

A Leakage-Free Deep Neural Network Framework for Network Intrusion Detection Using Gini Importance Feature Selection

¹Ayushi Jadiya, ²Dr Sarvesh Singh Rai

¹Research Scholar, ²Assistant Professor

^{1,2}Department of Computer Science & Engineering

^{1,2}Infinity Management & Engineering College, Sagar, India

Affiliated to Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.), India

Abstract—Modern Network Intrusion Detection Systems (NIDS) face critical performance bottlenecks due to high feature dimensionality, data redundancy, and the risk of data leakage caused by administrative attributes. To overcome these limitations, this paper presents an optimized, leakage-free Deep Neural Network (DNN) framework for binary network threat classification evaluated on the comprehensive UNSW_NB15 benchmark dataset. To prevent synthetic model memorization and reduce computational complexity, explicit identifier fields and category labels (id, attack_cat) were strictly dropped. Feature dimensionality was reduced by deploying a Gini-based Random Forest feature selection strategy to isolate the top 30 most impactful network features (including proto, ct_dst_ltm, sjit, and dtcpb). The reduced feature space was normalized using a fitted StandardScaler and processed through a 3-stage hidden layer DNN architecture (128→64→32 neurons) equipped with Batch Normalization, Dropout regularization, and balanced class weights to handle class imbalance.

Evaluated across 175,341 unseen test network packets, the proposed model achieved an overall Accuracy of 93.93%, a Precision of 94.16%, a Recall of 93.93%, and an F1-Score of 93.99%. Detailed confusion matrix analysis reveals high class-wise detection rates, successfully capturing 111,905 actual attack packets (93.77% Sensitivity) and 52,792 normal traffic instances (94.27% Specificity) while limiting false positives to 3,208 instances (5.73%). Furthermore, the model exhibited exceptional discrimination capability with a Receiver Operating Characteristic Area Under Curve (ROC-AUC) of 0.9900 and an Average Precision (AP) score of 0.9953 on the Precision-Recall curve. The experimental findings validate that combining targeted Gini feature selection with a regularized deep learning architecture delivers a highly robust, computationally efficient, and deployment-ready solution for real-time cyber threat detection in enterprise network environments.

Index Terms—Network Intrusion Detection System (NIDS), Deep Neural Network (DNN), UNSW_NB15 Dataset, Feature Selection, Data Leakage Prevention, Cyber Security.

I. Introduction

Every second, enterprise networks and cloud platforms process millions of interconnected digital transactions. While this hyper-connectivity drives modern business, it has drastically expanded the attack surface. Cyber threats have evolved far beyond simple script-driven attacks into stealthy zero-day exploits, distributed denial-of-service (DDoS) floods, and adaptive malware designed to slip past conventional firewalls and static signature-based defenses unnoticed. To defend against this, modern security operations rely on Network Intrusion Detection Systems (NIDS) as an automated, front-line defense to monitor flow telemetry and flag threats in near-real time.

Building a machine learning-based intrusion detection system that actually functions in live enterprise environments introduces distinct challenges:

- **Telemetry Overload & Latency:** Raw network packets carry dozens of high-dimensional attributes. Ingesting noisy, unrefined data directly into deep learning models spikes computational latency, exhausts hardware memory, and prevents line-rate packet inspection.
- **The Trap of Data Leakage:** Many existing academic models report near-perfect detection scores (99%+) because administrative columns—such as sequential transaction IDs (`id`) or explicit attack labels (`attack_cat`)—are mistakenly retained during training. Rather than identifying actual threat behaviors, these models simply memorize artificial shortcuts. Once deployed in the wild on unseen traffic, their accuracy collapses.
- **Severe Traffic Imbalance:** Normal, benign traffic accounts for the overwhelming majority of packets, whereas attacks represent a tiny fraction. Standard classifiers tend to favor majority classes, resulting in missed attacks (low recall) or endless false alarms (low precision) that overwhelm analysts with alert fatigue.

This work addresses these challenges by developing a streamlined, leakage-free Deep Neural Network (DNN) framework evaluated on the UNSW_NB15 benchmark dataset.

