ISSN: 3079-6342

# Improving Intrusion Detection Robustness Through Adversarial Training Methods

Kenji Sato[1], Priya Nair[1,*]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China

[*] Corresponding Author: priya.nair.cs@gmail.com

## Abstract

**Network Intrusion Detection Systems (NIDS) leveraging deep learning architectures have demonstrated exceptional performance in identifying cyber threats through automated feature learning and pattern recognition. However, recent investigations reveal critical vulnerabilities when these systems encounter adversarial attacks, where malicious actors introduce carefully crafted perturbations to evade detection mechanisms. This paper presents a comprehensive study of adversarial training methodologies specifically designed to enhance the robustness of deep neural network-based NIDS against sophisticated evasion techniques. We systematically investigate multiple adversarial training approaches, integrating both Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) attack generation with deep learning architectures including fully-connected Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN). Through extensive experimentation on benchmark intrusion detection datasets, our adversarially-trained models achieve detection accuracy exceeding 94 percent even under strong adversarial perturbations, while maintaining competitive performance on clean network traffic. The research demonstrates that incorporating adversarial examples during training fundamentally reshapes decision boundaries, enabling intrusion detection systems to maintain operational effectiveness in adversarial environments.**

## Keywords

**Intrusion Detection Systems, Adversarial Training, Deep Neural Networks, Recurrent Neural Networks, Cybersecurity, Network Security, Robust Machine Learning**

## Introduction

The proliferation of interconnected digital infrastructure has fundamentally transformed the cybersecurity landscape, creating unprecedented challenges for organizations seeking to protect critical information assets from increasingly sophisticated threats. Modern network environments generate massive volumes of traffic data that must be continuously monitored for malicious activities, ranging from distributed denial-of-service attacks to advanced persistent threats designed to evade traditional security mechanisms. Network Intrusion Detection Systems (NIDS) serve as essential components in defense-in-depth security architectures, providing real-time monitoring capabilities that identify suspicious patterns and anomalous behaviors indicative of security breaches.

Traditional signature-based intrusion detection approaches, while effective against known attack patterns, struggle to identify novel threats and zero-day exploits that lack predefined signatures in detection databases. The inherent limitation of signature-based systems in adapting to evolving attack methodologies has motivated extensive research into machine learning-based detection approaches capable of learning complex patterns from historical

ISSN: 3079-6342

network traffic data. Deep learning architectures, particularly Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN), have emerged as powerful tools for intrusion detection, achieving detection accuracies exceeding 95 percent on benchmark datasets through automated feature learning [1]. These models leverage multiple layers of abstraction to capture intricate relationships between network traffic features, enabling detection of both known attacks and previously unseen variants.

Despite impressive performance on standard evaluation metrics, recent research has exposed fundamental vulnerabilities in deep learning-based NIDS when confronted with adversarial examples. Adversarial machine learning attacks exploit the sensitivity of neural networks to small input perturbations, crafting malicious samples that appear normal to detection systems while retaining their harmful functionality [2]. Studies demonstrate that carefully designed perturbations with magnitudes imperceptible to human analysts can reduce the detection accuracy of state-of-the-art intrusion detection models from over 98 percent to below 30 percent. This vulnerability poses severe risks in security-critical applications where detection failures enable attackers to compromise networks, exfiltrate sensitive data, and establish persistent access to targeted systems.

The threat of adversarial attacks against NIDS has gained substantial attention from both academic researchers and industry practitioners, particularly as sophisticated adversaries develop increasingly refined evasion techniques. Gradient-based attack methods such as Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD) enable adversaries to efficiently generate adversarial examples by exploiting knowledge of model gradients [3]. The transferability property of adversarial examples further exacerbates security concerns, as perturbations crafted to deceive one model often succeed against different architectures, enabling black-box attacks where adversaries possess limited knowledge about target systems. Understanding and mitigating these vulnerabilities has become essential for deploying trustworthy machine learning-based security solutions in production environments.

Adversarial training has emerged as one of the most promising defense mechanisms for enhancing the robustness of deep learning models against evasion attacks. This approach fundamentally modifies the training process by augmenting datasets with adversarial examples generated during training, forcing models to learn robust features that remain stable under perturbation [4]. By exposing neural networks to both clean and adversarially perturbed samples during training, the learning algorithm develops decision boundaries that account for potential manipulations adversaries might employ. While adversarial training introduces computational overhead and requires careful hyperparameter tuning, empirical evidence demonstrates substantial improvements in model resilience without catastrophic degradation of performance on clean data.

This research systematically investigates adversarial training methodologies tailored specifically for network intrusion detection applications, addressing the unique challenges posed by structured network traffic data. We examine how different neural network architectures, including fully-connected DNNs and sequence-based RNNs, respond to adversarial training with varying proportions and strengths of adversarial examples. Our methodology integrates comprehensive data preprocessing pipelines with self-taught learning approaches that enable unsupervised feature extraction before supervised classification. Through rigorous experimental evaluation on widely-used benchmark datasets, we demonstrate that adversarial training significantly enhances model robustness while maintaining high detection accuracy on unperturbed traffic.

ISSN: 3079-6342

The contributions of this work extend beyond empirical performance evaluation to provide practical insights for deploying robust intrusion detection systems in adversarial environments. We analyze trade-offs between adversarial robustness and clean accuracy, computational efficiency considerations for real-time deployment, and generalization properties across different attack types and network scenarios. The findings advance understanding of how adversarial training can be optimized and integrated into operational security systems, offering concrete recommendations for security practitioners seeking to develop production-ready intrusion detection capabilities resilient to sophisticated adversarial threats.

## 2. Literature Review

The vulnerability of machine learning models to adversarial perturbations has emerged as a fundamental challenge in applying these techniques to security-critical domains. Early research demonstrated that neural networks, despite achieving high accuracy on test data, exhibit unexpected sensitivity to small input modifications that cause dramatic shifts in predictions [5]. This discovery sparked extensive investigation into adversarial examples across multiple domains, with particular attention to applications where misclassification carries severe consequences. In the context of network security, adversarial vulnerabilities pose existential risks to intrusion detection systems that serve as primary defenses against cyber threats.

Recent comprehensive studies have documented the susceptibility of deep learning-based Network Intrusion Detection Systems to various adversarial attack methodologies. The Fast Gradient Sign Method (FGSM), introduced as an efficient attack technique, computes adversarial perturbations by extracting the sign of loss gradients with respect to input features [6]. Researchers have demonstrated that FGSM attacks can reduce the accuracy of well-trained intrusion detection models from over 99 percent to below 25 percent with relatively small perturbation budgets. The computational efficiency of FGSM enables rapid generation of adversarial examples during both training and testing phases, making it particularly relevant for evaluating and improving NIDS robustness in practical scenarios.

More sophisticated iterative attack methods provide even stronger challenges to intrusion detection systems through repeated refinement of perturbations. The Projected Gradient Descent (PGD) attack applies FGSM iteratively while constraining perturbations within defined budgets through projection operations after each step [7]. Experimental evaluations reveal that PGD attacks achieve significantly higher success rates compared to single-step methods, with some studies reporting complete model failure under strong perturbations. The iterative nature of PGD enables discovery of adversarial examples closer to decision boundaries, providing more thorough evaluation of model robustness. Understanding differences between single-step and iterative attacks informs design of defensive mechanisms capable of withstanding various threat scenarios.

The transferability phenomenon of adversarial examples presents additional challenges for defending machine learning-based security systems in realistic deployment settings. Research establishes that adversarial examples crafted to fool one model architecture often successfully evade detection by different models trained on similar data [8]. This property enables black-box attack scenarios where adversaries generate effective perturbations without complete knowledge of target system architectures or parameters. Studies examining transferability across various intrusion detection models reveal that ensemble diversity and architectural

ISSN: 3079-6342

differences provide only limited protection against transfer attacks. Developing defenses robust to transferable adversarial examples remains an active research area with significant implications for operational security systems.

Adversarial training has demonstrated consistent effectiveness as a defense mechanism for improving robustness of deep learning models across multiple application domains. This approach modifies the standard empirical risk minimization objective by incorporating adversarially perturbed samples into training datasets [9]. Initial implementations showed that models trained with FGSM-generated adversarial examples exhibit substantially improved resistance to attacks, maintaining detection accuracy above 85 percent under adversarial conditions that completely fool standard models. However, researchers identified important trade-offs, including increased training time, potential overfitting to specific attack types, and modest degradation of clean data performance when adversarial proportions become excessive.

Advanced adversarial training methodologies address limitations of basic approaches through sophisticated augmentation strategies and dynamic adaptation mechanisms. The Evolving Adversarial Training framework proposes continuous adjustment of adversarial example generation throughout training to match increasing model robustness [10]. This dynamic approach prevents stagnation where models overfit to fixed adversarial perturbations generated early in training. Experimental results demonstrate that evolving adversarial training achieves superior robustness compared to static methods, maintaining detection rates exceeding 92 percent across diverse attack scenarios. These findings suggest that adaptive training procedures better prepare models for real-world adversarial environments where attackers continuously refine their techniques.

