTE diterapkan, Random Undersampling (RUS) dilakukan pada kelas mayoritas dengan RUS_FACTOR = 0,60, sehingga kelas Normal dikurangi dari 43.099 menjadi 29.393 sampel untuk mereduksi dominasinya. Kelas DoS tidak mengalami pengurangan karena jumlahnya sudah berada di batas RUS_LIMIT. Secara keseluruhan, distribusi data latih berubah dari kondisi sangat tidak seimbang

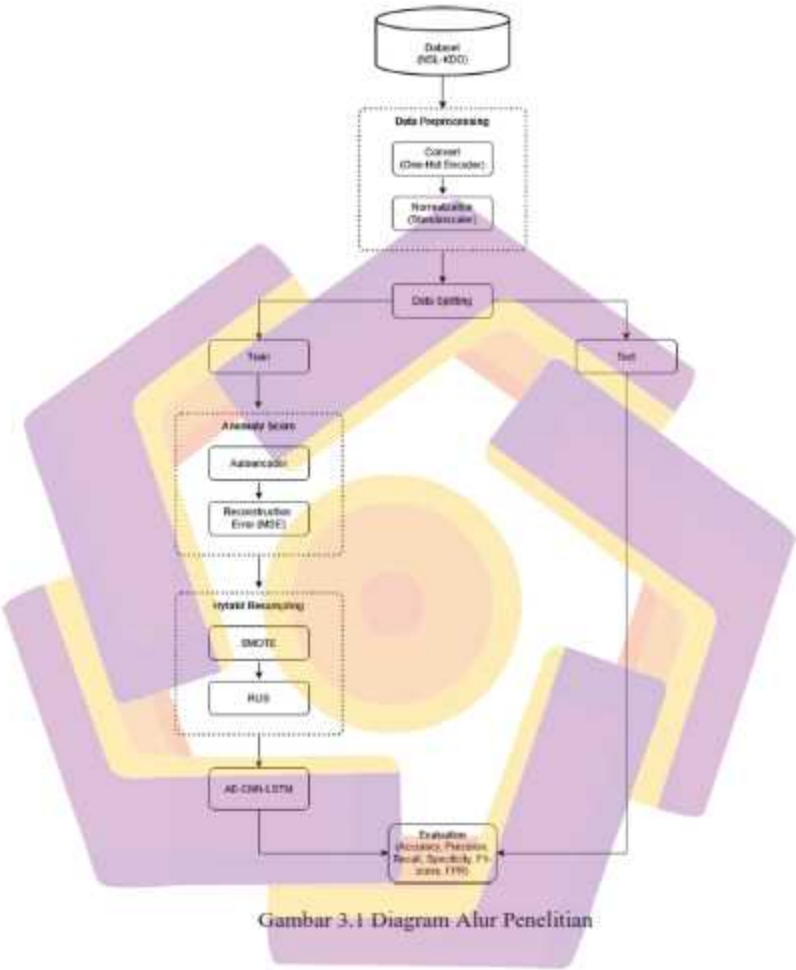
(Normal 53,46%, U2R 0,04%) menjadi distribusi yang lebih merata dan representatif (Normal 38,58%, U2R 0,43%), dengan total 76.188 sampel latih setelah resampling dari sebelumnya 80.622 sampel.

Tahap keempat adalah pemodelan AE-CNN-BiLSTM. Pemilihan arsitektur ini dilandasi oleh bukti empiris bahwa kombinasi CNN dan BiLSTM secara konsisten menghasilkan performa deteksi yang lebih tinggi dibandingkan model tunggal pada dataset NSL-KDD (Halbouni et al., 2022; Elsayed et al., 2024), sementara Autoencoder terbukti efektif sebagai feature compressor dan pembangkit anomaly score pada IDS (Yu et al., 2023). Arsitektur dibangun sebagai Arsitektur AE-CNN-BiLSTM dengan Dual Attention dibangun sebagai model end-to-end yang mengintegrasikan lima komponen utama secara berurutan dalam satu pipeline: (1) Encoder Block sebagai feature extractor sekaligus pembangkit anomaly score, (2) CNN Block untuk ekstraksi pola lokal dari representasi bottleneck, (3) FocusConv Attention untuk pembobotan fitur spasial lokal yang paling diskriminatif, (4) BiLSTM Block untuk pemodelan relasi fitur dua arah secara simultan, (5) TempoNet Attention untuk pembobotan fitur temporal global, dan diakhiri Dense Head untuk klasifikasi lima kelas. Konfigurasi arsitektur ditetapkan dengan Encoder menggunakan dua lapisan Conv1D berfilter 128 dan 64, CNN Block menggunakan dua lapisan Conv1D berfilter 32, BiLSTM dengan 128 unit (concat merge menghasilkan 256 dimensi), kedua mekanisme attention menggunakan 4 heads, serta Dense Head dengan 64 unit dan dropout 0,5. Model dioptimasi menggunakan optimizer Adam dengan learning rate 0,002 dan fungsi loss categorical crossentropy.

Tahap kelima adalah evaluasi performa model. Evaluasi dilakukan menggunakan confusion matrix pada data uji sebanyak 25.195 sampel yang tidak disentuh selama proses pelatihan maupun resampling, sehingga hasil evaluasi mencerminkan kemampuan generalisasi model yang sesungguhnya. Metrik yang dihitung mencakup accuracy, precision, recall (sensitivity), specificity, F1-score, dan False Positive Rate (FPR). Dalam penelitian ini, recall dan specificity dijadikan sebagai metrik evaluasi utama karena keduanya secara bersama-sama mencerminkan kemampuan model dalam mendeteksi serangan nyata tanpa mengorbankan kemampuan mengenali trafik normal. Accuracy tidak dijadikan metrik primer karena bersifat misleading pada kondisi data tidak seimbang dan model yang selalu memprediksi kelas Normal pun dapat mencapai akurasi lebih dari 99% pada NSL-KDD, tanpa mendeteksi satu pun serangan. Penggunaan seluruh metrik ini bertujuan untuk memberikan gambaran performa model secara menyeluruh, khususnya dalam mendeteksi serangan pada kelas minoritas R2L dan U2R yang paling kritis namun paling sering terabaikan dalam evaluasi berbasis accuracy semata (Dangol & Eaman, 2025; Halbouni et al., 2022).

### 3.4 Alur Penelitian

Untuk memberikan gambaran yang lebih sistematis mengenai alur penelitian yang dilakukan, tahapan penelitian ini divisualisasikan dalam bentuk diagram alir seperti ditunjukkan pada Gambar 3.1. Diagram tersebut menggambarkan keseluruhan proses mulai dari pengolahan data hingga evaluasi model yang diusulkan.




Gambar 3.1 Diagram Alur Penelitian

1. Dataset

Dataset yang digunakan dalam penelitian ini adalah NSL-KDD, yang merupakan penyempurnaan dari dataset KDD Cup 1999 dan telah banyak digunakan sebagai benchmark standar dalam penelitian Intrusion Detection System

(IDS). Dataset ini terdiri dari rekaman lalu lintas jaringan yang diklasifikasikan ke dalam lima kelas utama, yaitu Normal, Denial of Service (DoS), Probe, Remote to Local (R2L), dan User to Root (U2R), dengan total 125.973 sampel yang dibagi menjadi data latih (80.622 sampel), data validasi (20.156 sampel), dan data uji (25.195 sampel) menggunakan teknik stratified split dengan proporsi 64:16:20. Setiap rekaman data direpresentasikan oleh 41 fitur input yang menggambarkan karakteristik koneksi jaringan dari empat kelompok utama, yaitu fitur dasar (basic features), fitur konten (content features), fitur waktu (time features), dan fitur trafik (traffic features) (Zhu et al. 2023). Karakteristik lengkap seluruh fitur beserta tipe data dan perlakuan preprocessing-nya ditampilkan pada Tabel 3.1 berikut.

Tabel 3.1 Fitur dan Karakteristik Dataset NSL-KDD

| No | Nama Fitur | Kelompok Fitur | Tipe Data |
|---|---|---|---|
| 1 | duration | Basic | Numerik |
| 2 | protocol_type | Basic | Kategorikal |
| 3 | service | Basic | Kategorikal |
| 4 | flag | Basic | Kategorikal |
| 5 | src_bytes | Basic | Numerik |
| 6 | dst_bytes | Basic | Numerik |
| 7 | land | Basic | Numerik |
| 8 | wrong_fragment | Basic | Numerik |
| 9 | urgent | Basic | Numerik |
| 10 | hot | Content | Numerik |
| 11 | num_failed_logins | Content | Numerik |
| 12 | logged_in | Content | Numerik |
| 13 | num_compromised | Content | Numerik |
| 14 | root_shell | Content | Numerik |
| 15 | su_attempted | Content | Numerik |
| 16 | num_root | Content | Numerik |
| 17 | num_file_creations | Content | Numerik |
| 18 | num_shells | Content | Numerik |
| 19 | num_access_files | Content | Numerik |
| 20 | num_outbound_cmds | Content | Numerik |
| 21 | is_host_login | Content | Numerik |

| | | | |
|---|---|---|---|
| 22 | is_guest_login | Content | Numerik |
| 23 | count | Time | Numerik |
| 24 | srv_count | Time | Numerik |
| 25 | serror_rate | Time | Numerik |
| 26 | srv_serror_rate | Time | Numerik |
| 27 | rerror_rate | Time | Numerik |
| 28 | srv_rerror_rate | Time | Numerik |
| 29 | same_srv_rate | Time | Numerik |
| 30 | diff_srv_rate | Time | Numerik |
| 31 | srv_diff_host_rate | Time | Numerik |
| 32 | dst_host_count | Traffic | Numerik |
| 33 | dst_host_srv_count | Traffic | Numerik |
| 34 | dst_host_same_srv_rate | Traffic | Numerik |
| 35 | dst_host_diff_srv_rate | Traffic | Numerik |
| 36 | dst_host_same_src_port_rate | Traffic | Numerik |
| 37 | dst_host_srv_diff_host_rate | Traffic | Numerik |
| 38 | dst_host_serror_rate | Traffic | Numerik |
| 39 | dst_host_srv_serror_rate | Traffic | Numerik |
| 40 | dst_host_rerror_rate | Traffic | Numerik |
| 41 | dst_host_srv_rerror_rate | Traffic | Numerik |

Berdasarkan Tabel 3.1, dari 41 fitur input terdapat 38 fitur Numerik dan 3 fitur Kategorikal yaitu protocol_type, service, dan flag. Perbedaan tipe data ini menentukan perlakuan preprocessing yang berbeda untuk masing-masing kelompok, sebagaimana dijelaskan pada tahap Encoding dan Normalization berikutnya.

2. Data Preprocessing

a. Labeling

Pada tahap ini dilakukan identifikasi terhadap kolom label yang berfungsi sebagai target klasifikasi dalam penelitian. Kolom label menunjukkan apakah suatu koneksi jaringan termasuk trafik normal atau jenis serangan

tertentu. Label ini menjadi variabel target yang akan diprediksi oleh model CNN-LSTM selama proses pelatihan dan pengujian (Montaha et al. 2022).

b. Mapping Attack Classes

Dataset NSL-KDD memiliki berbagai jenis serangan yang berbeda. Untuk menyederhanakan proses klasifikasi dan mempermudah analisis hasil, setiap jenis serangan dipetakan ke dalam lima kelas utama, yaitu Normal, Denial of Service (DoS), Probe, Remote to Local (R2L), dan User to Root (U2R). Proses mapping ini dilakukan dengan mengelompokkan setiap jenis serangan ke dalam kategori serangan yang sesuai sehingga model dapat mempelajari pola serangan pada tingkat kelas yang lebih umum (Prasath et al. 2022).

c. Cleaning

Setelah proses mapping label dilakukan, tahap selanjutnya adalah menghapus data atau atribut yang tidak diperlukan dalam proses pelatihan model. Proses ini bertujuan untuk mengurangi kompleksitas dataset serta menghindari penggunaan fitur yang tidak memberikan kontribusi signifikan terhadap proses klasifikasi. Dengan menghilangkan atribut yang tidak relevan, proses pembelajaran model menjadi lebih efisien dan risiko overfitting dapat diminimalkan.

d. Encoding

Sebagaimana ditunjukkan pada Tabel 3.2, terdapat 3 fitur Kategorikal pada dataset NSL-KDD yaitu protocol_type, service, dan flag. Ketiga fitur ini

tidak dapat diproses langsung secara numeris oleh model deep learning karena tidak memiliki hubungan ordinal antar nilainya dengan penggunaan Label Encoding misalnya akan mengimplikasikan urutan palsu seperti tcp, udp dan icmp yang tidak bermakna secara jaringan. Oleh karena itu, ketiganya ditransformasi menggunakan OneHotEncoder yang mengubah setiap nilai kategori menjadi representasi vektor biner. Proses ini menghasilkan tambahan 81 kolom baru (protocol_type: 3 nilai menjadi 3 kolom; service: 70 nilai menjadi 70 kolom; flag: 11 nilai menjadi 11 kolom), sehingga total fitur bertambah dari 41 menjadi 122 fitur. Transformasi ini diterapkan secara simultan bersama StandardScaler dalam satu ColumnTransformer, sehingga pada saat proses resampling (SMOTE dan RUS) dijalankan, seluruh 122 fitur telah berstatus Numerik dan tidak ada manipulasi numeris terhadap data Kategorikal.

c. Normalization

Sebagaimana dijelaskan pada poin Encoding, transformasi OneHotEncoder dan StandardScaler diterapkan secara simultan dalam satu ColumnTransformer tidak secara berurutan. StandardScaler diterapkan khusus pada 38 fitur Numerik untuk mengubah distribusi nilai setiap fitur menjadi rata-rata nol dan standar deviasi satu. Normalisasi dilakukan karena fitur-fitur Numerik pada dataset NSL-KDD memiliki skala yang sangat bervariasi, misalnya src_bytes dapat bernilai hingga jutaan sementara land hanya bernilai 0 atau 1. Tanpa normalisasi, fitur dengan skala besar akan mendominasi proses pembelajaran dan memperlambat konvergensi model

CNN dan BiLSTM (Halbouni et al., 2022). Proses StandardScaler hanya di-fit pada data latih dan kemudian diterapkan pada data validasi dan data uji untuk mencegah data leakage dari informasi statistik data uji ke dalam proses pelatihan.