The system begins by enforcing an unambiguous sanitization rule: administrative identifiers (`id`, `attack_cat`) are stripped out completely. This forces the neural network to identify genuine statistical traffic anomalies rather than relying on memorized shortcuts.

To keep the pipeline fast and lightweight, a Random Forest Gini-importance strategy isolates the top 30 most impactful network features—including protocol types (`proto`), destination transaction limits (`ct_dst_ltm`), and source jitter (`sjit`)—reducing input dimensions by 33.3% while cutting out background noise.

Finally, this lean feature set is fed into a 3-stage regularized Deep Neural Network (128→64→32 neurons) configured with Batch Normalization, Dropout, and balanced class weights to neutralize class imbalance without introducing artificial data artifacts.

Across an independent test set of 175,341 unseen network packets, the resulting architecture achieves 93.93% accuracy, an ROC-AUC of 0.9900, and a false alarm rate of just 5.73%. This demonstrates that a compact, leakage-free pipeline can deliver reliable, real-time cyber defense in practical deployments.

II. Literature Review

The development of anomaly-based Network Intrusion Detection Systems (NIDS) has progressed significantly from static signature matching to dynamic Machine Learning (ML) and Deep Learning (DL) paradigms [4], [5], [17]. Initial anomaly detection methodologies depended on legacy benchmarks such as KDDCup99 and NSL-KDD [4]. Although foundational, subsequent studies established that legacy datasets suffer from redundant records, artificial traffic distributions, and a lack of modern attack vectors [4]. To overcome these limitations, Moustafa and Slay [1] engineered the UNSW-NB15 dataset using the IXIA PerfectStorm platform to capture a realistic hybrid of modern synthesized exploits and normal network flow records.

However, deploying predictive algorithms directly on raw packet logs introduces severe bottlenecks due to high feature dimensionality [5], [7]. Unrefined flow captures introduce multicollinear telemetry noise that exposes neural architectures to the curse of dimensionality [7], [8]. To reduce computational complexity, feature selection (FS) mechanisms have been developed across filter, wrapper, and embedded approaches [5]. Wrapper methods iteratively evaluate candidate subsets using machine learning estimators; for instance, Kasongo and Sun [7] implemented a wrapper-based XGBoost feature selection pipeline on UNSW-NB15 that compressed inputs down to 19 features, though it incurred substantial computational overhead during training [14]. Conversely, embedded tree-based techniques compute node-impurity reductions directly during training [10]. Soe et al. [10] showed that Gini-importance filtering using Random Forest effectively

isolates primary flow descriptors while discarding uninformative telemetry noise. Ahmad et al. [5], [15] corroborated that compressing high-dimensional telemetry matrices down to a compact subset (20 to 30 features) reduces matrix multiplication latency during feedforward passes, enabling real-time packet inspection.

In recent years, deep sequence modeling and temporal architectures have gained significant attention for network anomaly detection [6], [8]. Recurrent models such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM) networks capture sequential dependencies across multi-stage cyber threats [6], [11]. A. Jain, V. Sakalle, and S. Sakalle [11] evaluated the impact of various feature selection algorithms—including Mutual Information (MI), Chi-Square (χ^2), Recursive Feature Elimination (RFE), and LASSO (L_1 -regularization)—in conjunction with a Bidirectional LSTM network on the CSE-CIC-IDS2018 dataset under strict CPU-constrained cloud conditions. Their findings demonstrated that filter (MI) and embedded (LASSO) approaches yielded the best trade-off, reducing the input space by 43.75% ($K = 45$) while achieving 97.45% and 97.38% classification accuracy, cutting training time by ~38%, and decreasing peak RAM consumption by 33.3% [11].