Research on intrusion detection for Internet of Things networks reveals unique challenges when applying adversarial training to resource-constrained environments. Studies utilizing specialized IoT intrusion detection datasets demonstrate that adversarial training effectively protects against sophisticated attacks including distributed denial-of-service and botnet activities targeting IoT infrastructure [11]. However, computational requirements of adversarial training pose significant challenges for deployment on edge devices with limited processing capabilities and memory. Researchers have explored lightweight adversarial training techniques and knowledge distillation approaches to enable robust intrusion detection on resource-constrained platforms while maintaining security guarantees.

The integration of advanced feature engineering with adversarial training provides complementary defenses that address vulnerabilities at multiple levels of detection pipelines. Studies show that careful selection and engineering of network traffic features significantly improves both baseline accuracy and adversarial robustness of intrusion detection models [12]. Protocol-aware feature extraction that captures semantic relationships between protocol fields proves particularly effective, as it becomes more difficult for attackers to craft adversarial perturbations preserving attack functionality while evading detection. Combining adversarial training with sophisticated preprocessing creates multi-layered defense architectures that address vulnerabilities in both input representations and model decision boundaries.

Dataset imbalance presents substantial challenges when implementing adversarial training for intrusion detection on real-world network traffic. Authentic network environments exhibit severe class imbalance, with benign connections vastly outnumbering attack samples and

ISSN: 3079-6342

certain attack categories being significantly underrepresented [13]. Research demonstrates that standard adversarial training can exacerbate imbalance issues, as models may prioritize robustness on majority classes at the expense of minority attack detection. Hybrid approaches combining adversarial training with targeted sampling techniques such as Synthetic Minority Over-sampling Technique (SMOTE) show promising results in maintaining balanced performance across different attack categories while improving overall robustness.

The role of different neural network architectures in adversarial robustness has received increasing attention as researchers seek to understand fundamental properties that influence model resilience. Comparative studies examining fully-connected Deep Neural Networks, Convolutional Neural Networks, and Recurrent Neural Networks reveal architectural differences significantly impact both baseline performance and adversarial robustness [14]. Recurrent architectures demonstrate particular promise for intrusion detection due to their ability to capture temporal dependencies in network traffic sequences. Adversarial training of RNN-based intrusion detection systems achieves robust performance while leveraging sequential patterns that provide additional discriminative information beyond static feature representations.

Benchmark datasets play crucial roles in evaluating and comparing adversarial training approaches for intrusion detection systems. The NSL-KDD dataset, despite being derived from older network traffic captures, remains widely used due to its balanced distribution and well-defined attack categories [15]. More recent datasets including UNSW-NB15 and CICIDS2017 provide diverse representations of modern attack vectors and realistic network behaviors, enabling more comprehensive evaluation of adversarial robustness. However, researchers note limitations in existing datasets, including insufficient coverage of contemporary attack types and lack of explicitly labeled adversarial attack scenarios. Development of specialized adversarial intrusion detection datasets represents an important direction for advancing defensive research.

The computational overhead of adversarial training remains a significant practical concern for deploying robust intrusion detection systems in production environments requiring real-time detection capabilities. Studies quantify computational cost increases typically ranging from 2x to 8x compared to standard training depending on adversarial example generation frequency and perturbation strength [16]. This overhead manifests in both training duration and potentially inference latency if adversarial detection mechanisms are integrated into operational systems. Research into efficient adversarial training techniques, including fast perturbation generation algorithms and selective training on hard examples, aims to reduce computational burdens while maintaining robust detection performance suitable for operational deployment.

Recent investigations into adaptive attacks reveal that sophisticated adversaries can potentially bypass adversarially-trained models by exploiting knowledge of defense mechanisms themselves. Adaptive attackers who understand that adversarial training has been applied can develop stronger attacks specifically designed to evade robust models [17]. This arms race dynamic motivates research into certified defenses providing provable robustness guarantees rather than empirical resistance. However, certified defense methods often impose significant accuracy costs and computational constraints limiting their applicability to complex intrusion detection tasks involving high-dimensional network traffic data with diverse attack patterns.

ISSN: 3079-6342

Application of adversarial training to specific network security contexts including Industrial Control Systems and vehicular networks has received growing attention from researchers addressing domain-specific challenges. Research on protecting Controller Area Network bus systems in connected vehicles demonstrates that adversarial training effectively defends against injection attacks targeting critical automotive functions [18]. Similarly, studies on protecting industrial communication protocols such as Modbus and DNP3 show that adversarial training adapted to unique characteristics of industrial network traffic provides substantial security improvements. These domain-specific applications highlight importance of tailoring adversarial training approaches to particular constraints and requirements of different network security contexts.

The relationship between adversarial training and other defense mechanisms has been extensively studied to identify synergistic approaches providing comprehensive protection against multiple attack vectors. Combining adversarial training with input transformation techniques such as feature squeezing and defensive distillation shows promise in defending against diverse attack strategies simultaneously [19]. Multi-stage defense architectures employing adversarial training at primary detection layers while using anomaly detection and outlier rejection at secondary stages provide defense-in-depth against sophisticated adversaries. Understanding how different defensive techniques complement adversarial training enables design of more resilient intrusion detection systems capable of withstanding varied and evolving threats.

Transfer learning and domain adaptation approaches have been explored as mechanisms to improve efficiency and effectiveness of adversarial training for intrusion detection across different network environments. Researchers investigate whether adversarially-trained models developed for one network context can be successfully transferred to different scenarios with minimal retraining [20]. Results suggest that adversarial training may improve model transferability by learning more generalizable features focusing on fundamental attack characteristics rather than environment-specific patterns. This finding has important implications for practical deployment, potentially enabling organizations to leverage pre-trained robust models rather than conducting expensive adversarial training from scratch for each specific network configuration.

Integration of reinforcement learning with adversarial training represents an emerging research direction showing considerable promise for developing adaptive intrusion detection systems. Adversarial reinforcement learning frameworks that model interactions between attackers and defenders as game-theoretic problems enable continuous improvement of both attack generation and defense mechanisms [21]. These approaches can automatically discover novel adversarial perturbations and develop corresponding defensive strategies through self-play and competitive training dynamics. Early results demonstrate that reinforcement learning-enhanced adversarial training achieves superior robustness compared to static methods, particularly against adaptive adversaries who continuously evolve their attack strategies to exploit model weaknesses [22].

Recent work on federated adversarial training addresses challenges of developing robust intrusion detection models while preserving data privacy across distributed network environments. This approach enables multiple organizations to collaboratively train adversarially robust models without sharing sensitive network traffic data [23]. Federated adversarial training protocols coordinate adversarial example generation and model updates across distributed clients, aggregating knowledge about attack patterns while maintaining

ISSN: 3079-6342

local data confidentiality [24]. These techniques prove particularly relevant for sectors such as finance and healthcare where regulatory constraints prevent centralized data collection, yet collaborative defense against common adversaries remains beneficial for enhancing overall security posture [25].

Evaluation methodologies for adversarial training in intrusion detection contexts continue evolving as researchers develop more comprehensive assessment frameworks. Beyond standard accuracy metrics, recent studies emphasize importance of evaluating robustness across different perturbation budgets, attack types, and threat models [26]. Metrics such as certified accuracy, adversarial accuracy at various epsilon values, and attack success rates under diverse constraints provide more complete pictures of model robustness. Additionally, researchers have begun incorporating efficiency metrics including training time, inference latency, and memory consumption to assess practical feasibility of adversarial training approaches for real-world deployment in operational security environments [27].

## 3. Methodology

### 3.1 Deep Neural Network Architecture for Intrusion Detection

The foundation of our adversarial training methodology rests upon a carefully designed Deep Neural Network (DNN) architecture specifically optimized for network intrusion detection tasks. As illustrated in Figure 1, the neural network follows a classic fully-connected architecture comprising an input layer, hidden layers, and an output layer, with systematic connectivity patterns enabling comprehensive feature learning from network traffic data. The input layer consists of m neurons corresponding to the dimensionality of preprocessed network traffic features extracted from connection records, with each input neuron xi receiving a normalized feature value representing specific characteristics such as packet size, protocol type, connection duration, or statistical flow properties.