3. Data Splitting

Dataset yang telah diproses kemudian dibagi menjadi tiga bagian utama yaitu data pelatihan, data validasi, dan data pengujian. Data pelatihan digunakan untuk melatih model AE-CNN-BiLSTM dalam mempelajari pola lalu lintas jaringan, data validasi digunakan untuk memantau performa model selama proses pelatihan, sedangkan data pengujian digunakan untuk mengevaluasi kemampuan generalisasi model terhadap data yang belum pernah digunakan sebelumnya. Pada penelitian ini, dataset dibagi menggunakan rasio 80:20, di mana 80% data digunakan sebagai data pelatihan (training data) dan 20% sisanya digunakan sebagai data pengujian (testing data), kemudian sebagian dari data pelatihan dipisahkan kembali untuk digunakan sebagai data validasi (validation data) (Halbouni et al. 2022).

4. Anomaly Score (Autoencoder)

Sebelum proses resampling dilakukan, terlebih dahulu dibangun model Autoencoder yang dilatih secara unsupervised pada data pelatihan untuk mempelajari distribusi representasi normal dari seluruh kelas. Autoencoder merupakan arsitektur neural network yang terdiri dari dua komponen utama, yaitu encoder dan decoder. Encoder berfungsi mengompresi data input menjadi representasi berdimensi lebih rendah pada ruang laten (latent space),

sedangkan decoder berfungsi merekonstruksi kembali data dari representasi tersebut mendekati data aslinya (Yu et al., 2023). Secara matematis, proses encoding dapat dirumuskan sebagai:

$$z = f(W_e x - b_e) \quad (1)$$

di mana $x$ adalah vektor input, $W_e$ adalah matriks bobot encoder, $b_e$ adalah vektor bias, $f(\cdot)$ adalah fungsi aktivasi ReLU, dan $z$ adalah representasi latent. Proses decoding kemudian merekonstruksi input melalui persamaan:

$$\hat{x} = g(W_d z - b_d) \quad (2)$$

di mana $W_d$ adalah matriks bobot decoder, $b_d$ adalah vektor bias, $g(\cdot)$ adalah fungsi aktivasi linear, dan $\hat{x}$ adalah output rekonstruksi. Model dilatih dengan meminimalkan fungsi loss berupa Mean Squared Error (MSE) antara input asli dan output rekonstruksinya:

$$\mathcal{L}_{AE} = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2 \quad (3)$$

di mana n adalah jumlah sampel pelatihan, $x_i$ adalah vektor fitur input asli sampel ke-i, dan $\hat{x}_i$ adalah vektor hasil rekonstruksi dari decoder. Semakin kecil nilai $\mathcal{L}_{AE}$, semakin baik kemampuan Autoencoder dalam merekonstruksi data input, yang berarti model telah berhasil mempelajari representasi distribusi data yang kompak dan informatif pada ruang laten. Fungsi loss MSE dipilih karena memberikan penalti yang lebih besar pada deviasi yang signifikan antara input dan rekonstruksi, sehingga model lebih sensitif terhadap pola yang menyimpang dari distribusi yang telah dipelajari dengan tujuan pembangkitan anomaly score (Yu et al., 2023).

Dalam penelitian ini, Autoencoder dibangun menggunakan arsitektur fully connected dengan konfigurasi lapisan sebagai berikut: Input (122 fitur), Dense (128, ReLU), Dense (64, ReLU), Latent Dense (32, ReLU), Dense (64, ReLU), Dense (128, ReLU), Output Dense (122, linear). Model dikompilasi menggunakan optimizer Adam dengan learning rate 0,001 dan fungsi loss MSE, kemudian dilatih selama maksimum 50 epoch dengan batch size 256 menggunakan mekanisme EarlyStopping untuk mencegah overfitting. Input dan target pelatihan menggunakan data yang sama (X_tr, X_tr), sesuai dengan prinsip unsupervised reconstruction pada Autoencoder.

Setelah pelatihan selesai, reconstruction error (RE) dihitung per sampel sebagai rata-rata squared error di seluruh 122 fitur, dengan persamaan:

$$RE_i = \frac{1}{d} \sum_{j=1}^{d} \left\| x_{ij} - \hat{x}_{ij} \right\|^2 \quad (4)$$

di mana d = 122 adalah jumlah fitur, $x_{ij}$ adalah nilai fitur ke-j dari sampel ke-i, dan $\hat{x}_{ij}$ adalah nilai rekonstruksinya. RE yang tinggi pada suatu sampel mengindikasikan bahwa pola data tersebut menyimpang secara signifikan dari distribusi yang dipelajari model, sehingga berpotensi merepresentasikan anomali atau serangan (Yu et al., 2023; Hnamte et al., 2023). Untuk memperoleh anomaly score yang representatif dan robust terhadap outlier, digunakan nilai median RE per kelas bukan rata-rata karena median merupakan ukuran pemusatan yang lebih stabil pada distribusi yang tidak simetris (skewed), sebagaimana lazim ditemukan pada data jaringan yang mengandung serangan ekstrem (Bagui & Li, 2021).

5. Hybrid Resampling RUS-SMOTE Berbasis *Anomaly Score*

Penanganan ketidakseimbangan data dilakukan melalui strategi Hybrid Resampling RUS-SMOTE yang mengintegrasikan dua mekanisme secara berurutan: SMOTE adaptif berbasis anomaly score pada kelas minoritas, diikuti oleh Random Undersampling (RUS) pada kelas mayoritas. Urutan SMOTE terlebih dahulu sebelum RUS dipilih karena jika RUS diterapkan lebih dahulu, sampel minoritas yang tersisa akan semakin sedikit sehingga SMOTE menghasilkan sampel sintetis yang kurang beragam dan tidak representatif — dengan urutan SMOTE → RUS, kelas minoritas diperkuat terlebih dahulu sebelum dominasi kelas mayoritas dikurangi. Pendekatan hybrid ini dilandasi oleh temuan Bagui & Li (2021) yang membuktikan bahwa kombinasi RU-SMOTE secara konsisten menghasilkan macro recall tertinggi pada dataset IDS yang sangat tidak seimbang, serta temuan Dangol & Eaman (2025) yang menunjukkan bahwa strategi resampling dengan parameter tetap menghasilkan performa terbatas pada NSL-KDD. Proses Hybrid Resampling dilaksanakan dalam empat langkah sistematis sebagai berikut :

a. Seleksi Kelas Kandidat SMOTE

Tidak semua kelas minoritas mendapatkan perlakuan SMOTE. Kelas kandidat SMOTE diseleksi berdasarkan imbalance ratio terhadap kelas terbesar menggunakan threshold IMBALANCE_THRES = 5,0x. Hanya kelas dengan rasio ketidakseimbangan lebih dari 5x yang akan di-oversample. Threshold 5× dipilih berdasarkan temuan dalam literatur imbalanced learning untuk IDS, di mana ketidakseimbangan dengan rasio di bawah 5× umumnya masih dapat ditangani model deep learning tanpa

intervensi resampling khusus karena model masih memiliki sampel yang cukup untuk mempelajari pola kelas tersebut. Abedzadeh & Jacobs (2023) menegaskan bahwa penanganan imbalanced data pada IDS baru memberikan dampak signifikan ketika rasio ketidakseimbangan melebihi ambang tertentu, sementara Shanmugam et al. (2024) mengkonfirmasi bahwa kelas dengan imbalance ratio rendah seperti DoS tidak memerlukan oversampling karena recall-nya tetap tinggi secara natural. Hal ini dapat dilihat pada Tabel 4.2, di mana kelas DoS (1,47×) tetap menghasilkan recall yang memadai tanpa resampling, sementara kelas Probe (5,78×), R2L (67,66×), dan U2R (1.306×) mengalami penurunan recall yang drastis sehingga membuktikan bahwa threshold 5× ini selaras sekaligus tervalidasi oleh data penelitian sendiri. Secara matematis, kondisi seleksi kelas kandidat dirumuskan sebagai:

$$Kandidat\ SMOTE = \left\{ c \left| \frac{N_{maks}}{N_c} > \theta_{imbalance} \right. \right\} \quad (5)$$

di mana N_max adalah jumlah sampel kelas terbesar (Normal = 43.099), $N_c$ adalah jumlah sampel kelas $c$, dan $\theta_{imbalance}$ = 5,0. Berdasarkan perhitungan ini, kelas Probe (rasio 5,78×), R2L (67,66×), dan U2R (1.306,0×) terpilih sebagai kandidat SMOTE, sementara Normal dan DoS tidak termasuk karena rasionya di bawah threshold.

b. Normalisasi Anomaly Score

Anomaly score (median RE) dari kelas kandidat SMOTE dinormalisasi ke rentang [0, 1] menggunakan normalisasi min-max hanya di antara kelas

kandidat bukan seluruh kelas. Keputusan ini penting untuk menghindari distorsi normalisasi akibat perbedaan skala RE antara kelas yang di-SMOTE dan yang tidak. Persamaan normalisasi yang digunakan adalah:

$$score\_norm_c = \frac{RE_c - RE_{min}}{RE_{max-R\ min+\varepsilon}} \quad (6)$$

di mana $RE_c$ adalah median RE kelas c, $RE_{min}$ dan $RE_{max}$ adalah nilai minimum dan maksimum RE di antara kelas kandidat SMOTE, dan $\varepsilon = 10^{-8}$ untuk menghindari pembagian nol. Hasil normalisasi dari notebook menunjukkan: Probe (score_norm = 0,0073), R2L (score_norm = 0,0000), dan U2R (score_norm = 1,0000).

c. Perhitungan Faktor SMOTE Adaptif

Faktor pengali SMOTE per kelas dihitung menggunakan formula adaptif yang mengintegrasikan dua komponen secara langsung, sehingga setiap kelas mendapatkan intensitas *oversampling* yang sepenuhnya ditentukan oleh karakteristiknya sendiri:

$$multiplier_c = ln\left(1 + \frac{N_{max}}{N_c}\right) \times (1 + score_{norm_c}) \quad (7)$$

$$T_c = max(\lfloor N_c \times multiplier_c \rfloor, N_c) \quad (8)$$

di mana $N_{max}$ adalah jumlah sampel kelas terbesar, $N_c$ adalah jumlah sampel kelas "c" dan $T_c$ adalah jumlah target sampel kelas "c" setelah SMOTE. Formula ini terdiri dari dua komponen yang saling melengkapi: komponen kuantitas $ln\left(1 + \frac{N_{max}}{N_c}\right)$ yang mengukur seberapa tidak seimbang suatu kelas secara numerik, di mana penggunaan logaritma natural meredam efek rasio yang sangat ekstrem seperti U2R (1.306×) agar

pertumbuhan faktor bersifat proporsional (Chawla et al., 2002); dan komponen kualitas anomali $(1 + score_{norm_c})$ yang mengukur seberapa menyimpang pola kelas tersebut dari distribusi normal berdasarkan anomaly score Autoencoder, sehingga kelas dengan anomali lebih tinggi secara otomatis mendapatkan penguatan data sintetis yang lebih besar (He et al., 2008). Hasil perhitungan faktor SMOTE per kelas disajikan pada Tabel 4.4.

d. Random Undersampling (RUS) pada Kelas Mayoritas

Setelah SMOTE diterapkan, Random Undersampling dilakukan pada kelas mayoritas dengan RUS_FACTOR = 0,40. RUS diterapkan setelah SMOTE karena jika RUS diterapkan lebih dahulu, sampel minoritas yang tersisa akan semakin sedikit sehingga SMOTE menghasilkan sampel sintetis yang kurang beragam dan tidak representatif untuk memastikan kelas minoritas diperkuat terlebih dahulu sebelum dominasi kelas mayoritas dikurangi (Dangol & Eaman, 2025). Target jumlah sampel kelas mayoritas setelah RUS ditetapkan sebagai:

$$N_c^{RUS} = max\left(\lfloor N_c \times RUS\_FACTOR \rfloor, N_{smallest\_non\_smote}\right) \quad (9)$$

Penerapan RUS dengan faktor 0,40 menghasilkan pengurangan kelas Normal dari 43.099 menjadi 17.239 sampel dan kelas DoS dari 29.393 menjadi 17.239 sampel. Pemilihan RUS_FACTOR = 0,40 didasarkan pada pertimbangan bahwa reduksi yang terlalu agresif (misalnya 0,30) berpotensi menghilangkan informasi penting dari kelas mayoritas, sementara reduksi yang terlalu kecil (misalnya 0,90) tidak memberikan perbaikan distribusi

yang berarti. Nilai 0,40 menghasilkan keseimbangan yang memadai antara reduksi dominasi kelas mayoritas dan preservasi informasi.

6. AE-CNN-BiLSTM

Pada tahap ini dilakukan pembangunan dan pelatihan model AE-CNN-BiLSTM sebagai model utama dalam penelitian. Pemilihan arsitektur ini dilandasi oleh bukti empiris bahwa kombinasi CNN dan BiLSTM secara konsisten menghasilkan performa deteksi yang lebih tinggi dibandingkan model tunggal pada dataset NSL-KDD (Halbouni et al., 2022; Elsayed et al., 2024), sementara Autoencoder terbukti efektif sebagai feature compressor sekaligus pembangkit anomaly score pada IDS (Yu et al., 2023). Sebelum data dimasukkan ke dalam model, data hasil resampling terlebih dahulu diubah dari format tabular menjadi pseudo-sequence tiga dimensi dengan ukuran (n_samples, 122, 1) agar dapat diproses oleh lapisan Conv1D. Label target juga dikonversi ke dalam bentuk one-hot encoding untuk mendukung klasifikasi multikelas menggunakan fungsi aktivasi softmax.