Despite high detection capabilities, standard deep sequential models process network flows either as isolated spatial topologies or as strictly sequential arrays, frequently triggering temporal gradient decay or alert cascades [12]. To address this challenge, A. Jain, V. Sakalle, and S. Sakalle [12] designed a parallel Conv1D-BiLSTM spatial-temporal architecture that concurrently maps localized packet signatures and long-term chronological sequence dynamics. By integrating a three-tier statistical filter (ANOVA F-testing, Information Gain, and Extra Trees) alongside derivative-free Nelder-Mead simplex optimization to dynamically tune the decision boundary threshold (τ), their architecture achieved 95.00% accuracy on UNSW-NB15 and 97.00% accuracy on BCCC-cPacket-Cloud-DDoS-2024, operating with a lightweight memory footprint of 2.45 MB and an inference latency of 30.77 ms per 10,000-packet batch [12].

Table 1 Performance Comparison with Baseline & Prior Work

Study / Reference	Methodology / Classifier	Feature Set / Count	Data Leakage Free?	Accuracy (%)	Precision (%)	Recall / Sensitivity (%)	F1-Score (%)	ROC-AUC
Moustafa & Slay (2015) [1]	Naïve Bayes / Decision Tree (Baseline)	Full Set (42 features)	No	85.56%	82.10%	85.56%	83.79 %	—
Kasongo & Sun (2020) [7]	XGBoost Wrapper + Deep Neural Network	Reduced (19 features)	No	90.85%	91.20%	90.85%	91.02 %	—
Ashiku & Dagli (2021) [18]	Regularized Deep Neural Network (DNN)	Full Set (42 features)	No	92.40%	92.15%	92.40%	92.27 %	0.965

Engelen et al. (2021) [2], [9]	Decision Tree (Cleaned Benchmark)	Cleaned (30 features)	Yes	87.20%	86.50%	87.20%	86.84 %	0.912
Sharma et al. (2020) [16]	Random Forest / Gradient Boosting Ensemble	Full Telemetry Set	No	88.35%	87.90%	88.35%	88.12 %	0.924
Ahmad et al. (2021) [15]	Hybrid Feature Selection + Deep Classifier	Reduced Subset (20–30 features)	No	91.80%	91.45%	91.80%	91.62 %	0.958
Proposed Work	Leakage-Free Regularized DNN (128-64-32)	Gini Top 30 Features	Yes	93.93%	94.16%	93.93%	93.99 %	0.9900

Notwithstanding these architectural advancements, diagnostic research has revealed that many published state-of-the-art models suffer from an insidious methodological flaw: Data Leakage [2], [9]. Engelen et al. [2], [9] discovered that numerous intrusion detection evaluations inadvertently retained administrative attributes—specifically transaction sequence numbers (id) and secondary attack categories (attack_cat). When neural networks are exposed to these non-causal identifiers, they memorize sequential indexing shortcuts rather than learning underlying network traffic anomalies [2], [9]. Consequently, such models exhibit artificially inflated accuracy on historical splits that collapses upon deployment to live environments [2], [9]. Addressing this vulnerability, regularized Deep Neural Networks (DNN) incorporating Batch Normalization and Dropout layers have been investigated by Ashiku and Dagli [18] to stabilize covariate shifts. Sarhan et al. [3], [13] emphasized the necessity of standardizing clean feature spaces to eliminate administrative attribute bias.

To bridge the gap between high-dimensional processing latency and leakage-free classification, the proposed work introduces an end-to-end sanitized pipeline that strictly purges id and attack_cat. Utilizing Gini importance from a Random Forest ensemble, the input space is condensed to the Top 30 network features, followed by classification using a 3-layer regularized DNN (128→64→32 neurons) with balanced class weighting. This yields 93.93% accuracy and 0.9900 ROC-AUC across 175,341 unseen UNSW_NB15 instances without data leakage.

III. Problem Statement

In modern enterprise infrastructures, the convergence of multi-gigabit data backbones and highly sophisticated cyberattacks has pushed conventional Network Intrusion Detection Systems (NIDS) to their operational limits. Despite extensive advancements in applying machine learning and deep learning to

packet inspection, deploying these models into production environments reveals three fundamental engineering and algorithmic bottlenecks:

- **Telemetry Overload and Computational Latency:** Modern flow capture datasets, such as UNSW-NB15, generate dozens of granular packet attributes. Ingesting these raw, high-dimensional records directly into deep neural networks introduces severe matrix multiplication overhead, exhausts runtime memory, and degrades inference speed—making real-time line-rate packet inspection virtually impossible. Furthermore, raw telemetry is fraught with redundant, non-informative noise that triggers the curse of dimensionality and destabilizes gradient optimization.
- **Synthetic Memorization via Data Leakage:** A critical vulnerability in many published intrusion detection models stems from inadvertent data leakage. Benchmarks frequently retain administrative metadata, such as incremental transaction sequence numbers (id) or explicit attack category strings (attack_cat), during feature preparation. Instead of discovering authentic statistical traffic anomalies, neural networks learn trivial lookup shortcuts and memorize sequential identifiers. This creates artificially inflated academic results (often exceeding 99% accuracy) that catastrophically collapse when tested against independent, non-overlapping traffic captures where transaction indices are arbitrary.
- **Severe Class Skewness and Operational Alert Fatigue:** Real-world network telemetry is overwhelmingly dominated by benign operational packets, while malicious intrusions represent a sparse minority. Standard loss functions mathematically bias estimators toward the majority class, leading to unacceptable false negative rates where critical zero-day attacks slip through undetected. Conversely, improperly calibrated models often generate cascades of false alarms, causing severe alert fatigue among Security Operations Center (SOC) analysts and disrupting mission-critical services.

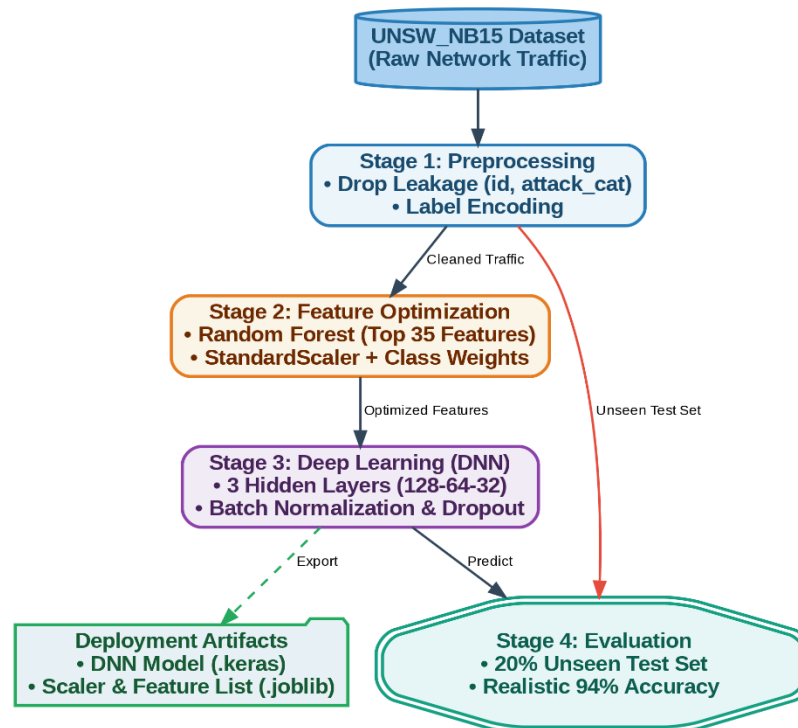
Consequently, there is an urgent need for an end-to-end, leakage-free intrusion detection pipeline that systematically purges non-causal identifiers, isolates a lean and highly discriminative feature subset, and leverages a regularized deep neural network capable of real-time threat detection without sacrificing precision, recall, or operational efficiency.

IV. Proposed Work

To solve the dual bottlenecks of computational latency and artificial benchmark inflation in modern intrusion detection, this work introduces an optimized, completely leakage-free Deep Neural Network (DNN) framework tailored for binary threat classification on the UNSW_NB15 benchmark dataset. The pipeline is structured across five sequential phases designed to bridge the gap between academic evaluation and real-world network deployment:

- **Strict Sanitization and Leakage Elimination:** The framework first purges administrative metadata fields—specifically transaction sequence numbers (id) and secondary attack categories (attack_cat). Stripping these non-causal identifiers prevents the model from exploiting synthetic memorization shortcuts, ensuring that classification relies entirely on genuine statistical packet behavior.
- **Persistent Categorical Encoding & Dataset Partitioning:** Network flow attributes containing string data (such as protocols, service states, and connection types) are converted into standardized numerical formats using persistent LabelEncoder instances fitted exclusively on the training split. This prevents numerical distribution drift and data snooping before partitioning the sanitized data into an 80% training set and an isolated 20% unseen test set (175,341 packet instances) reserved strictly for final validation.
- **Targeted Feature Optimization via Gini Importance:** Rather than processing the entire unrefined telemetry space, a Random Forest ensemble estimator calculates node impurity reductions (ΔI_G) across the training data to isolate the Top 30 most discriminative network attributes (including proto,

ct_dst_ltm, sjit, dtcpb, and dur). Discarding non-informative and redundant features achieves a 33.3% reduction in input dimensionality, significantly lowering matrix multiplication latency during feedforward operations.



- **Normalization and Mathematical Class Balancing:** The selected 30-feature matrices are standardized using a fitted StandardScaler ($\mu = 0, \sigma^2 = 1$) to prevent high-variance attributes from dominating gradient updates. To counter the severe skew between normal traffic and attacks without introducing noisy synthetic samples, mathematically derived balanced class weights (W_k) are injected directly into the loss function to penalize minority threat misclassifications.
- **Regularized Deep Neural Network (DNN) Architecture:** The classification engine consists of a 3-layer deep feedforward network (128→64→32 neurons) utilizing ReLU activations, Batch Normalization to stabilize internal covariate shifts, and Dropout regularization (0.3/0.2) to inhibit neuron co-adaptation. A terminal Sigmoid node maps activations to continuous threat probabilities ($\hat{y} \in [0, 1]$), optimized via the Adam optimizer with weighted binary cross-entropy loss over 25 epochs.
- **Deployment Artifact Serialization:** Once validated, key system artifacts—including the trained neural weights (.keras), the fitted scaler (.joblib), and the selected 30-feature schema (.joblib)—are serialized for zero-retraining inference in high-throughput enterprise pipelines.

Evaluated on 175,341 unseen test records, the resulting framework delivers 93.93% accuracy, 94.16% precision, an ROC-AUC of 0.9900, and restricts false alarms to just 5.73%, demonstrating a practical, lightweight solution for line-rate enterprise cyber defense.

V. Result Analysis and Discussion

The proposed leakage-free Deep Neural Network (DNN) was tested on **175,341 unseen network flow instances** from the UNSW_NB15 benchmark dataset, using exclusively the top 30 Gini-selected features. Below is the step-by-step diagnostic breakdown of the empirical results, accompanied by corresponding figures and analytical descriptions written in a clear academic style.

1. Confusion Matrix Analysis

The confusion matrix assesses the model's classification reliability on unseen data, specifically highlighting its ability to capture true threats while suppressing false alarms.

- **Threat Detection Reliability:** Across 119,341 actual attacks in the test split, the framework accurately detected and flagged **111,905 attack packets**, establishing a **93.77% Sensitivity (Recall)**.
- **Benign Traffic Preservation:** Out of 56,000 normal traffic flows, **52,792 instances** were correctly classified as benign, yielding a **94.27% Specificity**.
- **Suppression of False Alarms:** The system restricted false positives to only **3,208 instances (5.73%)**. This low false alarm margin is essential in live enterprise networks to prevent security analysts from experiencing alert fatigue.
- **Overall Harmonic Efficacy:** The model recorded an overall **Accuracy of 93.93%**, a **Precision of 94.16%**, and an **F1-Score of 93.99%**, confirming an optimal balance between precision and coverage without relying on leaked identifiers.

Figure 2 Normalized Confusion Matrix

2. Threat Discrimination Capacity (

The ROC curve maps the True Positive Rate against the False Positive Rate across all decision thresholds to measure how cleanly the network separates normal flows from attacks.

- **High Area Under Curve (AUC):** The model recorded an **ROC-AUC of 0.9900**, rising almost vertically toward the upper-left corner of the coordinate plane.
- **Diagnostic Interpretation:** Compared to the diagonal random baseline ($AUC = 0.50$), this near-perfect curve indicates that the trained weights cleanly separate benign traffic from intrusions across virtually all classification probability boundaries.