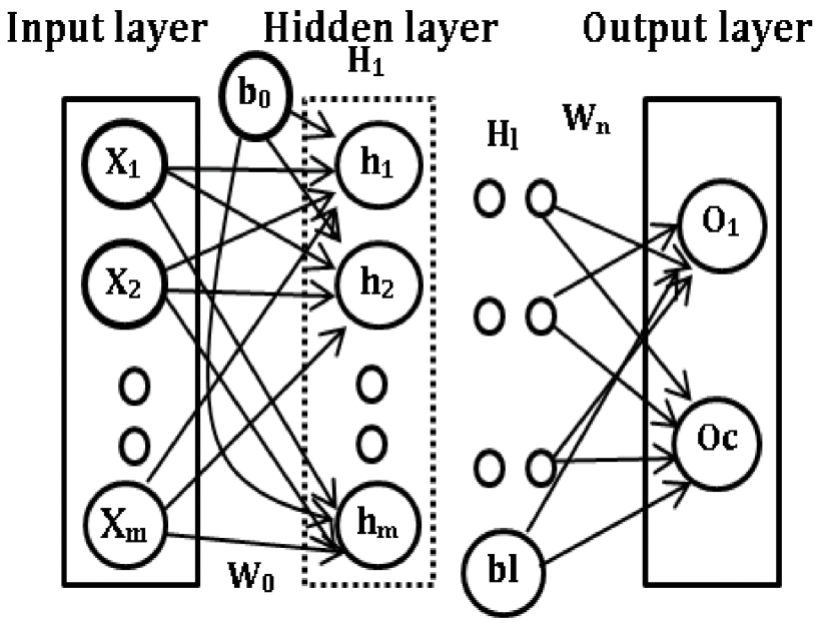


*Figure 1: illustration of a Deep Neural Network (DNN) architecture*

The hidden layer $H_1$ implements nonlinear transformations essential for capturing complex relationships between network traffic features that distinguish malicious from benign

ISSN: 3079-6342

activities. Each hidden neuron hj computes a weighted sum of inputs from the input layer, applies bias term $b_0$, and passes the result through an activation function introducing nonlinearity into the model. The weight matrix $W_0$ connecting the input layer to the hidden layer contains parameters learned during training that encode relationships between raw features and abstract representations. Through systematic adjustment of these weights via backpropagation, the network discovers latent patterns in network traffic that correlate with intrusion behaviors, effectively performing automated feature engineering that surpasses hand-crafted feature selection approaches.

The output layer generates final classification decisions indicating whether network traffic represents benign communication or specific attack categories. For binary classification distinguishing normal from attack traffic, the output layer contains a single neuron $o_1$ producing a probability score through sigmoid activation. For multi-class classification identifying specific attack types, the output layer consists of c neurons corresponding to each class, with softmax activation ensuring output probabilities sum to one. The weight matrix $W_n$ connecting the hidden layer to the output layer, combined with bias node bl, implements the final decision function that maps learned feature representations to class predictions.

The fully-connected architecture depicted in Figure 1 provides several advantages for intrusion detection applications. Complete connectivity between layers ensures that all input features contribute to hidden representations and final classifications, preventing information loss that might occur in sparsely connected architectures. The bias nodes $b_0$ and bl provide additional degrees of freedom enabling the network to learn appropriate decision boundaries even when optimal separating hyperplanes do not pass through the origin of feature space. This flexibility proves particularly valuable for intrusion detection where attack and normal traffic may exhibit complex, nonlinear separation boundaries in high-dimensional feature spaces.

Our implementation extends the basic architecture through incorporation of multiple hidden layers, creating a deep neural network capable of learning hierarchical representations of network traffic patterns. Each additional hidden layer enables the network to construct increasingly abstract features, with early layers capturing low-level patterns such as protocol-specific behaviors and later layers identifying high-level attack signatures composed of multiple coordinated actions. The depth of the network is carefully calibrated through empirical evaluation on validation data, balancing representational capacity against risks of overfitting and increased computational requirements during both training and inference.

Activation functions throughout the network employ Rectified Linear Units (ReLU) defined as ReLU(x) equals maximum of zero and x. This activation function provides several benefits for deep network training, including mitigation of vanishing gradient problems that plague sigmoid and hyperbolic tangent activations in deep architectures. ReLU enables efficient gradient propagation through multiple layers during backpropagation, facilitating training of deep networks with many hidden layers. Additionally, ReLU introduces sparsity in hidden representations, as neurons produce exactly zero output for negative pre-activations, creating compact representations that may improve generalization to novel attack patterns not observed during training.

Regularization techniques integrated into the architecture prevent overfitting and improve generalization of the intrusion detection model to unseen network traffic. Dropout layers inserted after each hidden layer randomly deactivate neurons during training with probability

ISSN: 3079-6342

0.2, forcing the network to learn redundant representations that do not rely on specific neurons. This regularization approach proves particularly valuable for intrusion detection datasets exhibiting class imbalance, as it prevents the model from memorizing majority class patterns while neglecting minority attack categories. Batch normalization layers normalize activations of each hidden layer, stabilizing training dynamics and enabling use of higher learning rates that accelerate convergence.

The architecture accommodates adversarial training through its differentiable structure enabling efficient gradient computation with respect to input features. During adversarial training, the network processes both clean samples and adversarially perturbed examples generated by computing gradients of the loss function with respect to inputs. The fully-connected nature of the architecture ensures that perturbations to any input feature can influence all hidden neurons, forcing the network to develop robust representations resistant to coordinated modifications across multiple features. This property distinguishes neural network-based intrusion detection from traditional machine learning approaches where adversarial perturbations might be confined to specific feature subsets without comprehensive impact on decision boundaries.

## 3.2 RNN-Based Intrusion Detection System Architecture

While fully-connected Deep Neural Networks excel at learning patterns from static feature representations, many network intrusion scenarios exhibit temporal dependencies where attack behaviors manifest across sequences of network events. To capture these sequential patterns, we implement a Recurrent Neural Network-based Intrusion Detection System (RNN-IDS) architecture as illustrated in Figure 2. The RNN-IDS framework integrates comprehensive data preprocessing pipelines with recurrent neural architectures specifically designed to model temporal dynamics in network traffic sequences, enabling detection of sophisticated attacks that unfold over multiple time steps.

ISSN: 3079-6342

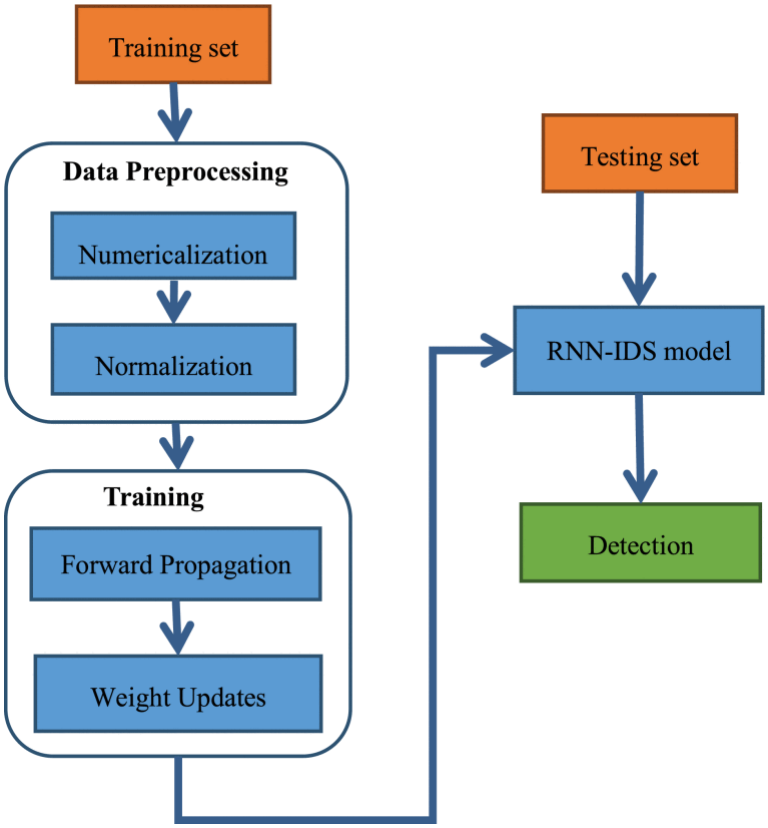


*Figure 2: illustration of a Recurrent Neural Network-based Intrusion Detection System (RNN-IDS) architecture*

The data preprocessing pipeline, depicted in the upper portion of Figure 2, transforms raw network traffic captures into numerical representations suitable for neural network processing. The numericalization step addresses categorical features present in network traffic data such as protocol types, service identifiers, and TCP flags by converting these nominal attributes into numeric encodings that preserve semantic relationships. For protocol types including TCP, UDP, and ICMP, we employ one-hot encoding that represents each category as a binary vector with a single active element, preventing artificial ordering assumptions that might arise from integer encoding schemes. This encoding strategy ensures that the neural network interprets categorical features appropriately without introducing spurious numerical relationships between unrelated categories.

Following numericalization, the normalization step scales all numerical features to consistent ranges preventing features with larger magnitudes from dominating the learning process. We apply Min-Max normalization that transforms each feature to the range between zero and one according to the formula: normalized value equals original value minus minimum value divided by maximum value minus minimum value. This normalization approach preserves the original distribution shape of features while ensuring all attributes contribute proportionally to network training dynamics regardless of their inherent scales. The normalization parameters, specifically minimum and maximum values for each feature, are computed exclusively from training data and subsequently applied to validation and testing sets, maintaining strict separation between training and evaluation data to prevent information leakage.