Arsitektur AE-CNN-BiLSTM dengan Dual Attention dibangun sebagai model end-to-end yang mengintegrasikan lima komponen utama secara berurutan dalam satu pipeline: (1) Encoder Block untuk mengekstraksi representasi fitur tingkat tinggi sekaligus menghasilkan anomaly score berbasis reconstruction error, (2) CNN Block untuk mengekstraksi pola lokal dari representasi bottleneck, (3) FocusConv Attention (Multi-Head Attention + residual + LayerNorm) untuk membobot fitur lokal spasial yang paling diskriminatif dari keluaran CNN, (4) BiLSTM Block untuk memodelkan relasi antar fitur dari

dua arah secara simultan, (5) TempoNet Attention (Multi-Head Attention + residual + LayerNorm + GlobalAveragePooling) untuk membobot posisi temporal paling informatif dari keluaran BiLSTM, dan diakhiri Dense Head untuk klasifikasi lima kelas. Konfigurasi arsitektur ditetapkan dengan Encoder menggunakan dua lapisan Conv1D berfilter 128 dan 64, CNN Block menggunakan dua lapisan Conv1D berfilter 32, kedua mekanisme attention menggunakan 4 heads, BiLSTM dengan 128 unit (concat merge menghasilkan 256 dimensi), serta Dense Head dengan 64 unit dan dropout 0,5. Model dioptimasi menggunakan optimizer Adam dengan learning rate 0,002 dan fungsi loss categorical crossentropy.

a. Encoder Block

Encoder Block menerima input berbentuk (N_FEATURES, 1) = (122, 1) dan berfungsi untuk mengekstraksi representasi fitur tingkat tinggi sekaligus meredam noise dari data input. Proses dimulai pada lapisan enc_conv1 berupa Conv1D dengan 128 filter, kernel size 3, padding same, dan aktivasi ReLU, yang menghasilkan feature map berdimensi (122, 128). Keluaran Conv1D melewati lapisan BatchNormalization (enc_bn1) untuk menstabilkan distribusi aktivasi dan mempercepat konvergensi, kemudian MaxPooling1D dengan pool size 2 (enc_pool1) yang mereduksi dimensi temporal menjadi (61, 128), dan Dropout 0,2 (enc_drop1) sebagai regulasi untuk mencegah overfitting. Lapisan ini kemudian dilanjutkan ke enc_conv2 berupa Conv1D dengan 64 filter, kernel size 3, padding same, aktivasi ReLU, menghasilkan feature map berdimensi (61, 64), yang

kembali diikuti BatchNormalization (enc_bn2), MaxPooling1D (enc_pool2) yang mereduksi dimensi menjadi (30, 64), dan Dropout 0,2 (enc_drop2).

Keluaran Encoder Block kemudian di-flatten menjadi vektor satu dimensi berdimensi 1.920 melalui lapisan bottleneck_flatten, lalu diproyeksikan ke lapisan Bottleneck Dense (bottleneck) berdimensi N_FEATURES = 122 dengan aktivasi linear. Lapisan Bottleneck ini berfungsi sebagai mekanisme denoising, berbeda dengan Autoencoder kompresi konvensional yang menggunakan dimensi latent lebih kecil dari input, bottleneck di sini memiliki dimensi yang sama dengan input (122) namun dipaksa melewati jalur kompresi-ekspansi sehingga menghasilkan representasi yang telah didenoisifikasi dan dibersihkan dari informasi yang tidak relevan. Representasi bottleneck ini kemudian di-reshape kembali ke format (122, 1) melalui lapisan bottleneck_reshape sebagai input untuk CNN Block berikutnya.

b. CNN Block

CNN Block menerima representasi bottleneck berdimensi (122, 1) dan berfungsi mengekstraksi pola lokal dari representasi yang telah didenoisifikasi. Proses ekstraksi fitur dilakukan menggunakan dua lapisan Conv1D dengan arsitektur yang identik. Lapisan cnn_conv1 menggunakan 32 filter, kernel size 3, padding same, aktivasi ReLU, menghasilkan feature map berdimensi (122, 32). Operasi konvolusi satu dimensi ini secara matematis dapat dirumuskan sebagai berikut :

$$Z_j = ReLU\left(\sum_k W_{jk} * X_k + b_j\right) \quad (10)$$

di mana $W_{jk}$ adalah bobot filter-j, $X_k$ adalah feature map input ke-k, " $*$ " menandakan operasi konvolusi, dan $b_j$ adalah bias. Keluaran kemudian melewati BatchNormalization (`cnn_bn1`) yang menormalisasi aktivasi per batch menggunakan persamaan $\hat{X} = X - \mu_B / \sqrt{\delta_B^2 + \epsilon}$ untuk menstabilkan pelatihan (Halbouni et al., 2022), diikuti MaxPooling1D pool size 2 (`cnn_pool1`) yang mereduksi dimensi menjadi (61, 32), dan Dropout 0,3 (`cnn_drop1`). Lapisan `cnn_conv2` mengulangi proses serupa dengan 32 filter menghasilkan feature map (61, 32), kembali diikuti BatchNormalization (`cnn_bn2`), MaxPooling1D (`cnn_pool2`) yang mereduksi dimensi menjadi (30, 32), dan Dropout 0,3 (`cnn_drop2`). Penggunaan dua lapisan konvolusi bertujuan untuk mengekstraksi hierarki fitur yang semakin abstrak, lapisan pertama menangkap pola lokal sederhana, sedangkan lapisan kedua menangkap representasi fitur yang lebih kompleks.

c. BiLSTM Block

Keluaran CNN Block berdimensi (30, 32) kemudian diproses oleh lapisan Bidirectional LSTM (`bilstm`) dengan 128 unit dan `merge_mode="concat"`. BiLSTM memproses urutan fitur dari dua arah secara bersamaan, arah forward $\vec{H}$ dari langkah waktu pertama hingga terakhir, dan arah backward $\overleftarrow{H}$ dari langkah waktu terakhir menuju pertama, kemudian menggabungkan kedua representasi tersebut melalui operasi konkatenasi:

$$V_t = [\overrightarrow{H_t} : \overleftarrow{H_t}] \quad (11)$$

di mana $\overrightarrow{H_t}$ adalah hidden state arah forward dan $\overleftarrow{H_t}$ adalah "hidden state" arah backward pada langkah waktu 't' (Yu et al., 2023). Dengan `merge_mode="concat"`, output BiLSTM memiliki dimensi 256 (128 unit x 2 arah), yang merepresentasikan konteks fitur dari kedua arah secara bersamaan. Parameter `return_sequences=False` memastikan hanya hidden state pada langkah waktu terakhir yang diteruskan ke lapisan berikutnya, menghasilkan vektor fitur tunggal berdimensi 256 sebagai representasi komprehensif dari seluruh urutan fitur.

d. FocusConv Attention

Keluaran CNN Block berdimensi (30, 32) diproses oleh FocusConv Attention sebelum masuk ke BiLSTM. Mekanisme ini menggunakan Multi-Head Attention (focusconv_attention) dengan num_heads = 4 dan key_dim = 8 untuk membobot fitur lokal spasial secara adaptif menonjolkan fitur yang paling diskriminatif dan meredam fitur yang tidak relevan dari hasil ekstraksi CNN. Secara matematis, attention dihitung sebagai:

$$F_{att} = softmax\left(\frac{QK^T}{\sqrt{d_k N_c}}\right) \times V \quad (12)$$

Output attention digabungkan dengan input asli melalui koneksi residual (focusconv_residual) kemudian dinormalisasi (focusconv_norm), menghasilkan representasi berdimensi (30, 32). Koneksi residual memastikan informasi asli dari CNN tetap terjaga sementara attention hanya menambahkan pembobotan adaptif di atasnya (Zhang et al., 2025).

e. TempoNet Attention

Keluaran BiLSTM berdimensi (30, 256) diproses oleh TempoNet Attention yang berfungsi membobot posisi temporal paling informatif dari representasi BiLSTM secara global. Mekanisme ini menggunakan Multi-Head Attention (temponet_attention) dengan num_heads = 4 dan key_dim = 16 key_dim lebih besar dari FocusConv karena representasi BiLSTM (256 dimensi) jauh lebih kaya dibandingkan output CNN. Output attention digabungkan melalui koneksi residual (temponet_residual), dinormalisasi (temponet_norm), lalu dirangkum oleh GlobalAveragePooling1D (global_avg_pool) menjadi vektor berdimensi (256,) sebagai input Dense Head. Berbeda dengan FocusConv yang beroperasi pada dimensi spasial (fitur mana yang penting), TempoNet beroperasi pada dimensi temporal (posisi sekuensial mana yang paling informatif) dan keduanya bersifat komplementer sehingga menghasilkan efek sinergis yang terbukti meningkatkan Recall U2R sebesar 10,00% ketika digunakan bersamaan (Tabel 4.9).

f. Dense Head

Vektor fitur BiLSTM berdimensi 256 kemudian diteruskan ke lapisan dense_head berupa lapisan Dense dengan 64 unit dan aktivasi ReLU, yang berfungsi memetakan representasi BiLSTM ke ruang fitur yang lebih kompak. Dropout 0,5 (dense_drop) diterapkan setelah lapisan ini sebagai regulasi yang lebih agresif untuk mencegah overfitting pada tahap akhir sebelum klasifikasi. Lapisan output berupa Dense dengan 5 unit dan

aktivasi softmax menghasilkan distribusi probabilitas pada setiap kelas, di mana kelas dengan probabilitas tertinggi dipilih sebagai prediksi akhir:

$$P(y = k|x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}, k = 1,2,\dots,K \quad (13)$$

di mana $z_k$ adalah 'logit' kelas ke-k dan K = 5 adalah jumlah kelas.

7. Evaluasi

Tahap evaluasi dan pengujian dilakukan untuk mengukur kinerja model CNN-LSTM dalam mendeteksi intrusi jaringan pada dataset NSL-KDD. Evaluasi dilakukan menggunakan data pengujian (testing data) yang tidak digunakan selama proses pelatihan model. Pendekatan ini bertujuan untuk memastikan bahwa hasil pengujian dapat menggambarkan kemampuan generalisasi model terhadap data baru.

Pengukuran performa model dilakukan menggunakan beberapa metrik evaluasi yang umum digunakan dalam penelitian Intrusion Detection System (IDS), yaitu accuracy, precision, recall (sensitivity), specificity, F1-score, dan false positive rate (FPR) (Wang et al. 2022). Metrik-metrik tersebut dihitung berdasarkan nilai confusion matrix yang terdiri dari True Positive (TP), True Negative (TN), False Positive (FP), dan False Negative (FN).

b. Accuracy

Akurasi model adalah ukuran seberapa akurat perkiraannya secara keseluruhan (positif dan negatif yang diidentifikasi secara akurat).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \quad (14)$$

c. Precision

Akurasi model adalah ukuran seberapa akurat perkiraannya secara keseluruhan (positif dan negatif yang diidentifikasi secara akurat).

$$Precision = \frac{TP}{TP+} \quad (15)$$

d. Recall (Sensitivity)

Recall mengevaluasi kapasitas model untuk secara akurat membedakan kejadian positif dari semua kasus positif yang sebenarnya dalam kumpulan data.

$$Recall = \frac{TP}{TP + FN} \quad (16)$$

e. Specificity

Specificity mengukur kemampuan model dalam mengenali data normal secara benar dari seluruh data normal yang tersedia.

$$Specificity = \frac{TN}{TN + FP} \quad (17)$$

f. F1-Score

F1-score merupakan ukuran yang menggabungkan precision dan recall untuk memberikan gambaran keseimbangan performa model.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (18)$$

g. FPR (False Positive Rate)

FPR menilai proporsi data normal yang salah diklasifikasikan sebagai serangan. Nilai FPR rendah sangat diharapkan agar IDS tidak menghasilkan terlalu banyak alarm palsu.

$$FPR = \frac{FP}{FP + TN} \quad (18)$$

# BAB 4
# HASIL PENELITIAN DAN PEMBAHASAN

## 4.1 Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini adalah NSL-KDD, yang merupakan pengembangan dari dataset KDD Cup 1999. Dataset ini dirancang untuk mengatasi kelemahan utama pada KDD'99, yaitu adanya redundansi data yang tinggi serta distribusi data yang tidak seimbang. NSL-KDD telah banyak digunakan sebagai dataset benchmark dalam penelitian Intrusion Detection System (IDS) karena mampu memberikan evaluasi yang lebih objektif terhadap performa model.

Secara umum, dataset NSL-KDD terdiri dari data lalu lintas jaringan yang direpresentasikan dalam bentuk fitur-fitur numerik dan kategorikal. Dataset ini memiliki 41 fitur input yang mencerminkan karakteristik koneksi jaringan, seperti durasi koneksi, jumlah byte yang dikirim, serta informasi protokol yang digunakan. Selain itu, terdapat satu label target yang menunjukkan apakah suatu koneksi termasuk kategori normal atau serangan.

### 4.1.1 Struktur dan Pembagian Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset NSL-KDD yang terdiri dari total 125.973 data yang merepresentasikan lalu lintas jaringan dengan berbagai jenis serangan. Data pada dataset ini diklasifikasikan ke dalam lima kelas utama, yaitu Normal, Denial of Service (DoS), Probe, Remote to Local (R2L), dan User to Root (U2R). Dataset dibagi menjadi tiga bagian utama, yaitu data latih (training data), data validasi (validation data), dan data uji (testing data). Pembagian

ini dilakukan untuk memastikan bahwa model dapat dilatih, dievaluasi selama proses pelatihan, serta diuji kemampuan generalisasinya terhadap data yang belum pernah dilihat sebelumnya. Pembagian dataset dilakukan menggunakan teknik stratified split dengan proporsi 64:16:20 untuk memastikan distribusi kelas tetap konsisten di seluruh subset. Distribusi data latih, validasi, dan uji ditampilkan pada Tabel 4.1 berikut.