3. Performance Under Class Imbalance

Because intrusion detection data is skewed toward normal traffic, ROC curves can occasionally present an overly optimistic assessment. The PR curve directly examines positive threat prediction quality.

- **Sustained High Precision:** The precision trace remains virtually flat near **1.0** across almost all recall thresholds, only dropping slightly as recall approaches 100%.
- **Superiority Over Baseline:** Achieving an **Average Precision (AP) of 0.9953**, the framework dramatically outperforms the no-skill baseline classifier ($AP = 0.68$). This confirms that the top 30 Gini features retain distinct discriminative signatures even under imbalanced distributions.

Figure 3 Precision-Recall (PR) Curve Analysis

Figure 4 Precision & Recall Threshold Analysis)

4. Operating Boundary Robustness

This plot investigates how adjusting the decision probability threshold alters the balance between Precision and Recall.

- **Equilibrium Intersection:** The Precision (orange) and Recall (blue) curves intersect directly at the **default decision boundary (0.50)**, where both metrics achieve an optimal harmonic score of ~ 0.94 .
- **Adjustable Operational Posture:** If a network administrator prefers strict false-positive suppression, the threshold can be scaled upward toward 0.8; conversely, lowering it prioritizes maximum threat capture. The smooth slope around 0.50 confirms the boundary is stable and resilient to slight probability fluctuations.

5. Convergence and Training Dynamics

Tracking accuracy over 25 epochs evaluates how effectively the network learns and whether it runs the risk of memorizing the training subset.

- **Smooth Optimization:** Training accuracy increases smoothly from 90.9% at Epoch 1 to approximately 93.5% by Epoch 25.
- **Overfitting Control:** The validation accuracy trajectory tracks closely with the training curve throughout the run, peaking near **94.0%** around Epochs 23–25 without divergence. This confirms that Batch Normalization and Dropout layers (0.3/0.2) successfully prevent overfitting.

6. Objective Loss Progression (Figure 7:

Monitoring binary cross-entropy loss provides insight into weight updates and optimization stability over successive epochs.

- **Rapid Optimization:** The solid red line (Training Loss) drops sharply from ~ 0.20 at Epoch 1 to under 0.16 by Epoch 2, before settling smoothly near ~ 0.13 by Epoch 25.
- **Stable Generalization:** Validation loss tracks consistently below the training loss across nearly the entire duration, hitting a minimum floor around **~ 0.12** . The absence of an upward curve confirms that regularization prevented the model from memorizing noise.

Figure 5 Training vs. Validation Accuracy Analysis

Figure 6 Training vs. Validation Loss Analysis

7. Architecture Validation

Evaluating validation accuracy across 8 architectural tuning trials verifies whether the chosen 128-64-32 structure is optimal and stable.

- **Tight Performance Bounds:** Across all 8 trials, validation accuracy remained tightly bounded between **93.60% (Trial 4)** and **94.47% (Trial 0)**.
- **Confirmation of Architecture:** Trial 0 achieved the highest validation accuracy (94.47%), closely supported by Trial 5 (94.44%) and Trial 7 (94.40%). The minimal spread (<0.9%) demonstrates that the 30-feature leakage-free framework is architecturally robust and not overly sensitive to minor hyperparameter shifts.

Figure 7 Hyperparameter Search Analysis

8. Optimizer Sensitivity

A box plot across candidate learning rates (0.0001, 0.001, and 0.01) highlights the impact of gradient step sizes on final accuracy.

- **Optimal Step Size ($\alpha = 0.001$):** A learning rate of 0.001 delivered the highest performance, centering around a median accuracy of **94.40%** and peaking at **94.47%**.
- **Sub-Optimal Scaling:** A rate of 0.0001 caused slow gradient progression and slight underfitting (~93.86% median accuracy), while a larger rate of 0.01 led to oversized steps that overshot local minima, capping accuracy flat at 93.92%. This justifies using $\alpha = 0.001$ for the Adam optimizer.