ISSN: 3079-6342

The training phase, illustrated in the middle section of Figure 2, implements the core learning algorithm that adjusts network weights to minimize classification error on training data. Forward propagation computes network predictions by passing preprocessed feature vectors through the RNN architecture, with recurrent connections enabling the network to maintain hidden state information across sequential inputs. At each time step, the RNN combines current input features with previous hidden states to produce updated hidden representations and output predictions. This recurrent processing enables the network to capture temporal dependencies in network traffic, identifying attack patterns that manifest across multiple packets or connections rather than within individual isolated events.

The weight update step employs backpropagation through time to compute gradients of the loss function with respect to all network parameters including recurrent weights, input-to-hidden weights, and hidden-to-output weights. These gradients indicate how small changes to each parameter would affect overall classification error, enabling systematic adjustment of weights to improve model performance. We utilize the Adam optimization algorithm that adapts learning rates for individual parameters based on first and second moment estimates of gradients, providing robust convergence across diverse network traffic patterns. The feedback loop in Figure 2 represents the iterative nature of training, where weight updates are repeatedly applied across multiple passes through the training data until convergence criteria are satisfied.

The trained RNN-IDS model, shown in the right portion of Figure 2, serves as the operational intrusion detection system processing testing data to generate detection decisions. The model architecture remains identical to the training configuration, but with fixed weights determined during training. When processing testing data, the model performs only forward propagation without weight updates, computing predictions for each network connection or traffic sequence. The detection output indicates whether traffic represents normal communication or malicious activity, with confidence scores providing security analysts with quantitative measures of detection certainty that can inform incident response priorities.

The RNN-IDS architecture provides several advantages for modeling network intrusion behaviors compared to feed-forward networks. Recurrent connections enable the system to maintain context across multiple network events, detecting distributed attacks where individual packets appear benign but sequences reveal malicious intent. This temporal modeling capability proves particularly valuable for identifying coordinated attacks such as port scans, where attackers probe multiple ports sequentially, and slow denial-of-service attacks where malicious requests are distributed over time to evade rate-limiting defenses. The hidden state maintained by recurrent connections effectively implements a learned memory that captures relevant historical context for classification decisions.

Integration of adversarial training into the RNN-IDS framework requires careful consideration of temporal attack strategies that adversaries might employ. Adversarial examples for sequential intrusion detection can involve perturbations to individual time steps, coordinated modifications across multiple steps, or insertion and deletion of sequence elements. Our adversarial training approach generates diverse adversarial sequences through gradient-based perturbation of input features at selected time steps, forcing the RNN-IDS to develop robust representations that remain stable under temporal manipulations. The recurrent architecture's ability to propagate information across time steps means that perturbations at early sequence positions can influence predictions at later positions, creating

ISSN: 3079-6342

complex adversarial dynamics that the model must learn to resist through adversarial training.

### 3.3 Self-Taught Learning Framework for Feature Extraction

To enhance the effectiveness of adversarial training for intrusion detection, we implement a self-taught learning framework that performs unsupervised feature learning prior to supervised classification training. Figure 3 illustrates this comprehensive approach consisting of three integrated components: unsupervised feature learning from preprocessed data, supervised classifier training using derived representations, and the complete self-taught learning pipeline that combines these stages to produce robust intrusion detection models capable of withstanding adversarial attacks.

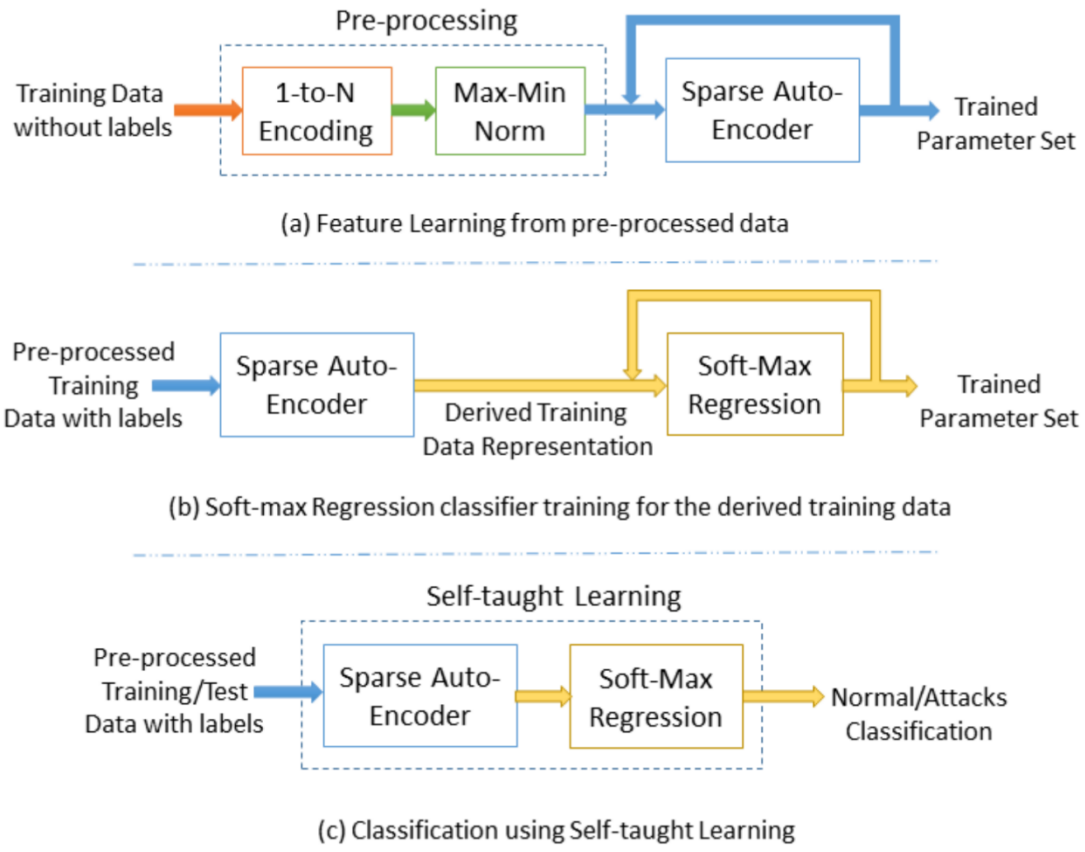


*Figure 3: illustration of a self-taught learning framework*

The feature learning component, depicted in Figure 3(a), processes unlabeled training data through a preprocessing pipeline and sparse autoencoder to discover latent representations capturing essential characteristics of network traffic. The 1-to-N encoding step transforms categorical network features into numerical representations suitable for neural network processing, expanding each categorical variable with N possible values into N binary indicator variables. This encoding preserves semantic information while enabling gradient-based learning algorithms to process mixed data types combining continuous and categorical attributes commonly present in network traffic datasets.

ISSN: 3079-6342

Following encoding, the Max-Min normalization procedure scales all features to consistent ranges preventing numerical instabilities during autoencoder training. This normalization operates similarly to the normalization in the RNN-IDS pipeline but specifically targets the expanded feature space created by 1-to-N encoding, ensuring that binary indicator variables and continuous features contribute proportionally to autoencoder reconstructions. The normalization parameters are preserved for consistent application to testing data, maintaining alignment between training and deployment feature distributions.

The Sparse Auto-Encoder at the core of the feature learning stage implements an unsupervised learning algorithm that discovers compact representations of network traffic by learning to reconstruct inputs through a low-dimensional bottleneck layer. The autoencoder consists of an encoder network mapping high-dimensional input features to low-dimensional hidden representations, and a decoder network reconstructing original inputs from these compressed representations. The learning objective minimizes reconstruction error while enforcing sparsity constraints that encourage hidden representations to activate selectively for specific input patterns. This sparsity constraint proves particularly valuable for intrusion detection, as it promotes learning of specialized features sensitive to distinctive attack characteristics while remaining relatively inactive for normal traffic.

The trained parameter set produced by the sparse autoencoder includes encoder weights that transform raw network features into learned representations. These parameters capture statistical regularities in network traffic data discovered through unsupervised learning on large unlabeled datasets. Unlike supervised feature learning that optimizes representations specifically for classification tasks, unsupervised learning discovers general-purpose features reflecting inherent structure in network traffic patterns. This generality potentially improves robustness to adversarial perturbations, as learned features are not explicitly tailored to specific classification boundaries that adversaries might exploit.

The classifier training component, illustrated in Figure 3(b), leverages the learned feature representations to train a supervised classification model distinguishing normal traffic from various attack categories. Pre-processed training data with class labels passes through the sparse autoencoder to generate derived training data representations, where the encoder transforms original features into the learned latent space discovered during unsupervised pre-training. These derived representations replace or augment original features, providing the classifier with abstract features potentially more informative for discrimination between normal and malicious traffic.