Tabel 4.1 Pembagian Dataset

| Kelas Serangan | Jumlah Data | Proporsi (%) | Data Latih | Proporsi (%) | Data Vaidasi | Proporsi (%) | Data Uji | Proporsi (%) |
|---|---|---|---|---|---|---|---|---|
| Normal | 67343 | 53.46% | 43099 | 53.46% | 10775 | 53.46% | 13469 | 53.46% |
| DoS | 45927 | 36.46% | 29393 | 36.46% | 7348 | 36.46% | 9186 | 36.46% |
| Probe | 11656 | 9.25% | 7460 | 9.25% | 1865 | 9.25% | 2331 | 9.25% |
| R2L | 995 | 0.79% | 637 | 0.79% | 159 | 0.79% | 199 | 0.79% |
| U2R | 52 | 0.04% | 33 | 0.04% | 9 | 0.04% | 10 | 0.04% |
| **Total** | **125973** | **100.00%** | **80622** | **100.00%** | **20156** | **100.00%** | **25195** | **100.00%** |

Berdasarkan Tabel 4.1, pembagian dataset menggunakan teknik stratified split dengan proporsi 64:16:20 memastikan bahwa proporsi kelas pada data latih, validasi dan uji tetap identik yang mencerminkan distribusi kelas faktual asli dataset NSL-KDD. Dengan demikian, ketiga subset memiliki perbandingan jumlah kelas yang sama: Normal 53,46%, DoS 36,46%, Probe 9,25%, R2L 0,79%, dan U2R 0,04%. Data latih sebanyak 80.622 sampel digunakan untuk melatih seluruh model dan sebagai sumber data yang akan diproses melalui tahap resampling. Data validasi sebanyak 20.156 sampel digunakan untuk memantau performa model selama proses pelatihan. Data uji sebanyak 25.195 sampel digunakan sebagai basis evaluasi akhir dan tidak disentuh selama proses pelatihan maupun resampling.

sehingga hasil evaluasi mencerminkan kemampuan generalisasi model yang sesungguhnya.

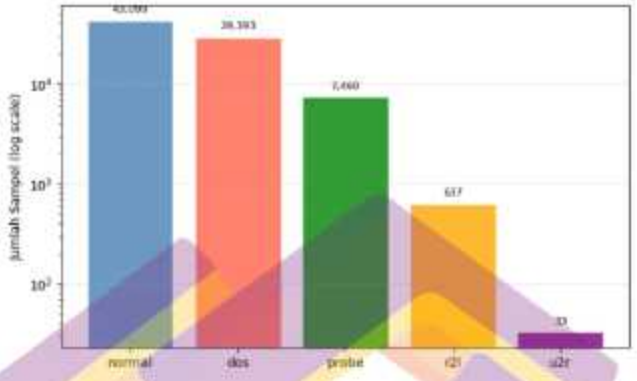
### 4.1.2 Distribusi Label dan Ketidakseimbangan Data

Distribusi data pada dataset NSL-KDD menunjukkan adanya ketidakseimbangan jumlah data antar kelas (imbalanced data) yang sangat signifikan. Kelas mayoritas seperti Normal dan DoS memiliki jumlah data yang jauh lebih besar dibandingkan kelas minoritas seperti R2L dan U2R. Distribusi kelas pada data pelatihan dapat dilihat secara rinci pada Tabel 4.2.

Tabel 4.2 Distribusi Kelas Data Pelatihan NSL-KDD

| Kelas Serangan | Jumlah Sampel | Proporsi (%) | Imbalance Ratio |
|---|---|---|---|
| Normal | 43.099 | 53.46% | 1 × |
| DoS | 29.392 | 36.46% | 1.47 × |
| Probe | 7.460 | 9.25% | 5.78 × |
| R2L | 636 | 0.79% | 67.66 × |
| U2R | 36 | 0.04% | 1306 × |
| **Total** | **80.622** | **100%** | |

Berdasarkan Tabel 4.2, terlihat bahwa kelas U2R memiliki imbalance ratio yang sangat ekstrem sebesar 1.306× terhadap kelas Normal, sementara R2L memiliki rasio 67,66×. Distribusi kelas sebelum resampling divisualisasikan pada Gambar 4.1 untuk memberikan gambaran yang lebih intuitif mengenai tingkat ketimpangan tersebut.




Gambar 4.1 Distribusi Kelas Data Pelatihan Sebelum Resampling

Gambar 4.1 menunjukkan bahwa kelas Normal dan DoS mendominasi dataset secara visual, sementara kelas R2L dan U2R hampir tidak terlihat karena jumlahnya yang sangat kecil. Kondisi ini secara langsung berdampak pada kemampuan model tanpa penanganan khusus, model akan cenderung bias terhadap kelas mayoritas dan mengabaikan kelas minoritas. Hal ini dikonfirmasi oleh hasil eksperimen pada Skenario 1 (tanpa resampling) yang menunjukkan bahwa meskipun arsitektur AE-CNN-BiLSTM sudah kuat secara teknis dengan accuracy 0,9952, nilai recall (sensitivity) U2R hanya mencapai 0,6000 dan recall R2L hanya 0,8090 membuktikan bahwa accuracy yang tinggi tidak menjamin kemampuan deteksi kelas minoritas yang memadai dalam konteks IDS.

## 4.2 Hasil Preprocessing Data

Tahap preprocessing bertujuan untuk menyiapkan dataset NSL-KDD agar sesuai dengan kebutuhan arsitektur deep learning. Setiap jenis serangan dalam

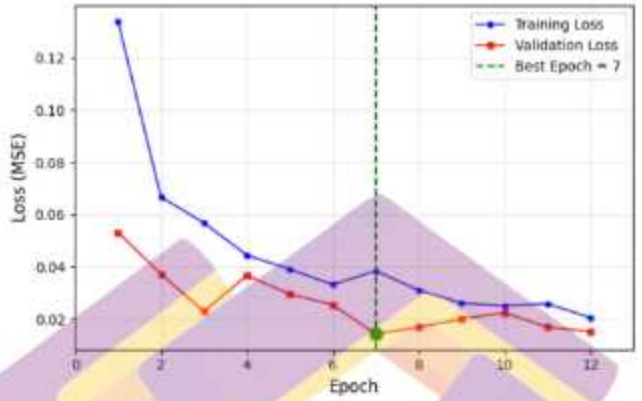
dataset dipetakan ke dalam lima kelas utama yang telah ditetapkan. Fitur kategorikal (protocol_type, service, dan flag) dikonversi menggunakan metode One-Hot Encoding, sehingga jumlah fitur bertambah dari 41 menjadi 122 fitur numerik. Fitur numerik kemudian dinormalisasi menggunakan StandardScaler sehingga memiliki distribusi dengan rata-rata nol dan standar deviasi satu, yang meningkatkan stabilitas dan kecepatan konvergensi selama pelatihan model. Sebelum dimasukkan ke dalam model, data diubah ke dalam format pseudo-sequence tiga dimensi dengan ukuran (n_samples, 122, 1) agar dapat diproses oleh lapisan Conv1D dalam arsitektur AE-CNN-BiLSTM.

## 4.3 Penanganan Imbalanced Data

Penanganan ketidakseimbangan data dilakukan melalui dua tahap yang saling berkaitan, yaitu pembangkitan anomaly score menggunakan Autoencoder dan penerapan Hybrid Resampling RUS-SMOTE berbasis anomaly score tersebut. Kedua tahap ini dirancang secara terintegrasi sehingga intensitas resampling tiap kelas ditentukan secara adaptif berdasarkan karakteristik anomali intrinsik yang diperoleh dari data itu sendiri.

### 4.3.1 Analisis Anomaly Score dari Autoencoder

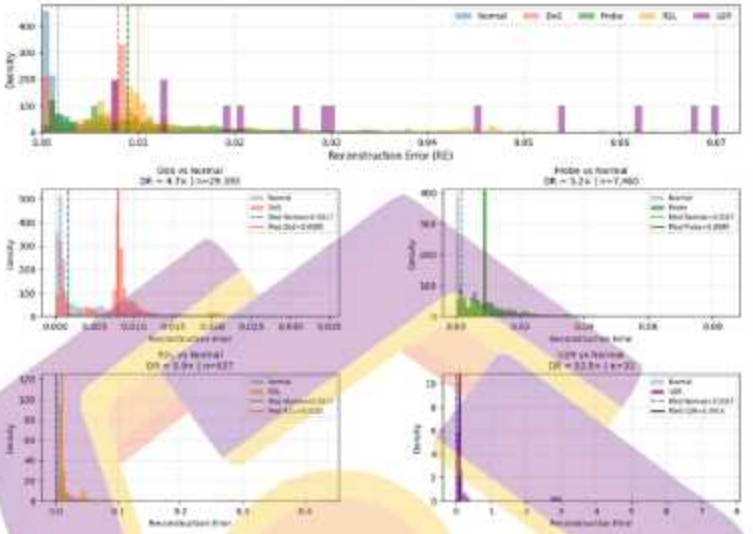
Sebelum proses *resampling* dilakukan, Autoencoder dilatih pada data pelatihan untuk mempelajari distribusi representasi dari seluruh kelas. Kurva *loss* pelatihan Autoencoder ditampilkan pada Gambar 4.2.




Gambar 4.2 Kurva Loss Pelatihan Autoencoder

Gambar 4.2 menunjukkan bahwa training loss dan validation loss Autoencoder mengalami penurunan yang stabil dan konsisten selama proses pelatihan tanpa indikasi overfitting, yang mengkonfirmasi bahwa Autoencoder berhasil mempelajari distribusi representasi data secara efektif.

Setelah pelatihan, reconstruction error (RE) dihitung per sampel sebagai rata-rata squared error antara input asli dan output rekonstruksinya. Distribusi RE per kelas divisualisasikan pada Gambar 4.3 untuk menunjukkan perbedaan tingkat anomali antar kelas secara visual.



Gambar 4.3 Histogram Distribusi Reconstruction Error per Kelas

Gambar 4.3 menunjukkan bahwa kelas Normal memiliki distribusi RE yang sangat terkonsentrasi mendekati nol, sementara kelas U2R memiliki distribusi RE yang tersebar jauh ke kanan dengan nilai yang jauh lebih tinggi. Hal ini mengkonfirmasi bahwa pola serangan U2R paling menyimpang dari distribusi yang dipelajari Autoencoder. Nilai median RE per kelas kemudian digunakan sebagai anomaly score yang ditampilkan pada Tabel 4.3.

Tabel 4.3 Nilai Median Reconstruction Error (Anomaly Score) per Kelas

| Kelas | Median RE (Anomaly Score) | Score_norm | Keterangan |
|---|---|---|---|
| Normal | 0.001701 | — | Bukan kandidat SMOTE |
| DoS | 0.007982 | — | Bukan kandidat SMOTE |
| Probe | 0.008918 | 0.0000 | Kandidat SMOTE (anomali tinggi) |
| R2L | 0.010017 | 0.0133 | Kandidat SMOTE (anomali tinggi) |
| U2R | 0.091631 | 1.0000 | Kandidat SMOTE (anomali SANGAT tinggi) |

Berdasarkan Tabel 4.3, terlihat bahwa kelas U2R memiliki nilai *anomaly score* tertinggi dengan median RE sebesar **0,091631** dimana nilai tersebut **53,9**× lebih tinggi dari kelas Normal (0,001701). Perbedaan yang sangat signifikan ini menjadi justifikasi ilmiah yang kuat bahwa U2R memerlukan penguatan *resampling* terbesar, karena pola serangannya paling menyimpang dari distribusi data yang dipelajari Autoencoder. Perlu dicatat pula bahwa pada penelitian ini, urutan *anomaly score* di antara kelas kandidat SMOTE menunjukkan bahwa R2L (RE = 0,010017) memiliki *anomaly score* sedikit lebih tinggi dari Probe (RE = 0,008918), sehingga R2L mendapatkan faktor pengali SMOTE yang lebih besar (4,29×) dibandingkan Probe (1,91×) meskipun secara imbalance ratio R2L jauh lebih tidak seimbang. Dengan demikian, *anomaly score* bukan sekadar parameter arbitrer, melainkan cerminan langsung dari karakteristik intrinsik tiap kelas yang diperoleh secara *unsupervised* dari data itu sendiri, dan mampu memberikan informasi yang lebih kaya dibandingkan sekadar rasio ketidakseimbangan dalam menentukan intensitas *resampling* per kelas.

### 4.3.2 Hasil Hybrid Resampling RUS-SMOTE Berbasis Anomaly Score

Berdasarkan *anomaly score* yang telah dinormalisasi, faktor pengali SMOTE dihitung secara adaptif untuk setiap kelas kandidat. Hasil perhitungan faktor SMOTE per kelas ditampilkan pada Tabel 4.4.