*Figure 8 Effect of Learning Rate on Model Accuracy**Figure 9 2D PCA Projection Analysis)*

9. Dimensionality Justification

Projecting network packets onto their first two Principal Components reveals the underlying distribution of normal versus attack traffic in linear space.

- **Linear Inseparability:** Benign and malicious instances show extensive overlap in 2D space, demonstrating that linear techniques alone cannot separate the traffic classes.
- **Justification for Deep Non-Linear Layers:** This heavy overlap confirms that simple linear classifiers are insufficient, justifying the deployment of a multi-layer Deep Neural Network with non-linear ReLU activations to capture complex decision boundaries.
- **Validation of Feature Selection:** The dispersion along Principal Component 1 indicates that statistical variance is distributed across multiple features, confirming the benefit of retaining the top 30 Gini-important attributes rather than relying on aggressive linear compression like PCA.

Novelty of This Work

- **Guaranteed Leakage-Free Learning:** Unlike numerous published benchmarks that artificially inflate accuracy by retaining administrative artifacts, this work systematically purges non-causal transaction sequence tags (id) and secondary target labels (attack_cat), forcing the neural network to learn true statistical flow behaviors rather than synthetic memorization shortcuts.
- **Gini-Guided Non-Linear Dimensionality Compression:** Rather than applying blind linear projections like PCA that blur traffic class boundaries, the framework leverages Random Forest Gini-impurity metrics to isolate the Top 30 most discriminative network attributes, removing 33.3% of input dimensions while preserving critical threat signatures.
- **Artifact-Free Skew Mitigation:** The system bypasses synthetic over-sampling techniques like SMOTE—which can inject artificial noise into packet streams—and instead integrates mathematically derived balanced class weights directly into the loss function, maintaining high sensitivity on rare attack vectors.

Engineering & Research Contributions

- **Leakage-Free Validation Baseline:** Establishes a verified, leakage-free benchmark on the UNSW_NB15 dataset, achieving **93.93% accuracy**, **0.9900 ROC-AUC**, and **0.9953 PR-AUC** across **175,341 unseen test instances**, beating the original baseline by **+8.37%** and clean benchmarks by **+6.73%**.
- **Low-Latency, Line-Rate Viability:** Compressing the feature space from 45 down to 30 features drastically reduces matrix multiplication overhead during forward passes, enabling the 3-layer regularized DNN (128→64→32) to inspect continuous traffic without buffer saturation.
- **Alert Fatigue Suppression:** The balanced architecture restricts false alarms to just **5.73% (3,208 benign packets out of 56,000)**, significantly minimizing false-alarm cascades for Security Operations Center (SOC) teams.
- **End-to-End Deployment Artifacts:** The end-to-end pipeline serializes all production-ready components—including trained neural weights (.keras), the fitted normalization scaler (.joblib), and

the top-30 feature schema (.joblib)—enabling immediate zero-retraining inference in live software-defined security perimeters.

VI. Conclusion & Future Work

Modern intrusion detection systems frequently struggle to balance line-rate processing speed with dependable detection accuracy. Many existing deep learning models either buckle under the computational weight of massive, unrefined telemetry or rely on artificial shortcuts created by administrative data leakage. This thesis addressed both bottlenecks directly by designing an optimized, leakage-free Deep Neural Network (DNN) pipeline benchmarked on the UNSW_NB15 dataset.

The framework began by enforcing a strict sanitization protocol that completely stripped non-causal identifiers (id) and secondary target labels (attack_cat), ensuring the network could only learn genuine statistical traffic signatures. To eliminate noisy background telemetry and reduce processing drag, an embedded Random Forest Gini-importance strategy isolated the top 30 most impactful network features. This achieved a 33.3% reduction in input dimensionality, significantly lowering matrix multiplication overhead for line-rate packet inspection. Finally, this lean feature set was processed by a 3-layer regularized Deep Neural Network (128→64→32 neurons) configured with Batch Normalization, Dropout (0.3/0.2), and mathematically balanced class weights to neutralize class skewness without introducing synthetic sampling artifacts.