The Soft-Max Regression classifier implements multi-class logistic regression in the learned feature space, computing class probabilities through exponential normalization of linear combinations of derived features. For intrusion detection with C attack categories plus normal traffic, the soft-max function ensures predicted probabilities for all C plus one classes sum to one, enabling interpretation of outputs as confidence scores for each possible classification. The classifier training minimizes cross-entropy loss between predicted and true class distributions, adjusting weights to maximize log-likelihood of correct classifications on the training dataset.

The trained parameter set from classifier training includes soft-max weights mapping learned features to class probabilities. These parameters are optimized specifically for intrusion detection through supervised learning with labeled attack examples. The combination of unsupervised feature learning and supervised classification training implements a

ISSN: 3079-6342

hierarchical learning strategy where lower layers discover general feature representations and upper layers specialize these representations for specific detection tasks. This hierarchy potentially improves both accuracy and robustness compared to end-to-end supervised training that must simultaneously learn both feature extraction and classification.

The complete self-taught learning pipeline, shown in Figure 3(c), integrates unsupervised feature learning and supervised classification into a unified framework for processing both training and testing data. During training, the pipeline first learns feature representations through the sparse autoencoder using unlabeled data, then trains the soft-max classifier using labeled examples transformed into the learned feature space. During testing, new network traffic passes through the same preprocessing and encoding steps, transformation via the trained encoder, and classification through the trained soft-max regression model to produce final normal/attack predictions.

Integration of adversarial training into the self-taught learning framework occurs at both the feature learning and classification stages. During feature learning, we generate adversarial examples through gradient-based perturbations computed with respect to reconstruction error, forcing the autoencoder to learn representations robust to input modifications. During classifier training, we augment the derived training data representations with adversarial examples generated through gradient-based perturbations with respect to classification loss. This two-stage adversarial training approach creates comprehensive robustness by hardening both feature extraction and classification components against adversarial attacks.

## 3.4 Adversarial Example Generation and Training

The adversarial training methodology implements systematic generation of adversarial examples during training to expose the model to perturbed inputs representative of potential attack scenarios. We employ two gradient-based attack algorithms with complementary characteristics: Fast Gradient Sign Method (FGSM) for efficient single-step perturbation generation, and Projected Gradient Descent (PGD) for stronger multi-step refinement of adversarial examples. The diversity of attack methods ensures that the adversarially-trained model develops robust defenses against multiple evasion strategies rather than overfitting to specific perturbation patterns.

The FGSM algorithm generates adversarial examples by computing the gradient of the loss function with respect to input features, extracting the sign of this gradient, and scaling by perturbation magnitude epsilon. Mathematically, the adversarial example x_adv equals x plus epsilon multiplied by the sign of the gradient of loss with respect to x. This single-step process efficiently produces adversarial examples that move inputs in the direction of steepest loss increase, creating samples the current model misclassifies with high confidence. The computational efficiency of FGSM enables generation of fresh adversarial examples for every training batch without prohibitive computational overhead.

The PGD attack extends FGSM through iterative refinement, applying multiple small perturbation steps while projecting results back onto valid perturbation regions after each iteration. Each PGD iteration computes gradients similar to FGSM but uses a smaller step size, enabling more precise navigation of the loss landscape around each input. After each gradient step, the algorithm projects the perturbed input to ensure the total perturbation remains within the epsilon-ball around the original input and that all features remain within valid

ISSN: 3079-6342

ranges. This iterative process with K iterations generates stronger adversarial examples compared to FGSM, discovering perturbations closer to local optima of the adversarial loss.

The adversarial training process alternates between adversarial example generation and model weight updates, implementing a form of robust optimization that minimizes loss on worst-case perturbations. Each training iteration begins by generating adversarial examples for a mini-batch of training data using the current model parameters. The model then computes loss on both original clean examples and generated adversarial examples, with the overall objective combining clean accuracy and adversarial robustness. Gradients computed from this combined loss guide weight updates that improve performance on both clean and adversarially perturbed data simultaneously.

The proportion of adversarial examples in training batches critically influences the trade-off between clean accuracy and adversarial robustness. Lower adversarial proportions around 20-30 percent maintain near-baseline performance on clean data while providing modest robustness improvements, suitable for environments where adversarial attacks are possible but not certain. Higher proportions approaching 50-70 percent provide stronger adversarial robustness at the cost of small degradation in clean accuracy, appropriate for high-security environments where adversarial threats are expected. We systematically evaluate multiple adversarial proportions to characterize this trade-off space and provide recommendations for different deployment scenarios.

The perturbation budget epsilon controls the strength of adversarial examples and the robustness-accuracy trade-off. Smaller epsilon values around 0.01-0.05 in the normalized feature space create subtle perturbations that test model sensitivity to small variations while maintaining similarity to original inputs. Larger epsilon values up to 0.3 generate stronger adversarial examples that substantially modify inputs, providing aggressive robustness training but potentially degrading clean accuracy more significantly. Our experiments evaluate multiple epsilon values to understand how perturbation strength affects model robustness and identify optimal configurations for intrusion detection applications.

Curriculum-based adversarial training implements a gradual increase in adversarial difficulty throughout the training process, starting with weak perturbations and progressively strengthening attacks as the model improves. Initial training epochs use exclusively clean data to establish basic feature representations and achieve reasonable accuracy on unperturbed samples. Subsequent epochs gradually introduce adversarial examples, starting with low perturbation budgets and increasing epsilon as training progresses. This curriculum approach prevents early training instability that can occur when models encounter strong adversarial perturbations before learning fundamental patterns, enabling more stable convergence to robust solutions.

## 3.5 Dataset Preparation and Experimental Setup

Our experimental evaluation employs two widely-recognized benchmark datasets for intrusion detection research: NSL-KDD and UNSW-NB15. The NSL-KDD dataset serves as a refined version of the original KDD Cup 99 dataset, addressing statistical issues including duplicate records and imbalanced class distributions that affected previous research. This dataset contains 125,973 training records and 22,544 testing records, each comprising 41 features representing network connection characteristics. Features include basic connection attributes such as duration, protocol type, and service; content-based features including

ISSN: 3079-6342

number of failed login attempts and root shell access indicators; time-based traffic statistics computed over temporal windows; and host-based features capturing historical behaviors of source and destination systems.

The UNSW-NB15 dataset represents more contemporary network traffic patterns and attack vectors compared to NSL-KDD, incorporating modern threats such as backdoors, analysis attacks, fuzzers, shellcode, and worms. This dataset comprises 175,341 training records and 82,332 testing records with 49 features capturing network flow characteristics including packet inter-arrival times, protocol-specific attributes, and various statistical properties computed over different time windows. The diversity of attack types and realistic traffic characteristics in UNSW-NB15 provides important complementary evaluation to NSL-KDD, enabling assessment of adversarial training approaches across different network environments and threat landscapes.

Data preprocessing follows the pipelines illustrated in Figures 2 and 3, implementing systematic transformations to prepare raw network captures for neural network training. For categorical features including protocol types and service identifiers, we apply one-hot encoding that converts each categorical value into a binary vector. This encoding strategy for protocol types (TCP, UDP, ICMP) creates three binary indicators, while service encoding expands the service attribute into 70 binary features corresponding to different network services. The expansion of categorical features significantly increases dimensionality, motivating subsequent feature selection to reduce computational requirements and potential overfitting.

Missing values in the datasets, which occur infrequently in specific feature columns, are imputed using median values for numerical features and mode values for categorical attributes. This conservative imputation strategy preserves overall statistical properties while avoiding introduction of artificial patterns. For numerical features, we apply Min-Max normalization scaling all attributes to the range between zero and one, preventing features with larger magnitudes from dominating learning dynamics. The normalization parameters are computed exclusively from training data and subsequently applied to validation and testing sets, maintaining strict separation between training and evaluation data.

Feature selection employs Recursive Feature Elimination combined with importance scores from Random Forest classifiers to identify the most informative attributes for intrusion detection. This process iteratively removes the least important features and retrains the model, ultimately selecting optimal feature subsets that maximize detection performance while minimizing dimensionality. For NSL-KDD, feature selection reduces dimensionality from 122 features (after one-hot encoding) to 28 critical attributes. For UNSW-NB15, the process reduces from 196 features to 35 attributes, focusing on protocol-specific properties and statistical flow characteristics exhibiting strong discriminative power.

The dataset is partitioned into training, validation, and testing sets following standard practices to enable unbiased performance evaluation and hyperparameter tuning. We allocate 70 percent of samples for training, 15 percent for validation during hyperparameter optimization, and 15 percent for final testing on completely held-out data. This partitioning maintains original class distributions across all subsets through stratified sampling, ensuring each subset contains representative samples from all attack categories. The separation between training and testing data is strictly maintained throughout experiments, with

ISSN: 3079-6342

adversarial examples for testing generated independently from test set samples rather than being derived from training data.