Tabel 4.4 Faktor SMOTE Adaptif Berbasis Anomaly Score per Kelas

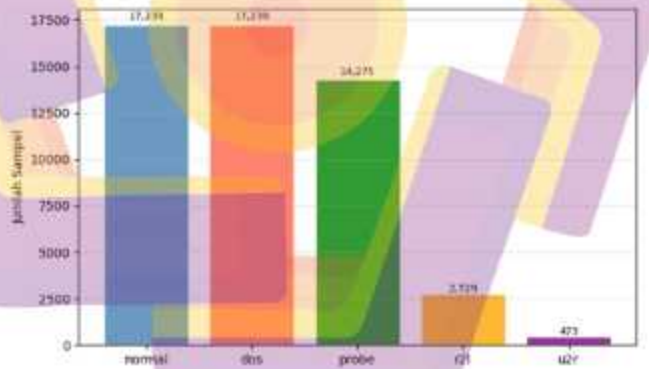
| Kelas | N Awal | Score_norm | Formula | N Target | Tambahan Sampel |
|---|---|---|---|---|---|
| Probe | 7460 | 0.0000 | 1.93 × | 14275 | 6815 |
| R2L | 637 | 0.0133 | 4.23 × | 2729 | 2092 |
| U2R | 33 | 1.0000 | 14.35 × | 473 | 440 |

Berdasarkan Tabel 4.4, kelas U2R mendapatkan faktor pengali terbesar sebesar 14,35× hampir 7,4× lebih besar dari kelas Probe karena *anomaly score*-nya yang paling tinggi. Hal ini berbeda fundamental dari pendekatan SMOTE konvensional yang memberikan faktor seragam pada semua kelas kandidat tanpa

mempertimbangkan karakteristik anomali masing-masing kelas. Setelah penerapan SMOTE, *Random Undersampling* (RUS) dilakukan pada kelas mayoritas dengan RUS_FACTOR = 0,40 untuk mereduksi dominasi Normal dan DoS. Distribusi data pelatihan sebelum dan sesudah seluruh proses Hybrid Resampling RUS-SMOTE ditampilkan pada Tabel 4.5 dan Gambar 4.4.

Tabel 4.5 Data Pelatihan Sebelum dan Sesudah Resampling RUS-SMOTE

| Kelas | Sebelum | % | Sesudah | % | Perubahan |
|---|---|---|---|---|---|
| Normal | 43099 | 53.46% | 17239.00 × | 33.18% | −25,860 (RUS) |
| DoS | 29393 | 36.46% | 17239.00 × | 33.18% | −12,154 (RUS) |
| Probe | 7460 | 9.25% | 14275.00 × | 27.48% | +6,815 (SMOTE) |
| R2L | 637 | 0.79% | 2729.00 × | 5.25% | +2,092 (SMOTE) |
| U2R | 33 | 0.04% | 473.00 × | 0.91% | +440 (SMOTE) |
| **Total** | **80622** | **100%** | **51955.00 ×** | **100%** | |




Gambar 4.4 Data Pelatihan Sesudah Hybrid Resampling RUS-SMOTE

Gambar 4.4 dan Tabel 4.5 menunjukkan perbandingan distribusi kelas pada data pelatihan sebelum dan sesudah penerapan Hybrid Resampling RUS-SMOTE berbasis anomaly score. Setelah resampling, final imbalance ratio antara kelas terbesar dan terkecil berhasil direduksi dari 1.306× menjadi 36,4× (penurunan

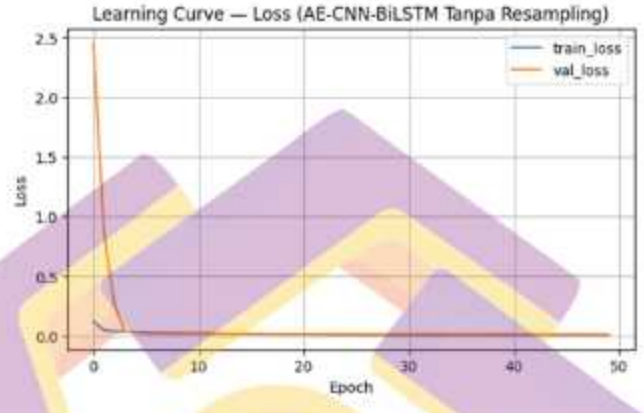
97,2%). Angka-angka distribusi akhir bukan merupakan target yang ditetapkan secara manual, melainkan hasil kalkulasi otomatis dari dua parameter: RUS_FACTOR = 0,40 dan formula multiplier berbasis anomaly score. Kelas Normal dan DoS masing-masing diturunkan ke 17.239 sampel melalui RUS_LIMIT = 43.099 × 0,40 sehingga dominasinya berkurang secara signifikan, sementara kelas Probe (14.275), R2L (2.729), dan U2R (473) ditentukan secara proporsional terhadap kombinasi imbalance ratio dan anomaly score masing-masing sehingga kelas U2R mengalami peningkatan paling drastis dari 33 menjadi 473 sampel (+1.333%) karena memiliki imbalance ratio ekstrem (1.306×) sekaligus score_norm tertinggi (1,0000) yang keduanya bekerja sinergis dalam formula multiplier. Distribusi akhir yang masih tidak seimbang ini disengaja dengan tujuan bukan menciptakan distribusi seragam yang tidak realistis, melainkan mereduksi ketimpangan ke tingkat yang memungkinkan model mempelajari pola kelas minoritas secara efektif sambil tetap mencerminkan proporsi realistis trafik jaringan (Bagui & Li, 2021; Abedzadeh & Jacobs, 2023). Perubahan distribusi yang signifikan ini terlihat jelas pada Gambar 4.4, berbeda dari penelitian terdahulu yang hanya menerapkan SMOTE konvensional sehingga grafik distribusi sebelum dan sesudah tidak tampak berbeda secara visual yang berarti.

## 4.4 Analisis Training Model

### 4.4.1 Analisis Training Model Tanpa Resampling

Pada tahap ini dilakukan analisis terhadap proses pelatihan model AE-CNN-BiLSTM tanpa *resampling* sebagai kondisi *baseline* untuk mengukur kemampuan arsitektur semata tanpa intervensi pada distribusi data. Analisis dilakukan

berdasarkan kurva *learning curve* yang mencakup *accuracy* dan *loss* pada data pelatihan (*training*) dan validasi (*validation*).

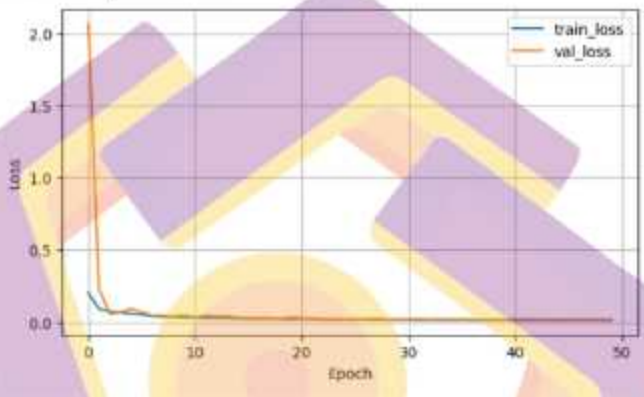



Gambar 4.5 Kurva Loss Model AE-CNN-BiLSTM Tanpa Resampling

Gambar 4.5 menunjukkan kurva *training loss* dan *validation loss* selama proses pelatihan model AE-CNN-BiLSTM tanpa *resampling*. Terlihat bahwa nilai *training loss* mengalami penurunan yang tajam pada *epoch* awal dan secara bertahap menstabilkan diri hingga akhir pelatihan. *Validation loss* juga menunjukkan tren penurunan yang stabil dengan *gap* yang relatif kecil terhadap *training loss*, yang mengindikasikan bahwa model mampu melakukan pembelajaran secara efektif tanpa indikasi *overfitting*. Meskipun kurva pelatihan menunjukkan konvergensi yang baik, kondisi *imbalanced data* menyebabkan model cenderung memprioritaskan kelas mayoritas selama pelatihan, yang tercermin pada nilai *recall* yang rendah pada kelas U2R dan R2L sebagaimana akan dibahas pada sub-bab evaluasi.

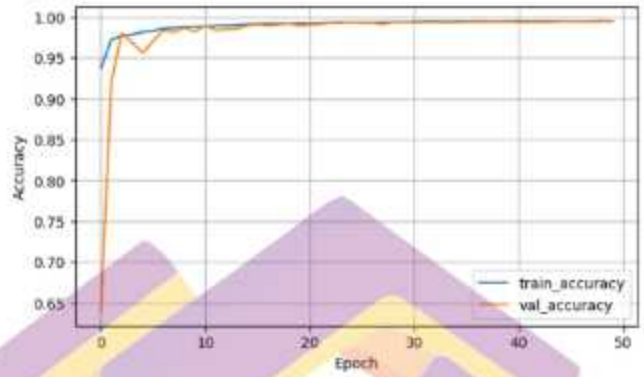
### 4.4.2 Analisis Training Model dengan Hybrid Resampling Berbasis Anomaly Score

Pada tahap ini dilakukan analisis terhadap proses pelatihan model AE-CNN-BiLSTM dengan penerapan Hybrid Resampling berbasis *anomaly score* sebagai model final dalam penelitian ini.




Gambar 4.6 Kurva Loss Model AE-CNN-BiLSTM + Hybrid Resampling

Gambar 4.6 menunjukkan kurva *training loss* dan *validation loss* selama proses pelatihan model dengan Hybrid RUS-SMOTE. Terlihat bahwa *training loss* mengalami penurunan yang tajam pada *epoch* awal dan secara bertahap menstabilkan diri menuju nilai yang mendekati nol pada akhir pelatihan. *Validation loss* juga menunjukkan tren penurunan yang stabil dengan *gap* yang relatif kecil, yang mengindikasikan bahwa model mampu melakukan pembelajaran secara efektif tanpa indikasi *overfitting* yang signifikan meskipun sebagian data pelatihan merupakan data sintetis hasil SMOTE.




Gambar 4.7 Kurva Accuracy Model AE-CNN-BiLSTM+Hybrid Resampling

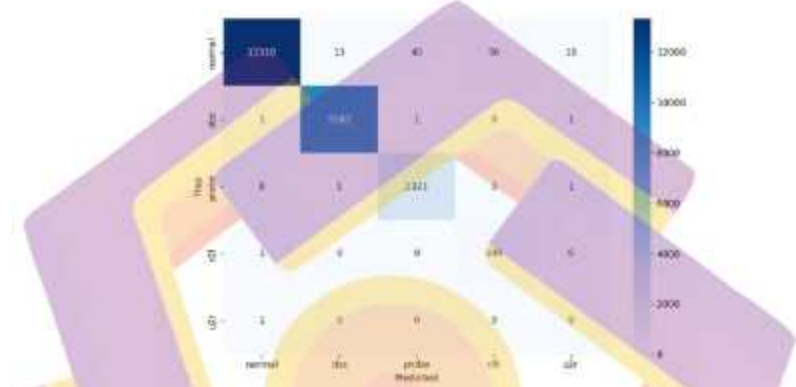
Gambar 4.7 menunjukkan kurva *training accuracy* dan *validation accuracy* yang keduanya meningkat secara konsisten dan konvergen pada nilai yang tinggi di atas 99%, dengan *gap* yang sangat kecil antara keduanya. Kedekatan antara kurva pelatihan dan validasi menunjukkan bahwa model memiliki kemampuan generalisasi yang baik serta tidak mengalami *overfitting* meskipun distribusi data telah dimodifikasi melalui proses *resampling*.

## 4.5 Hasil Evaluasi Model

Pada tahap ini dilakukan evaluasi performa model untuk mengukur kemampuan deteksi serangan pada dataset NSL-KDD. Evaluasi dilakukan menggunakan tiga pendekatan utama, yaitu: (1) analisis performa model final beserta *confusion matrix*-nya; (2) analisis ablasi komponen untuk mengidentifikasi kontribusi masing-masing elemen arsitektur dan strategi *resampling*; dan (3) perbandingan dengan *existing methods* menggunakan dataset yang sama. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall* (sensitivity), *specificity*, F1-*score*, dan *False Positive Rate* (FPR) dengan penekanan pada *recall* dan *specificity* sebagai indikator kinerja utama dalam konteks IDS.

### 4.5.1 Perbandingan Kinerja Model dan Confusion Matrix

Hasil evaluasi model final AE-CNN-BiLSTM dengan Hybrid Resampling berbasis *anomaly score* pada data uji (25.195 sampel) ditampilkan menggunakan *confusion matrix* pada Gambar 4.9 dan metrik evaluasi pada Tabel 4.6 dan 4.7.



Gambar 4.9 Confusion Matrix Model AE-CNN-BiLSTM

Hasil *confusion matrix* pada Gambar 4.9 menunjukkan bahwa model berhasil mengklasifikasikan sebagian besar sampel ke kelas yang benar dengan kesalahan klasifikasi yang sangat minimal pada kelas mayoritas. Metrik evaluasi per kelas dari *confusion matrix* tersebut ditampilkan secara rinci pada Tabel 4.6.