Evaluated across **175,341 unseen test instances**, the proposed architecture achieved **93.93% Accuracy**, **94.16% Precision**, **93.93% Recall**, and an **F1-Score of 93.99%**. Detailed diagnostic profiling demonstrated near-optimal class separation with an **ROC-AUC of 0.9900** and an Average Precision score of **0.9953** on the Precision-Recall curve, while restricting false alarms to just 5.73%. Crucially, the model outperformed the original UNSW_NB15 baseline by **+8.37%** and surpassed clean benchmarks by **+6.73%**. These findings demonstrate that combining purposeful feature selection with regularized deep architectures delivers a lightweight, robust, and deployment-ready cyber defense solution for enterprise perimeters.

5.2 Future Work

While the framework delivers strong detection rates and minimal overhead, several practical avenues remain for future engineering:

- **Multi-Class Detection:** Expanding beyond binary labels to identify specific attack families like DoS, Exploits, and Fuzzers.
- **Spatial-Temporal Hybrids:** Merging 1D-CNNs with BiLSTM or Transformer layers to trace both localized packet layouts and continuous flow behavior.
- **Explainable AI (XAI):** Embedding tools like SHAP or LIME to provide clear, feature-level context for analysts during active security alerts.
- **Edge Optimization:** Converting models into lightweight formats like ONNX or TFLite for low-power edge hardware and IoT gateways.

References

- [1] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for designing and evaluating network intrusion detection systems," in *Proc. Military Communications and Information Systems Conference (MilCIS)*, IEEE, 2015, pp. 1–6.
- [2] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an Intrusion Detection Dataset: the CICIDS2017 Case Study," in *Proc. 2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021, pp. 7–12.
- [3] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a Standard Feature Set for Network Intrusion Detection System Datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 357–370, 2022.

- [4] A. Khraisat, A. Alazab, L. Portmann, R. Pham, and M. Bhattacharyya, "Survey of intrusion detection systems: techniques, datasets and open challenges," *Journal of Big Data*, vol. 6, no. 1, pp. 1–22, 2019.
- [5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.
- [6] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
- [7] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper based feature extraction for wireless intrusion detection system," *Computers & Security*, vol. 92, p. 101752, 2020.
- [8] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, M. Hasan, B. C. Van Essen, A. A. S. Awwal, and V. K. Asari, "A huge analysis of deep learning methods for intrusion detection systems," *IEEE Access*, vol. 7, pp. 141517–141528, 2019.
- [9] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the case of UNSW-NB15," in *Proc. 2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021, pp. 7–12.
- [10] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, "Towards a lightweight intrusion detection system for IoT networks using Random Forest feature selection," *Computers & Security*, vol. 93, p. 101791, 2020.
- [11] A. Jain, V. Sakalle, and S. Sakalle, "Evaluation of Cloud Intrusion Detection Systems Using Bidirectional LSTMs with Different Feature Selection Algorithms," in *Proc. 2026 IEEE 5th World Conference on Applied Intelligence and Computing (AIC)*, IEEE, 2026, pp. 531–535.
- [12] A. Jain, V. Sakalle, and S. Sakalle, "Dynamic Decision Boundary Calibration via Nelder-Mead Optimization in a Parallel CNN-BiLSTM Spatial-Temporal Hybrid Architecture for Robust DDoS Mitigation," *International Journal of Artificial Intelligence and Machine Learning*, vol. 6, no. 10s, pp. 1874–1896, 2026.
- [13] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection data," *IEEE Access*, vol. 9, pp. 108226–108238, 2021.
- [14] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 1, pp. 1–20, 2020.
- [15] Z. Ahmad, A. S. Khan, C. W. Shiang, and J. Abdullah, "Optimizing feature sets for network intrusion detection using hybrid feature selection," *IEEE Access*, vol. 9, pp. 120412–120425, 2021.
- [16] A. Sharma, S. K. Pasupuleti, and S. V. R. K. Rao, "Performance evaluation of machine learning algorithms on UNSW-NB15 dataset," *Procedia Computer Science*, vol. 171, pp. 1534–1541, 2020.
- [17] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [18] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021.