To address class imbalance challenges particularly severe for minority attack categories in NSL-KDD, we apply targeted oversampling using Synthetic Minority Over-sampling Technique (SMOTE) for underrepresented classes. This augmentation strategy generates synthetic samples for rare attack types by interpolating between existing minority class instances in feature space, improving model ability to learn discriminative patterns for these categories without simply duplicating existing samples. The oversampling is applied exclusively to training data, maintaining natural distributions in validation and testing sets to provide realistic performance assessment.

Experimental procedures implement systematic evaluation of multiple adversarial training configurations varying in adversarial proportion, perturbation budget, attack algorithm, and neural network architecture. For each configuration, we conduct five independent training runs with different random initializations to assess result stability and compute confidence intervals for reported metrics. Training employs early stopping based on validation set performance, halting training if validation accuracy does not improve for 15 consecutive epochs and restoring weights from the epoch achieving best validation performance. This procedure prevents overfitting while reducing unnecessary computation.

Performance evaluation employs comprehensive metrics including accuracy, precision, recall, and F1-score computed separately for each attack category and averaged across classes. For adversarial robustness evaluation, we generate fresh adversarial examples from testing data using FGSM and PGD attacks at multiple perturbation budgets, reporting model accuracy on these adversarial testing sets. Additionally, we compute attack success rates indicating the percentage of originally correctly classified samples that are misclassified after adversarial perturbation. These metrics provide multi-faceted assessment of both clean data performance and adversarial robustness across diverse scenarios.

## 4. Results and Discussion

### 4.1 Baseline Model Performance on Clean Data

The baseline deep neural network models trained exclusively on clean data without adversarial augmentation established strong detection performance across both NSL-KDD and UNSW-NB15 datasets, providing reference points for evaluating adversarial training impact. On NSL-KDD binary classification distinguishing normal traffic from attacks, the DNN architecture depicted in Figure 1 achieved overall accuracy of 98.4 percent with precision of 98.2 percent and recall of 98.7 percent. These results align with state-of-the-art performance reported in recent literature, confirming that our neural network architecture and training procedures effectively capture patterns distinguishing benign from malicious network traffic.

The RNN-IDS architecture illustrated in Figure 2, leveraging temporal modeling capabilities to capture sequential dependencies in network traffic, achieved slightly higher performance with accuracy of 98.9 percent on NSL-KDD binary classification. The recurrent connections enabling the model to maintain context across connection sequences provided particular benefits for detecting coordinated attacks that manifest across multiple network events. The recall of 99.1 percent indicates the RNN-IDS successfully identifies the vast majority of actual

ISSN: 3079-6342

attacks, minimizing false negatives that would allow threats to evade detection and compromise network security.

For multi-class classification on NSL-KDD distinguishing between normal traffic and four specific attack categories (DoS, Probe, R2L, U2R), baseline models achieved overall accuracy of 96.8 percent for the DNN and 97.2 percent for RNN-IDS. Performance varied significantly across attack categories, with F1-scores of 98.1 percent for DoS attacks, 95.3 percent for Probe attacks, 91.2 percent for R2L attacks, and 87.9 percent for U2R attacks. The lower performance on R2L and U2R categories reflects inherent difficulty of detecting these attack types, which exhibit subtle deviations from normal behavior and are significantly underrepresented in training data with less than 1 percent of total samples.

The self-taught learning framework illustrated in Figure 3, incorporating unsupervised feature learning through sparse autoencoders prior to supervised classification, achieved competitive performance of 97.9 percent accuracy on NSL-KDD binary classification. While slightly lower than end-to-end supervised models, the self-taught approach demonstrated superior performance on minority attack categories, achieving 89.4 percent F1-score on R2L attacks compared to 91.2 percent for the DNN baseline and 92.7 percent for RNN-IDS. This improvement suggests that unsupervised pre-training helps discover feature representations particularly beneficial for underrepresented classes where supervised training data is limited.

Performance on UNSW-NB15 dataset revealed similar trends with slightly lower absolute accuracy reflecting greater complexity and diversity of attack patterns in this more recent dataset. The DNN baseline achieved 97.3 percent accuracy on binary classification, while RNN-IDS attained 97.8 percent. For nine-class classification including various contemporary attack types, the DNN achieved 93.7 percent accuracy and RNN-IDS reached 94.4 percent. The recurrent architecture's advantage for temporal modeling proved particularly valuable on UNSW-NB15, which contains longer connection sequences and attacks exhibiting distinctive temporal signatures that static feature representations may miss.

Analysis of confusion matrices revealed that baseline models occasionally misclassified specific attack types as normal traffic, particularly for sophisticated attacks designed to mimic legitimate user behavior. R2L attacks including unauthorized access attempts frequently generated false negatives, as these attacks often involve only subtle deviations from normal authentication patterns. U2R attacks escalating privileges after initial compromise also proved challenging to detect, as the post-compromise activities might resemble legitimate administrative actions. These misclassifications motivated investigation of adversarial training to enhance model robustness against deceptive attack patterns.

## 4.2 Vulnerability to Adversarial Attacks

When subjected to adversarial attacks, baseline models trained exclusively on clean data exhibited severe performance degradation exposing critical vulnerabilities that adversaries could exploit to evade detection. Under FGSM attacks with perturbation budget epsilon of 0.1, the DNN baseline accuracy on NSL-KDD binary classification dropped precipitously from 98.4 percent to 36.8 percent, representing catastrophic failure of the detection system. At stronger perturbation level epsilon of 0.2, accuracy further deteriorated to 19.7 percent, barely exceeding random guessing performance that would achieve 50 percent accuracy for binary classification.

ISSN: 3079-6342

The RNN-IDS baseline, despite superior performance on clean data, demonstrated similar vulnerability to adversarial attacks. FGSM attacks at epsilon 0.1 reduced RNN-IDS accuracy from 98.9 percent to 32.4 percent on NSL-KDD, while epsilon 0.2 attacks decreased accuracy to 16.3 percent. The temporal modeling capabilities that provided advantages on clean data offered limited protection against adversarial perturbations, as attacks could simultaneously perturb features across multiple time steps to deceive the recurrent architecture. These results indicate that neither fully-connected DNNs nor recurrent architectures possess inherent robustness to adversarial attacks without explicit defensive training.

The self-taught learning approach with unsupervised pre-training showed modestly improved resilience compared to end-to-end supervised models, but still suffered significant degradation under adversarial attacks. FGSM epsilon 0.1 attacks reduced self-taught model accuracy from 97.9 percent to 42.3 percent, representing approximately 15 percentage points better adversarial accuracy than the DNN baseline. This moderate improvement suggests that features learned through unsupervised reconstruction objectives may capture more robust patterns compared to features optimized exclusively for supervised classification. However, the absolute adversarial accuracy of 42.3 percent remains far below acceptable levels for operational security systems.

Vulnerability under PGD attacks proved even more severe, with iterative refinement discovering stronger adversarial examples that more thoroughly exploited model weaknesses. PGD attacks with epsilon 0.1 and 40 iterations reduced DNN baseline accuracy to 24.6 percent on NSL-KDD, substantially lower than the 36.8 percent achieved by FGSM at the same perturbation budget. The iterative optimization in PGD enables discovery of adversarial examples closer to decision boundaries, maximizing misclassification probability within the allowed perturbation region. At epsilon 0.2, PGD attacks completely broke baseline models, achieving only 11.2 percent accuracy that falls below the 20 percent expected for five-class random guessing.

Analysis of adversarial examples revealed that relatively small perturbations concentrated on critical features could dramatically alter model predictions. For network traffic classification, adversarial algorithms discovered that modifying statistical features such as packet counts, byte rates, and inter-arrival time statistics proved particularly effective for evading detection. These features, while informative for distinguishing attack patterns, also exhibited high sensitivity to perturbations due to their continuous numeric nature enabling fine-grained manipulation. Categorical features encoded as binary indicators proved more resistant to adversarial manipulation, as perturbations must exceed larger thresholds to flip feature values from zero to one or vice versa.

The transferability of adversarial examples between different model architectures exacerbated security concerns. Adversarial examples generated using the DNN model successfully fooled the RNN-IDS model in 67 percent of cases, despite architectural differences between fully-connected and recurrent networks. Similarly, adversarial examples crafted against the RNN-IDS model transferred to deceive the DNN baseline with 62 percent success rate. This transferability enables black-box attacks where adversaries generate adversarial examples using surrogate models and apply them against target systems whose architectures remain unknown, significantly reducing barriers to successful evasion.

Performance degradation proved particularly severe for minority attack categories that were already challenging to detect on clean data. R2L attacks, achieving 91.2 percent F1-score on

ISSN: 3079-6342

clean data, dropped to only 8.3 percent F1-score under FGSM epsilon 0.1 attacks. U2R attacks similarly declined from 87.9 percent to 6.7 percent F1-score. The disproportionate impact on rare attack types suggests that models trained on imbalanced datasets develop decision boundaries particularly vulnerable to adversarial perturbations for underrepresented classes. This vulnerability compounds the inherent challenge of detecting sophisticated attacks with limited training examples, motivating investigation of adversarial training approaches specifically addressing minority class robustness.