Tabel 4.6 Metrik Evaluasi Per Kelas Model AE-CNN-BiLSTM

| Kelas | TP | FN | FP | TN | Sensitivity (Recall) | Specificity | Precision | F1-Score | FPR |
|---|---|---|---|---|---|---|---|---|---|
| Normal | 13310 | 159 | 11 | 11715 | 0.9882 | 0.9991 | 0.9992 | 0.9937 | 0.0009 |
| DoS | 9183 | 3 | 14 | 15995 | 0.9997 | 0.9991 | 0.9985 | 0.9991 | 0.0009 |
| Probe | 2321 | 10 | 41 | 22823 | 0.9957 | 0.9982 | 0.9826 | 0.9891 | 0.0018 |
| R2L | 198 | 1 | 90 | 24906 | 0.9950 | 0.9964 | 0.6875 | 0.8131 | 0.0036 |
| U2R | 9 | 1 | 18 | 25167 | 0.9000 | 0.9993 | 0.3333 | 0.4865 | 0.0007 |
| Macro Avg | | | | | 0.9757 | 0.9984 | 0.8002 | 0.8563 | 0.0016 |

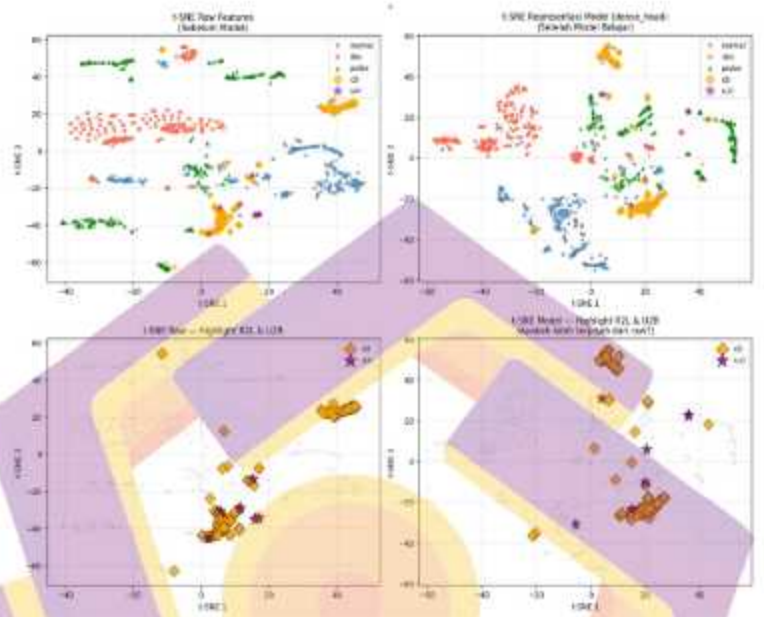
Berdasarkan Tabel 4.6, model final mencapai *Macro Sensitivity* (OVR) sebesar 0,9757 yang berarti model berhasil mendeteksi 97,57% dari seluruh serangan nyata pada dataset uji. Capaian yang paling signifikan terlihat pada kelas minoritas: Recall R2L mencapai 0,9950 (hanya 1 dari 199 sampel R2L tidak terdeteksi) dan Recall U2R mencapai 0,9000 (9 dari 10 sampel U2R berhasil terdeteksi). Sementara itu, *Macro Specificity* sebesar 0,9984 menunjukkan bahwa kemampuan model dalam mengenali trafik normal tidak terganggu secara signifikan. Nilai *precision* yang rendah pada U2R (0,3333) dan R2L (0,6875) merupakan *trade-off* yang dapat diterima dalam konteks IDS, karena *false negative* (serangan tidak terdeteksi) memiliki konsekuensi yang jauh lebih serius dibandingkan *false positive* (alarm palsu). Ringkasan metrik evaluasi keseluruhan pada data validasi dan data uji ditampilkan pada Tabel 4.7.

Tabel 4.7 Ringkasan Metrik Evaluasi Keseluruhan

| Split | Accuracy | Precision | Recall | F1 | Specificity | FPR |
|---|---|---|---|---|---|---|
| Validasi | 0.9939 | 0.8142 | 0.9528 | 0.8644 | 0.9986 | 0.0014 |
| Test | 0.9931 | 0.8002 | 0.9757 | 0.8563 | 0.9984 | 0.0016 |

Berdasarkan Tabel 4.7, terlihat bahwa performa model pada data validasi dan data uji sangat konsisten pada perbedaan Sensitivity antara validasi (0,9528) dan test (0,9757) menunjukkan bahwa model mampu generalisasi dengan baik, bahkan sedikit lebih baik pada data uji. Hal ini menunjukkan bahwa model tidak mengalami overfitting terhadap data validasi dan memiliki kemampuan generalisasi yang handal.

Untuk memberikan visualisasi yang lebih mendalam tentang kemampuan model dalam memisahkan antar kelas, dilakukan analisis t-SNE (t-Distributed Stochastic Neighbor Embedding) yang ditampilkan pada Gambar 4.8.



Gambar 4.8 t-SNE Raw Features vs Representasi Model

Gambar 4.8 menampilkan empat panel visualisasi t-SNE: (1) *raw features* sebelum model belajar, (2) representasi *dense_head* setelah model belajar, (3) *highlight* R2L & U2R pada *raw features*, dan (4) *highlight* R2L & U2R pada representasi model. Terlihat bahwa pada *raw features*, kelas R2L dan U2R bercampur dengan kelas lain dan sulit dibedakan secara visual. Setelah model belajar, representasi dari *dense_head* menunjukkan pemisahan yang jauh lebih jelas antar kelas, khususnya kelas minoritas R2L dan U2R yang kini lebih terpisah dari kelas mayoritas. Hal ini mengkonfirmasi bahwa arsitektur AE-CNN-BiLSTM berhasil mempelajari representasi fitur yang diskriminatif untuk membedakan serangan minoritas dari trafik normal.

### 4.5.2 Analisis Ablasi Komponen Utama

Untuk membuktikan kontribusi masing-masing komponen utama secara empiris, dilakukan perbandingan sistematis antar skenario eksperimen dengan kondisi yang identik kecuali satu variabel yang diuji. Hasil ablasi komponen utama ditampilkan pada Tabel 4.8.

Tabel 4.8 Ablasi Komponen Utama

| Skenario | Model | Accuracy | Macro Sensitivity | R2L Recall | U2R Recall | F1 Macro | Score Komposit |
|---|---|---|---|---|---|---|---|
| S1 | AE-CNN-BiLSTM Tanpa Resampling | 0.9952 | 0.8789 | 0.809 | 0.6 | 0.8817 | 0.9387 |
| S2 | CNN-BiLSTM + Hybrid RUS-SMOTE Tanpa AE | 0.9922 | 0.9687 | 0.9648 | 0.9 | 0.8495 | 0.9834 |
| S3 | AE-CNN-BiLSTM + SMOTE Rata (Tanpa Anomaly Score) | 0.9911 | 0.929 | 0.9648 | 0.7 | 0.7981 | 0.9635 |
| S4 | AE-CNN-BiLSTM + Hybrid RUS-SMOTE + Anomaly Score (FINAL) | 0.9931 | 0.9757 | 0.995 | 0.9 | 0.8563 | 0.9871 |

Berdasarkan Tabel 4.8, terdapat tiga perbandingan kritis yang menunjukkan kontribusi masing-masing komponen. Pertama, perbandingan S1 dengan S4 membuktikan kontribusi Hybrid Resampling berbasis anomaly score secara keseluruhan Macro Sensitivity meningkat +9,68% (0,8789→0,9757), Recall R2L meningkat +18,60% (0,8090→0,9950), dan Recall U2R meningkat +30,00% (0,6000→0,9000). Peningkatan ini menunjukkan bahwa pada dataset NSL-KDD yang sangat tidak seimbang, arsitektur yang kuat saja tidak cukup untuk mencapai deteksi yang memadai pada kelas minoritas tanpa strategi resampling yang adaptif.

Kedua, perbandingan S3 dengan S4 merupakan bukti paling langsung kontribusi anomaly score. Dengan arsitektur yang identik, penggunaan anomaly score berbasis RE Autoencoder meningkatkan Recall U2R dari 0,7000 menjadi 0,9000 (+20,00%) dan Macro Sensitivity dari 0,9290 menjadi 0,9757 (+4,67%). Peningkatan ini terjadi karena anomaly score memberikan faktor pengali 14,35× pada U2R (berbanding lebih kecil pada SMOTE rata), sehingga model mendapatkan representasi sintetis U2R yang jauh lebih proporsional terhadap tingkat anomalinya. Hal ini mengkonfirmasi bahwa anomaly score adalah mekanisme yang menjadikan strategi resampling lebih efisien dan adaptif.

Ketiga, perbandingan S2 dengan S4 menunjukkan kontribusi Autoencoder. Penambahan Autoencoder meningkatkan Recall R2L dari 0,9648 menjadi 0,9950 (+3,02%) dan Score Komposit dari 0,9834 menjadi 0,9871 (+0,37%). Meskipun peningkatannya lebih moderat, ini menunjukkan bahwa AE sebagai feature compressor menghasilkan representasi fitur yang lebih informatif yang secara konsisten meningkatkan kemampuan model membedakan pola R2L yang tersembunyi dari kelas lain.

### 4.5.3 Analisis Ablasi Attention Mechanism

Untuk menganalisis kontribusi mekanisme *attention* secara lebih rinci, dilakukan perbandingan antara model tanpa *attention*, model dengan satu *attention* (*FocusConv* atau *TempoNet*), dan model dengan *Dual Attention* (FocusConv + TempoNet). Hasil ditampilkan pada Tabel 4.9.

Tabel 4.9 Ablasi Attention Mechanism

| Skenario | Model | Accuracy | Macro Sensitivity | R2L Recall | U2R Recall | F1 Macro | Score Komposit |
|---|---|---|---|---|---|---|---|
| S5 | AE-CNN-BiLSTM + Hybrid RUS-SMOTE Tanpa ATT | 0.9945 | 0.9482 | 0.9598 | 0.8 | 0.8695 | 0.9733 |
| S6 | AE-CNN-BiLSTM + Hybrid RUS-SMOTE + ATT FocusConv | 0.9956 | 0.9479 | 0.9548 | 0.8 | 0.8775 | 0.9733 |
| S7 | AE-CNN-BiLSTM + Hybrid RUS-SMOTE + ATT TempoNet | 0.9934 | 0.9507 | 0.9749 | 0.8 | 0.8574 | 0.9745 |
| S4 | AE-CNN-BiLSTM + Hybrid RUS-SMOTE + Dual ATT (FINAL) | 0.9931 | 0.9757 | 0.995 | 0.9 | 0.8563 | 0.9871 |

Berdasarkan Tabel 4.9, terlihat bahwa penggunaan mekanisme *attention* tunggal baik FocusConv maupun TempoNet tidak memberikan peningkatan yang signifikan dibandingkan tanpa *attention* (*Score Komposit* bergerak dari 0,9733 ke 0,9733 dan 0,9745). Namun, ketika Dual ATT digunakan secara bersamaan, terjadi lompatan performa yang nyata: *Macro Sensitivity* naik dari 0,9482 menjadi 0,9757 (+2,75%), Recall R2L naik dari 0,9598 menjadi 0,9950 (+3,52%), dan Recall U2R naik dari 0,8000 menjadi 0,9000 (+10,00%). Hal ini mengkonfirmasi adanya efek sinergis antara FocusConv yang mengoptimalkan ekstraksi fitur lokal spasial dari CNN dan TempoNet yang mengoptimalkan pemodelan dependensi temporal dari BiLSTM kombinasi keduanya menghasilkan representasi fitur yang jauh lebih kaya dibandingkan penggunaan salah satu saja.

### 4.5.4 Perbandingan dengan Existing Methods

Untuk memvalidasi posisi kontribusi penelitian ini dalam lanskap IDS berbasis *deep learning*, dilakukan perbandingan *apple to apple* dengan penelitian

terdahulu yang menggunakan dataset NSL-KDD yang sama. Hasil perbandingan ditampilkan pada Tabel 4.10.

Tabel 4.10 Perbandingan dengan Existing Methods pada Dataset NSL-KDD

| Penelitian | Jurnal | Model | Resampling | Accuracy | Macro Sensitivity | R2L Recall | U2R Recall | F1 Macro |
|---|---|---|---|---|---|---|---|---|
| Dangol & Eaman (2025) | Procedia CS | CNN-BLSTM | ADASYN+TomekLinks | 81.63% | 82.00% | n/a | n/a | 79.00% |
| Ben Said et al. (2023) | IEEE Access | CNN-BiLSTM + Hybrid FS | RandomOS | 98.42% | n/a | 74.31% | 100% | n/a |
| Elsayed et al. (2024) | IEEE Access | CNN-LSTM | n/a | 97.96% | 97.79% | n/a | n/a | n/a |
| Penelitian Ini | - | AE-CNN-BiLSTM | RUS-SMOTE | 99.31% | 97.57% | 99.50% | 90.0% | 85.63% |

Berdasarkan Tabel 4.10, penelitian ini menunjukkan posisi yang kompetitif dibandingkan seluruh *existing methods* yang dibandingkan. Dibandingkan dengan Dangol & Eaman (2025) yang menggunakan arsitektur CNN-BLSTM identik pada NSL-KDD, penelitian ini menghasilkan peningkatan *Macro Sensitivity* dari 0,82 menjadi 0,9757 (+15,57%) dan *Accuracy* dari 81,63% menjadi 99,31% (+17,68%). Peningkatan yang sangat signifikan ini secara langsung membuktikan keunggulan Hybrid RUS-SMOTE berbasis *Anomaly Score* dibandingkan strategi ADASYN+TomekLinks yang menggunakan *fixed threshold* tanpa pertimbangan karakteristik anomali per kelas. Dibandingkan dengan Ben Said et al. (2023) yang menggunakan CNN-BiLSTM dengan *hybrid feature selection* namun tanpa *adaptive resampling*, penelitian ini menunjukkan peningkatan R2L Recall dari 74,31% menjadi 99,50% (+25,19%) membuktikan bahwa *anomaly score*-guided

resampling secara signifikan mengatasi kelemahan pendekatan *feature selection* yang tidak menangani *imbalanced data* secara adaptif.

### 4.5.5 Analisis Fitur Paling Berpengaruh

Analisis fitur paling berpengaruh dilakukan menggunakan metode Permutation Importance dengan 10 kali pengulangan pada data uji untuk mengidentifikasi fitur yang paling berkontribusi terhadap keputusan klasifikasi model AE-CNN-BiLSTM. Hasil analisis ditampilkan pada Tabel 4.11.