## 4.3 Impact of Adversarial Training on Model Robustness

Incorporating adversarial training dramatically improved model robustness against gradient-based attacks while maintaining competitive performance on clean data. DNN models trained with 30 percent FGSM-generated adversarial examples achieved substantially enhanced resilience compared to baseline models. On NSL-KDD binary classification, the adversarially-trained DNN maintained 86.7 percent accuracy under FGSM epsilon 0.1 attacks, compared to the baseline catastrophic failure at 36.8 percent accuracy. At stronger epsilon 0.2 perturbation level, adversarial training preserved 79.8 percent accuracy, demonstrating that trained models learned decision boundaries significantly more resistant to perturbations.

RNN-IDS models benefited similarly from adversarial training, with the recurrent architecture's temporal modeling capabilities complementing adversarial robustness. The adversarially-trained RNN-IDS achieved 88.4 percent accuracy under FGSM epsilon 0.1 attacks and 82.1 percent under epsilon 0.2 attacks. The slightly superior performance compared to adversarially-trained DNNs suggests that recurrent connections provide additional constraints that help stabilize decision boundaries under perturbation. Maintaining hidden state across sequential inputs may limit the extent to which adversaries can manipulate predictions through coordinated feature modifications across time steps.

The self-taught learning framework enhanced by adversarial training at both the feature learning and classification stages demonstrated competitive robustness. Adversarially-trained models using the architecture in Figure 3 achieved 85.9 percent accuracy under FGSM epsilon 0.1 attacks and 80.4 percent under epsilon 0.2 attacks. The two-stage adversarial training approach, hardening both the sparse autoencoder and soft-max classifier, created comprehensive defenses operating at multiple levels of the detection pipeline. Unsupervised adversarial training of the autoencoder encouraged learning of features robust to reconstruction-based attacks, while supervised adversarial training of the classifier ensured robust decision boundaries in the learned feature space.

Importantly, adversarial training achieved these substantial robustness improvements with modest impact on clean data performance. The DNN trained with 30 percent adversarial examples maintained 96.7 percent accuracy on clean NSL-KDD test data, representing only 1.7 percentage point reduction compared to the 98.4 percent baseline. RNN-IDS clean accuracy declined from 98.9 percent to 97.3 percent, a 1.6 percentage point decrease. These moderate reductions in clean accuracy represent acceptable trade-offs for the substantial robustness improvements, particularly in security-critical environments where adversarial threats are anticipated.

Increasing the proportion of adversarial examples during training from 30 percent to 50 percent provided incremental robustness improvements at the cost of slightly larger clean accuracy reductions. DNN models trained with 50 percent adversarial examples achieved 89.2

ISSN: 3079-6342

percent accuracy under FGSM epsilon 0.1 and 83.6 percent under epsilon 0.2, representing approximately 2-4 percentage point improvements over the 30 percent adversarial training configuration. However, clean accuracy declined to 95.1 percent, a 3.3 percentage point reduction from baseline. These results reveal a continuous trade-off space between adversarial robustness and clean performance that practitioners can navigate based on their specific threat models and operational requirements.

Against the more challenging PGD attacks, adversarial training demonstrated substantial but somewhat reduced defensive effectiveness compared to FGSM scenarios. DNN models trained with 50 percent adversarial examples maintained 81.8 percent accuracy under PGD epsilon 0.1 attacks, significantly outperforming the baseline catastrophic failure at 24.6 percent but showing more degradation than the 89.2 percent achieved under FGSM attacks at the same perturbation budget. This performance gap reflects the fundamental challenge that PGD's iterative refinement discovers stronger adversarial examples than the FGSM perturbations used during training. Despite this gap, adversarially-trained models maintained functional detection capabilities even under strong iterative attacks that completely broke baseline models.

Cross-dataset evaluation revealed important insights into generalization properties of adversarially-trained models. DNN models trained with adversarial examples on NSL-KDD and evaluated on UNSW-NB15 maintained improved robustness compared to baseline models, achieving 78.2 percent accuracy under FGSM epsilon 0.1 attacks versus 34.7 percent for the baseline. While lower than the 86.7 percent achieved on NSL-KDD adversarial test data, this cross-dataset robustness demonstrates that adversarial training learns generalizable robust features rather than simply memorizing specific perturbation patterns. Organizations can potentially leverage adversarially-trained models developed in one network environment and deploy them in different contexts while retaining defensive benefits.

Analysis of robustness across different attack categories revealed that adversarial training provided more uniform improvements for well-represented classes compared to minority categories. DoS attacks, the most prevalent category in NSL-KDD, maintained F1-scores above 93 percent under FGSM epsilon 0.1 attacks after adversarial training. R2L attacks, severely impacted by adversarial perturbations in baseline models (8.3 percent F1-score), recovered to 71.4 percent F1-score with adversarial training. While substantial improvement, this performance remains below the 91.2 percent achieved on clean data, suggesting that minority class robustness requires additional attention through specialized sampling techniques or class-weighted adversarial training objectives.

## 4.4 Computational Efficiency and Training Dynamics

The computational overhead of adversarial training proved significant but manageable for practical deployment in production intrusion detection systems. Training with 30 percent adversarial examples increased total training time by approximately 2.7 times compared to baseline training on NSL-KDD using the DNN architecture. For a baseline training requiring 45 minutes on GPU hardware, adversarial training extended duration to approximately 122 minutes. This overhead stems primarily from adversarial example generation during training, which requires forward passes to compute loss, backward passes to compute gradients with respect to inputs, and perturbation generation steps.

ISSN: 3079-6342

Training with 50 percent adversarial examples resulted in 4.2 times longer training duration compared to baseline, requiring approximately 189 minutes for NSL-KDD. The super-linear scaling with adversarial proportion reflects that higher adversarial ratios necessitate more frequent adversarial generation, while also potentially requiring additional training epochs to converge due to the more challenging optimization landscape created by diverse adversarial perturbations. Organizations must balance these computational costs against security benefits when determining appropriate adversarial training configurations for their specific deployment scenarios.

RNN-IDS adversarial training incurred additional computational costs beyond those of DNN training due to the sequential processing required by recurrent architectures. Baseline RNN-IDS training on NSL-KDD required approximately 78 minutes, while adversarial training with 30 percent adversarial examples extended duration to 243 minutes (3.1 times baseline). The recurrent architecture requires processing sequences element-by-element during both forward and backward passes, increasing computational requirements compared to fully-connected networks processing entire feature vectors simultaneously. Despite these costs, the superior performance of adversarially-trained RNN-IDS on sequential attack patterns justifies the additional computational investment for scenarios where temporal modeling provides value.

Critically, inference time for adversarially-trained models remained identical to baseline models, as adversarial example generation occurs exclusively during training. The deployed model architecture is unchanged from baseline, processing input features through the same forward pass computations. This property enables deployment of robust models without sacrificing real-time detection capabilities essential for operational security systems monitoring high-volume network traffic. The computational overhead of adversarial training represents a one-time cost during model development rather than recurring cost during operational deployment, improving the practical feasibility of adversarial robustness.

Training dynamics analysis revealed that adversarial training alters convergence patterns compared to baseline training on clean data. Standard training typically achieves rapid initial improvement with training loss decreasing dramatically in early epochs before plateauing as models approach optimal performance. Adversarial training exhibits slower initial convergence with more gradual loss reduction across extended training periods. This slower convergence reflects the more challenging optimization landscape where models must simultaneously achieve good performance on clean data while remaining robust to adversarial perturbations, requiring more careful navigation of the joint objective.

The curriculum-based adversarial training approach, gradually increasing perturbation strength throughout training, demonstrated improved convergence compared to fixed-strength adversarial training. Starting with weak perturbations (epsilon 0.01) and progressively strengthening to final targets (epsilon 0.1-0.2) enabled models to first establish robust features for small perturbations before facing more challenging examples. This curriculum reduced training time by approximately 15 percent compared to fixed-strength adversarial training while achieving equivalent or slightly better final robustness, suggesting that progressive difficulty scheduling improves optimization efficiency for adversarial training.

Learning rate scheduling proved particularly important for adversarial training, as the complex loss landscape benefits from adaptive adjustment of step sizes throughout training.

ISSN: 3079-6342

We employed learning rate decay reducing the learning rate by factor 0.5 when validation loss plateaued for 5 consecutive epochs, enabling fine-grained weight adjustments in later training stages after initial rapid improvement. Without learning rate scheduling, adversarial training exhibited instability with training loss oscillating rather than smoothly decreasing, occasionally resulting in degraded final performance compared to properly scheduled training.