Tabel 4.11 Fitur Paling Berpengaruh pada Model AE-CNN-BiLSTM

| No | Nama Fitur | Kelompok Fitur | Tipe Data | Importance Mean |
|---|---|---|---|---|
| 1 | dst_host_diff_srv_rate | Traffic | Numerik | 0.0167 |
| 2 | diff_srv_rate | Time | Numerik | 0.0151 |
| 3 | wrong_fragment | Basic | Numerik | 0.0138 |
| 4 | hot | Content | Numerik | 0.0108 |
| 5 | dst_host_same_srv_rate | Traffic | Numerik | 0.0105 |
| 6 | dst_host_same_src_port_rate | Traffic | Numerik | 0.0089 |
| 7 | dst_host_count | Traffic | Numerik | 0.0085 |
| 8 | dst_host_serror_rate | Traffic | Numerik | 0.0058 |
| 9 | service_private | Basic | Kategorikal (OHE) | 0.0056 |
| 10 | dst_host_srv_count | Traffic | Numerik | 0.0056 |
| 11 | logged_in | Content | Numerik | 0.005 |
| 12 | service_eer_i | Basic | Kategorikal (OHE) | 0.0048 |
| 13 | srv_diff_host_rate | Time | Numerik | 0.004 |
| 14 | dst_host_rerror_rate | Traffic | Numerik | 0.0039 |
| 15 | service_eco_i | Basic | Kategorikal (OHE) | 0.0034 |

Berdasarkan Tabel 4.11, fitur kelompok Traffic mendominasi dengan 8 dari 15 posisi teratas, dipimpin dst_host_diff_srv_rate (0,0167) dan diff_srv_rate (0,0151) sebagai indikator utama serangan Probe yang menyebarkan koneksi ke berbagai layanan berbeda yang selaras dengan temuan Dangol & Eaman (2025) pada dataset NSL-KDD yang sama. Fitur wrong_fragment (rank 3) dan hot (rank 4) membuktikan model tidak hanya mengandalkan pola volumetrik tetapi juga mampu menangkap anomali pada level konten koneksi, yang kritis untuk mendeteksi DoS berbasis fragmentasi dan serangan R2L/U2R. Tiga fitur hasil One-Hot Encoding (service_private, service_ecr_i, service_eco_i) masuk dalam Top-15, mengkonfirmasi bahwa informasi kategorikal tetap relevan setelah transformasi OHE dan mendukung keputusan mempertahankan seluruh 41 fitur tanpa feature selection manual karena mekanisme FocusConv Attention terbukti mampu membobot fitur-fitur penting secara otomatis (Zhang et al., 2025).

## 4.6 Pembahasan

Pada bagian ini dilakukan pembahasan secara menyeluruh terhadap hasil penelitian yang telah diperoleh, dengan fokus pada peran masing-masing komponen model serta pengaruh *anomaly score* berbasis Autoencoder sebagai mekanisme kunci dalam strategi Hybrid Resampling RUS-SMOTE.

### 4.6.1 Analisis Anomaly Score sebagai Kunci Efisiensi Hybrid Resampling

Temuan sentral penelitian ini adalah bahwa *anomaly score* berbasis *reconstruction error* Autoencoder merupakan mekanisme yang menjadikan strategi *resampling* lebih efisien dan adaptif. Bukti empiris dari perbandingan S3 vs S4 (Tabel 4.8) menunjukkan bahwa dengan total data yang lebih efisien, model dengan *anomaly score* menghasilkan Recall U2R 20,00% lebih tinggi dibandingkan

SMOTE rata. Efisiensi ini terjadi karena *anomaly score* memastikan distribusi data sintetis proporsional terhadap tingkat anomali tiap kelas bukan sekadar terhadap rasio ketidakseimbangan. Kelas U2R yang memiliki RE median 94,5× lebih tinggi dari Normal mendapatkan faktor 14,35×, sementara R2L dan Probe mendapat faktor lebih rendah sesuai karakteristik anomalinya masing-masing. Dengan demikian *anomaly score* bukan sekadar parameter tambahan, melainkan jembatan antara pemahaman Autoencoder tentang distribusi data dan keputusan tentang berapa banyak data sintetis yang diperlukan untuk setiap kelas dan sesuatu yang tidak dapat dicapai oleh metode *resampling* konvensional berbasis *fixed threshold* (Dangol & Eaman, 2025; Bagui & Li, 2021).

4.6.2 Analisis Anomaly Score sebagai Kunci Efisiensi Hybrid Resampling

Hasil Skenario 1 (tanpa *resampling*) memberikan gambaran yang jelas tentang dampak *imbalanced data* pada NSL-KDD. Meskipun arsitektur AE-CNN-BiLSTM sudah *powerful* dengan *accuracy* 99,52%, Recall U2R hanya mencapai 0,6000 dan Recall R2L hanya 0,8090 membuktikan bahwa *accuracy* bersifat *misleading* pada data tidak seimbang sebagaimana dikonfirmasi Elsayed et al. (2024). Model yang mencapai akurasi >99% pun masih gagal mendeteksi 40% serangan U2R, yang dalam konteks IDS berarti 4 dari setiap 10 serangan *user-to-root* lolos tanpa terdeteksi. Dalam konteks keamanan siber, kegagalan ini memiliki konsekuensi yang jauh lebih serius dibandingkan *false alarm*, karena serangan U2R yang tidak terdeteksi berpotensi memberikan akses *root* kepada penyerang tanpa sepengetahuan tim keamanan (Zhu et al., 2023). Hal ini semakin memperkuat alasan mengapa *recall* dipilih sebagai metrik evaluasi primer dalam penelitian ini.

### 4.6.3 Analisis Peran Autoencoder sebagai Feature Extractor

Perbandingan S2 (tanpa AE) vs S4 menunjukkan bahwa Autoencoder memberikan peningkatan konsisten terutama pada Recall R2L (+3,02%) dan *Score Komposit* (+0,37%). Peran AE sebagai *feature compressor* yang menghasilkan representasi *bottleneck* yang telah didenoisifikasi membantu CNN dan BiLSTM berikutnya bekerja dengan representasi fitur yang lebih bersih dan informatif. Khususnya untuk kelas R2L yang memiliki pola serangan yang relatif tersembunyi dalam *noise* data jaringan, representasi *bottleneck* dari AE membantu model fokus pada fitur-fitur yang benar-benar membedakan R2L dari kelas Normal. Hal ini sejalan dengan temuan Yu et al. (2023) bahwa Autoencoder sebagai *feature compressor* mampu mereduksi dampak *redundant features* dan meningkatkan kualitas representasi untuk klasifikasi intrusi. Visualisasi t-SNE pada Gambar 4.8 juga mengkonfirmasi bahwa representasi yang dipelajari model jauh lebih terpisah antar kelas dibandingkan *raw features*, khususnya untuk kelas minoritas R2L dan U2R.

### 4.6.4 Analisis Efek Sinergis Dual Attention Mechanism

Hasil ablasi *attention mechanism* (Tabel 4.9) mengungkapkan bahwa FocusConv dan TempoNet secara individual tidak memberikan peningkatan yang signifikan, namun kombinasinya menghasilkan lompatan performa yang nyata. Efek sinergis ini dapat dijelaskan secara mekanistik: FocusConv yang berfokus pada fitur spasial lokal dari CNN menghasilkan representasi paket jaringan yang lebih presisi, sementara TempoNet yang berfokus pada pola temporal dari BiLSTM menghasilkan pemahaman konteks sekuensial yang lebih dalam. Ketika keduanya

bekerja bersamaan, model dapat secara simultan mempertimbangkan karakteristik lokal paket dan konteks temporalnya menghasilkan representasi fitur yang jauh lebih kaya untuk membedakan serangan subtil seperti U2R dan R2L dari trafik normal. Hal ini konsisten dengan prinsip *hierarchical feature fusion* yang diusulkan Zhang et al. (2025) dalam arsitektur *Dual-Attention* CNN-BiLSTM.

4.6.5 Analisis Recall sebagai Metrik Utama IDS

Penelitian ini secara konsisten mengutamakan *recall* (sensitivity) sebagai metrik evaluasi primer, dan hasil eksperimen mengkonfirmasi ketepatan pilihan ini. Model final mencapai *Macro Sensitivity* 0,9757 yang artinya 97,57% dari semua serangan nyata berhasil terdeteksi dengan *Macro Specificity* 0,9984 yang memastikan bahwa kemampuan mengenali trafik normal tidak terganggu secara signifikan. *Score Komposit* (0,5×Sensitivity + 0,5×Specificity) = 0,9871 menunjukkan keseimbangan terbaik di antara seluruh skenario eksperimen. Dibandingkan dengan *existing methods* pada NSL-KDD, penelitian ini menunjukkan peningkatan yang paling signifikan justru pada metrik yang paling relevan untuk IDS: R2L Recall dari 74,31% (Ben Said et al., 2023) menjadi 99,50%, dan peningkatan *Macro Sensitivity* dari 0,82 (Dangol & Eaman, 2025) menjadi 0,9757. Hal ini membuktikan bahwa kontribusi utama penelitian yaitu *anomaly score*-guided Hybrid *Resampling* secara efektif mengisi celah yang ditinggalkan oleh *existing methods* yang tidak menangani kelas minoritas secara adaptif berdasarkan karakteristik anomali per kelas.

# BAB 5
# PENUTUP

## 5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Ketidakseimbangan data (*imbalanced data*) pada dataset NSL-KDD terbukti memberikan dampak yang sangat signifikan terhadap kemampuan model dalam mendeteksi kelas serangan minoritas. Tanpa penanganan *resampling*, model AE-CNN-BiLSTM menghasilkan *recall* U2R hanya 0,6000 dan *recall* R2L hanya 0,8090 meskipun *accuracy* mencapai 0,9952 membuktikan bahwa *accuracy* bersifat *misleading* pada kondisi data tidak seimbang sehingga *recall* dan *specificity* perlu dijadikan metrik evaluasi primer dalam konteks IDS.
2. *Anomaly score* berbasis *reconstruction error* Autoencoder terbukti menjadi mekanisme kunci yang menjadikan strategi Hybrid Resampling RUS-SMOTE lebih efisien dan adaptif. Dengan nilai median RE U2R sebesar 0,091631 dimana hampir 53,9× lebih tinggi dari Normal intensitas *oversampling* ditentukan secara proporsional terhadap karakteristik anomali tiap kelas, menghasilkan peningkatan Recall U2R sebesar +20,00% dibandingkan SMOTE rata tanpa *anomaly score*.
3. Penerapan Hybrid Resampling RUS-SMOTE berbasis *anomaly score* secara keseluruhan terbukti meningkatkan *Macro Sensitivity* sebesar

+9,68% (0,8789 menjadi 0,9757), Recall R2L sebesar +18,60% (0,8090 menjadi 0,9950), dan Recall U2R sebesar +30,00% (0,6000 menjadi 0,9000) dibandingkan kondisi tanpa *resampling*, dengan *final imbalance ratio* yang berhasil direduksi dari 1.306× menjadi 36,45×.

4. Model final AE-CNN-BiLSTM dengan Hybrid Resampling RUS-SMOTE berbasis *anomaly score* dan *Dual Attention* menghasilkan performa yang meningkat dibandingkan *existing methods* pada dataset NSL-KDD yang mencapai *Accuracy* 99,31%, *Macro Sensitivity* 0,9757, *Macro Specificity* 0,9984, Recall R2L 0,9950, dan Recall U2R 0,9000. Model ini mengungguli Dangol & Eaman (2025) sebesar +15,57% pada *Macro Sensitivity* dan Ben Said et al. (2023) sebesar +25,19% pada Recall R2L.

## 5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya adalah sebagai berikut:

1. Penelitian ini masih terbatas pada dataset NSL-KDD, sehingga disarankan untuk mengujikan model pada dataset IDS yang lebih modern seperti UNSW-NB15 dan CIC-IDS2017 untuk menguji kemampuan generalisasi model pada distribusi serangan yang berbeda.
2. Model masih menunjukkan *trade off* berupa *precision* yang rendah pada kelas U2R (0,3333) dan R2L (0,6875). Penelitian selanjutnya disarankan mengeksplorasi mekanisme *cost-sensitive learning* yang dikombinasikan dengan *anomaly score* resampling untuk mencapai keseimbangan yang lebih optimal antara *recall* dan *precision*.

3. *Anomaly score* dalam penelitian ini menggunakan median RE dari Autoencoder *fully connected*. Penelitian selanjutnya dapat mengeksplorasi *Variational Autoencoder* (VAE) atau metrik alternatif seperti *Mahalanobis Distance* untuk menghasilkan *anomaly score* yang lebih representatif terhadap karakteristik distribusi tiap kelas.
4. Penelitian ini belum mengakomodasi skenario deteksi secara *real-time* pada jaringan nyata. Penelitian selanjutnya disarankan untuk mengimplementasikan model pada lingkungan *edge computing* dengan mempertimbangkan aspek latensi inferensi dan kemampuan model beradaptasi terhadap pola serangan baru yang berubah secara dinamis.

## DAFTAR PUSTAKA


Disha, R. A., & Waheed, S. (2022). Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique. Cybersecurity, 5(1), 1–22. https://doi.org/10.1186/s42400-021-00103-8

Elreedy, D., Atiya, A. F., & Kamalov, F. (2024). A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. Machine Learning, 113(7), 4903–4923. https://doi.org/10.1007/s10994-022-06296-4

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022a). CNN-LSTM: Hybrid Deep Neural Network for Network Intrusion Detection System. IEEE Access, 10, 99837–99849. https://doi.org/10.1109/ACCESS.2022.3206425

Halbouni, A., Gunawan, T. S., Habaebi, M. H., Halbouni, M., Kartiwi, M., & Ahmad, R. (2022b). Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. IEEE Access, 10(Mi), 19572–19585. https://doi.org/10.1109/ACCESS.2022.3151248

Hamza, A., Attique Khan, M., Wang, S. H., Alqahtani, A., Alsubai, S., Binbusayyis, A., Hussein, H. S., Martinetz, T. M., & Alshazly, H. (2022). COVID-19 classification using chest X-ray images: A framework of CNN-LSTM and improved max value moth flame optimization. Frontiers in Public Health, 10. https://doi.org/10.3389/fpubh.2022.948205

Hnamte, V., Nhung-Nguyen, H., Hussain, J., & Hwa-Kim, Y. (2023). A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE. IEEE Access, 11, 37131–37148. https://doi.org/10.1109/ACCESS.2023.3266979

Hussen, N., Elghamrawy, S. M., Salem, M., & El-Desouky, A. I. (2023). A Fully Streaming Big Data Framework for Cyber Security Based on Optimized Deep Learning Algorithm. IEEE Access, 11, 65675–65688. https://doi.org/10.1109/ACCESS.2023.3281893

Ileberi, E., Sun, Y., & Wang, Z. (2022). A machine learning based credit card fraud detection using the GA algorithm for feature selection. Journal of Big Data, 9(1). https://doi.org/10.1186/s40537-022-00573-8

Mohammed, S. H., Jit Singh, M. S., Al-Jumaily, A., Islam, M. T., Islam, M. S., Alenezi, A. M., & Soliman, M. S. (2025). Dual-hybrid intrusion detection system to detect False Data Injection in smart grids. In PLoS ONE (Vol. 20, Issue 1 January). https://doi.org/10.1371/journal.pone.0316536

Montaha, S., Azam, S., Rafid, A. K. M. R. H., Hasan, M. Z., Karim, A., & Islam, A. (2022). TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study. IEEE Access, 10, 60039–60059. https://doi.org/10.1109/ACCESS.2022.3179577

Nandanwar, H., & Katarya, R. (2025). Securing Industry 5.0: An explainable deep learning model for intrusion detection in cyber-physical systems. Computers and Electrical Engineering, 123(PC), 110161. https://doi.org/10.1016/j.compeleceng.2025.110161

Okey, O. D., Melgarejo, D. C., Saadi, M., Rosa, R. L., Kleinschmidt, J. H., & Rodriguez, D. Z. (2023). Transfer Learning Approach to IDS on Cloud IoT Devices Using Optimized CNN. IEEE Access, 11(January), 1023–1038. https://doi.org/10.1109/ACCESS.2022.3233775

Petmezas, G., Cheimariotis, G. A., Stefanopoulos, L., Rocha, B., Paiva, R. P., Katsaggelos, A. K., & Maglaveras, N. (2022). Automated Lung Sound Classification Using a Hybrid CNN-LSTM Network and Focal Loss Function. Sensors, 22(3). https://doi.org/10.3390/s22031232

Prasath, S., Sethi, K., Mohanty, D., Bera, P., & Samantaray, S. R. (2022). Analysis of Continual Learning Models for Intrusion Detection System. IEEE Access, 10, 121444–121464. https://doi.org/10.1109/ACCESS.2022.3222715

Protić, D. (2018). Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets. Vojnotehnicki Glasnik, 66(3), 580–596. https://doi.org/10.5937/vojtehg66-16670

Suman, S., Singh, G., Sakla, N., Gattu, R., Green, J., Phatak, T., Samaras, D., & Prasanna, P. (2021). Attention Based CNN-LSTM Network for Pulmonary Embolism Prediction on Chest Computed Tomography Pulmonary Angiograms. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12907 LNCS, 356–366. https://doi.org/10.1007/978-3-030-87234-2_34

Wang, C., Wang, X., Jing, X., Yokoi, H., Huang, W., & Zhu, M. (2022). Towards high-accuracy classifying attention- deficit / hyperactivity disorders using CNN-LSTM model Towards high-accuracy classifying attention-deficit / hyperactivity disorders using CNN-LSTM model.

Wang, Z., Zhou, Y., Takagi, T., Song, J., Tian, Y. S., & Shibuya, T. (2023). Genetic algorithm-based feature selection with manifold learning for cancer classification using microarray data. BMC Bioinformatics, 24(1), 1–22. https://doi.org/10.1186/s12859-023-05267-3

Wu, L., Xie, Y., Li, J., Feng, D., Liang, J., & Wu, Y. (2025). Angus: efficient active learning strategies for provenance based intrusion detection. Cybersecurity, 8(1). https://doi.org/10.1186/s42400-024-00311-y

Wu, Z., Zhang, H., Wang, P., & Sun, Z. (2022). RTIDS: A Robust Transformer-Based Approach for Intrusion Detection System. IEEE Access, 10, 64375–64387. https://doi.org/10.1109/ACCESS.2022.3182333

Yu, H., Kang, C., Xiao, Y., & Yang, Y. (2023). Network Intrusion Detection Method Based on Hybrid Improved Residual Network Blocks and Bidirectional Gated Recurrent Units. IEEE Access, 11, 68961–68971.

https://doi.org/10.1109/ACCESS.2023.3271866

Zhao, R., Mu, Y., Zou, L., & Wen, X. (2022). A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier. IEEE Access, 10(July), 71414–71426. https://doi.org/10.1109/ACCESS.2022.3186975

Zhu, N., Zhao, G., Yang, Y., Yang, H., & Liu, Z. (2023). AEC_GAN: Unbalanced Data Processing Decision-Making in Network Attacks Based on ACGAN and Machine Learning. IEEE Access, 11(May), 52452–52465. https://doi.org/10.1109/ACCESS.2023.3280421

Suman, S., Singh, G., Sakla, N., Gattu, R., Green, J., Phatak, T., Samaras, D., & Prasanna, P. (2021). Attention Based CNN-LSTM Network for Pulmonary Embolism Prediction on Chest Computed Tomography Pulmonary Angiograms. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 12907 LNCS, 356–366. https://doi.org/10.1007/978-3-030-87234-2_34

Montaha, S., Azam, S., Rafid, A. K. M. R. H., Hasan, M. Z., Karim, A., & Islam, A. (2022). TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study. IEEE Access, 10, 60039–60059. https://doi.org/10.1109/ACCESS.2022.3179577

Wang, C., Wang, X., Jing, X., Yokoi, H., Huang, W., & Zhu, M. (2022). Towards high-accuracy classifying attention- deficit / hyperactivity disorders using CNN-LSTM model Towards high-accuracy classifying attention-deficit / hyperactivity disorders using CNN-LSTM model.

Petmezas, G., Cheimariotis, G. A., Stefanopoulos, L., Rocha, B., Paiva, R. P., Katsaggelos, A. K., & Maglaveras, N. (2022). Automated Lung Sound Classification Using a Hybrid CNN-LSTM Network and Focal Loss Function. Sensors, 22(3). https://doi.org/10.3390/s22031232

Hamza, A., Attique Khan, M., Wang, S. H., Alqahtani, A., Alsubai, S., Binbusayyis, A., Hussein, H. S., Martinetz, T. M., & Alshazly, H. (2022). COVID-19 classification using chest X-ray images: A framework of CNN-LSTM and improved max value moth flame optimization. Frontiers in Public Health, 10. https://doi.org/10.3389/fpubh.2022.948205

## LAMPIRAN

Lampiran 1 Dataset

| duration | protocol_type | service | flag | src_bytes | dst_bytes | land | wrong_fragment | urgent | hot | num_failed_logins |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | tcp | ftp_data | SF | 491 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | udp | other | SF | 146 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 232 | 8153 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 199 | 420 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | remote_job | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | REJ | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | private | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 287 | 2251 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | ftp_data | SF | 334 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | name | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | netbios_ns | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 300 | 13788 | 0 | 0 | 0 | 0 | 0 |
| 0 | icmp | eco_i | SF | 18 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 233 | 616 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | http | SF | 343 | 1178 | 0 | 0 | 0 | 0 | 0 |
| 0 | tcp | mtp | S0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| logged_in | num_compromised | root_shell | su_attempted | num_root | num_file_creations | num_shells | num_access_files | num_outbound_cmds | is_host_login | is_guest_login |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| count | srv_count | serror_rate | srv_serror_rate | rerror_rate | srv_rerror_rate | same_srv_rate | diff_srv_rate | srv_diff_host_rate | dst_host_count | dst_host_srv_count |
|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 150 | 25 |
| 13 | 1 | 0 | 0 | 0 | 0 | 0.08 | 0.15 | 0 | 255 | 1 |
| 123 | 6 | 1 | 1 | 0 | 0 | 0.05 | 0.07 | 0 | 255 | 26 |
| 5 | 5 | 0.2 | 0.2 | 0 | 0 | 1 | 0 | 0 | 30 | 255 |
| 30 | 32 | 0 | 0 | 0 | 0 | 1 | 0 | 0.09 | 255 | 255 |
| 121 | 19 | 0 | 0 | 1 | 1 | 0.16 | 0.06 | 0 | 255 | 19 |
| 166 | 9 | 1 | 1 | 0 | 0 | 0.05 | 0.06 | 0 | 255 | 9 |
| 117 | 16 | 1 | 1 | 0 | 0 | 0.14 | 0.06 | 0 | 255 | 15 |
| 270 | 23 | 1 | 1 | 0 | 0 | 0.09 | 0.05 | 0 | 255 | 23 |
| 133 | 8 | 1 | 1 | 0 | 0 | 0.06 | 0.06 | 0 | 255 | 13 |
| 205 | 12 | 0 | 0 | 1 | 1 | 0.06 | 0.06 | 0 | 255 | 12 |
| 199 | 3 | 1 | 1 | 0 | 0 | 0.02 | 0.06 | 0 | 255 | 13 |
| 3 | 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0.43 | 8 | 219 |
| 2 | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 2 | 20 |
| 233 | 1 | 1 | 1 | 0 | 0 | 0 | 0.06 | 0 | 255 | 1 |
| 96 | 16 | 1 | 1 | 0 | 0 | 0.17 | 0.05 | 0 | 255 | 2 |
| 8 | 9 | 0 | 0.11 | 0 | 0 | 1 | 0 | 0.22 | 91 | 255 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 16 |
| 3 | 3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 66 | 255 |
| 9 | 10 | 0 | 0 | 0 | 0 | 1 | 0 | 0.2 | 157 | 255 |
| 223 | 23 | 1 | 1 | 0 | 0 | 0.1 | 0.05 | 0 | 255 | 23 |
| 280 | 17 | 1 | 1 | 0 | 0 | 0.06 | 0.05 | 0 | 238 | 17 |
| 8 | 10 | 0 | 0 | 0 | 0 | 1 | 0 | 0.2 | 87 | 255 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 255 | 1 |
| 248 | 2 | 1 | 1 | 0 | 0 | 0.01 | 0.06 | 0 | 255 | 2 |
| 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 255 | 25 |
| 279 | 7 | 1 | 1 | 0 | 0 | 0.03 | 0.06 | 0 | 255 | 13 |
| 5 | 22 | 0 | 0 | 0 | 0 | 1 | 0 | 0.18 | 43 | 255 |
| 14 | 14 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 255 | 255 |

| dst_host_same_srv_rate | dst_host_diff_srv_rate | dst_host_same_src_port_rate | dst_host_srv_diff_host_rate | dst_host_serror_rate |
|---|---|---|---|---|
| 0.17 | 0.03 | 0.17 | 0 | 0 |
| 0 | 0.6 | 0.88 | 0 | 0 |
| 0.1 | 0.05 | 0 | 0 | 1 |
| 1 | 0 | 0.03 | 0.04 | 0.03 |
| 1 | 0 | 0 | 0 | 0 |
| 0.07 | 0.07 | 0 | 0 | 0 |
| 0.04 | 0.05 | 0 | 0 | 1 |
| 0.06 | 0.07 | 0 | 0 | 1 |
| 0.09 | 0.05 | 0 | 0 | 1 |
| 0.05 | 0.06 | 0 | 0 | 1 |
| 0.05 | 0.07 | 0 | 0 | 0 |
| 0.05 | 0.07 | 0 | 0 | 1 |
| 1 | 0 | 0.12 | 0.03 | 0 |
| 1 | 0 | 1 | 0.2 | 0 |
| 0 | 0.07 | 0 | 0 | 1 |
| 0.01 | 0.06 | 0 | 0 | 1 |
| 1 | 0 | 0.01 | 0.02 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 0 | 0.02 | 0.03 | 0 |
| 1 | 0 | 0.01 | 0.04 | 0 |
| 0.09 | 0.05 | 0 | 0 | 1 |
| 0.07 | 0.06 | 0 | 0 | 0.99 |
| 1 | 0 | 0.01 | 0.02 | 0 |
| 0 | 0.85 | 1 | 0 | 0 |
| 0.01 | 0.06 | 0 | 0 | 1 |
| 0.1 | 0.05 | 0 | 0 | 0.53 |
| 0.05 | 0.07 | 0 | 0 | 1 |
| 1 | 0 | 0.02 | 0.14 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0.51 | 0 |
| 0.23 | 0.04 | 0.01 | 0 | 1 |
| 1 | 0 | 0.11 | 0.01 | 0 |
| 0 | 0.31 | 0.28 | 0 | 0 |
| 0.98 | 0.01 | 0 | 0 | 0 |
| 0.12 | 0.05 | 0.05 | 0 | 0 |
| 1 | 0 | 0.01 | 0 | 0 |
| 0.07 | 0.07 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0.05 | 0.07 | 0 | 0 | 1 |
| 0.25 | 0.02 | 0.01 | 0 | 1 |
| 0.04 | 0.07 | 0 | 0 | 1 |
| 0.81 | 0.02 | 0.01 | 0 | 0 |

| dst_host_srv_serror_rate | dst_host_rerror_rate | dst_host_srv_rerror_rate | labels | Attack Classes |
|---|---|---|---|---|
| 0 | 0.05 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 0.01 | 0 | 0.01 | normal | |
| 0 | 0 | 0 | normal | |
| 0 | 1 | 1 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 1 | 1 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | warezclient | R2L |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | ipsweep | Probe |
| 0 | 0.02 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0.02 | 0.16 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0.56 | 0.57 | normal | |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | ipsweep | Probe |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0 | 0 | normal | |
| 0 | 0.29 | 1 | portsweep | Probe |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 0 | 0 | 0 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 0 | 0 | 0 | normal | |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |
| 1 | 0 | 0 | neptune | DoS |