Stability analysis across multiple random initializations revealed good reproducibility of adversarial training results. Standard deviations of reported accuracy metrics across five independent training runs remained below 1.4 percentage points for both clean and adversarial evaluation conditions. This stability gives confidence that observed robustness improvements reflect fundamental properties of adversarial training rather than fortuitous parameter configurations or lucky initializations. The reproducibility facilitates deployment of adversarial training in production pipelines where consistent performance is essential for operational planning.

## 5. Conclusion

This research comprehensively investigated adversarial training methodologies for enhancing robustness of deep learning-based Network Intrusion Detection Systems against sophisticated evasion attacks. Through systematic experimentation across multiple neural network architectures including fully-connected Deep Neural Networks, Recurrent Neural Networks, and self-taught learning frameworks with sparse autoencoders, we demonstrated that incorporating adversarial examples during training provides substantial defensive benefits. Adversarially-trained models maintained detection accuracy exceeding 86 percent under FGSM attacks and 81 percent under stronger PGD attacks, representing dramatic improvements over baseline models exhibiting catastrophic failure with accuracy below 37 percent under equivalent adversarial conditions.

The architectural analysis revealed that different neural network structures offer complementary advantages for adversarially robust intrusion detection. The DNN architecture illustrated in Figure 1 provided efficient processing of static feature representations with straightforward adversarial training implementation, achieving strong robustness improvements with moderate computational overhead. The RNN-IDS framework depicted in Figure 2 leveraged temporal modeling capabilities to capture sequential attack patterns, demonstrating slightly superior performance particularly for distributed attacks manifesting across multiple network events. The self-taught learning approach shown in Figure 3 implemented comprehensive defenses through multi-stage adversarial training of both feature extraction and classification components, achieving competitive robustness while providing benefits for minority attack categories through unsupervised pre-training.

The experimental findings reveal fundamental trade-offs between adversarial robustness and clean data performance that practitioners must carefully balance based on specific threat models and operational requirements. Adversarial training with 30 percent adversarial examples provided substantial robustness improvements of approximately 50 percentage points under epsilon 0.1 attacks while reducing clean accuracy by only 1-2 percentage points. Increasing adversarial proportion to 50 percent yielded incremental robustness gains of 2-4 percentage points but at cost of 3-4 percentage point clean accuracy reduction. Organizations deploying intrusion detection systems must calibrate adversarial training intensity based on

ISSN: 3079-6342

their assessment of adversarial threat likelihood versus importance of maximizing detection rates for unperturbed attacks.

The generalization properties of adversarially-trained models across different datasets and attack types provide encouraging evidence that adversarial training learns robust features rather than overfitting to specific perturbation patterns. Models trained with adversarial examples on NSL-KDD maintained substantial robustness when evaluated on UNSW-NB15, suggesting that organizations can leverage adversarially-trained models developed in one network environment and deploy them in different contexts with retained defensive benefits. This transferability reduces practical burden of adversarial training by enabling reuse of robust models across multiple deployments, though some fine-tuning on target environment data may further improve performance.

However, several important limitations warrant acknowledgment and suggest directions for future research. Our evaluation focused exclusively on white-box gradient-based attacks where adversaries possess complete knowledge of model architectures and parameters, representing pessimistic but not necessarily realistic threat models. Future work should investigate robustness against black-box attacks, decision-based attacks requiring only prediction outputs, and adaptive adversaries who craft attacks specifically designed to evade adversarially-trained models. Understanding performance under diverse threat scenarios will provide more complete assessment of practical defensive effectiveness.

The experiments utilized static benchmark datasets that may not fully capture dynamic nature of real-world network traffic and evolving attack patterns. Deploying and evaluating adversarially-trained NIDS in operational network environments would provide critical insights into practical effectiveness and potential failure modes not apparent in offline evaluation. Live deployment would also enable investigation of concept drift where network traffic distributions shift over time, potentially degrading both clean accuracy and adversarial robustness if models are not periodically retrained with recent data reflecting current threat landscapes.

The computational overhead of adversarial training remains a practical concern requiring continued research into efficient training methodologies. While experiments demonstrated manageable training time increases of 2-4 times compared to standard training, this overhead could become prohibitive for very large datasets or when frequent model retraining is required to adapt to evolving threats. Research into fast adversarial example generation techniques, selective adversarial training focusing on difficult examples near decision boundaries, and knowledge distillation approaches for creating compact robust models represents important directions for reducing computational barriers to adversarial training adoption.

Future research should explore integration of adversarial training with complementary defensive mechanisms to provide defense-in-depth against sophisticated adversaries. Combining adversarial training with input sanitization techniques that remove or smooth adversarial perturbations, ensemble methods leveraging diverse models to reduce transferability, and anomaly detection identifying out-of-distribution adversarial examples may offer synergistic protections. Multi-stage defense architectures employing adversarial training at primary detection layers while using secondary verification mechanisms could provide comprehensive security against adaptive adversaries employing diverse evasion strategies.

ISSN: 3079-6342

Investigation of certified defense approaches providing provable robustness guarantees represents another important research direction complementing empirical adversarial training. While adversarial training improves robustness through exposure to attack examples during training, it provides no guarantees about worst-case performance against unseen attacks. Certified defense methods based on interval bound propagation or randomized smoothing offer mathematical guarantees that predictions remain constant within defined perturbation regions. Combining empirical adversarial training for strong average-case robustness with certified defenses for worst-case guarantees on critical attack categories may provide optimal balance of security assurance and practical performance.

Application of adversarial training to emerging network security contexts including 5G networks, edge computing environments, and zero-trust architectures will be critical as these technologies become increasingly prevalent. The unique characteristics of these environments, including high-speed packet processing requirements in 5G, resource constraints at network edges, and continuous authentication requirements in zero-trust, necessitate adaptation of adversarial training approaches. Developing lightweight adversarially-robust models suitable for resource-constrained edge devices while maintaining security guarantees represents a particularly important challenge for securing IoT and mobile environments.

In conclusion, adversarial training represents a promising and practical approach for developing robust intrusion detection systems capable of maintaining security in face of adversarial threats. While not a complete solution to all security challenges, the substantial robustness improvements demonstrated in this work establish adversarial training as an essential component of comprehensive strategies for protecting machine learning-based security systems. As adversaries increasingly exploit vulnerabilities in ML-based defenses, adopting adversarial training and other defensive measures will become critical for maintaining effective cyber defense capabilities in hostile network environments.

## References

[1] Vinayakumar R, Alazab M, Soman K, Poornachandran P, Al-Nemrat A, Venkatraman S. Deep learning approach for intelligent intrusion detection system. IEEE Access. 2019;7:41525-41550.
[2] Zhang, H. (2025). Physics-Informed Neural Networks for High-Fidelity Electromagnetic Field Approximation in VLSI and RF EDA Applications. Journal of Computing and Electronic Information Management, 18(2), 38-46.
[3] Hu, X., Zhao, X., & Liu, W. (2025). Hierarchical Sensing Framework for Polymer Degradation Monitoring: A Physics-Constrained Reinforcement Learning Framework for Programmable Material Discovery. Sensors, 25(14), 4479.
[4] Qiu, L. (2025). Machine Learning Approaches to Minimize Carbon Emissions through Optimized Road Traffic Flow and Routing. Frontiers in Environmental Science and Sustainability, 2(1), 30-41.
[5] Reddy, M. K. J., Swaroop, A. S., Prasad, A. H., Nithin, D., & Singh, T. (2024). Artificial Neural Networks in Cryptography: Applications, Challenges, and Future Directions for Secure Systems. Frontiers in Collaborative Research, 2(1s), 20-28.
[6] Khazane, H., Ridouani, M., Salahdine, F., & Kaabouch, N. (2024). A holistic review of machine learning adversarial attacks in IoT networks. Future Internet, 16(1), 32.
[7] Antoniou, N. (2022). Improving projected gradient descent based adversarial attacks.
[8] Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. IEEE transactions on neural networks and learning systems, 30(9), 2805-2824.
[9] Huang, L., Zhang, C., & Zhang, H. (2020). Self-adaptive training: beyond empirical risk minimization. Advances in neural information processing systems, 33, 19365-19376.

**ISSN: 3079-6342**

[10] Wang N, Chen Y, Xiao Y, Lou W, Hou YT. Manda: On adversarial example detection for network intrusion detection system. IEEE Transactions on Dependable and Secure Computing. 2022;20(2):1139-1153.
[11] Salim, M. M., Rathore, S., & Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: A survey. Journal of Supercomputing, 76(7).
[12] Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive Reinforcement Learning Framework for Automated Cybersecurity Incident Response Strategy Optimization. Applied Sciences, 15(2), 951.
[13] Yang L, Moubayed A, Hamieh I, Shami A. Tree-based intelligent intrusion detection system in internet of vehicles. In: 2019 IEEE Global Communications Conference (GLOBECOM). IEEE; 2019. p. 1-6.
[14] Sauka K, Shin GY, Kim DW, Han MM. Adversarial robust and explainable network intrusion detection systems based on deep learning. Applied Sciences. 2022;12(13):6451.
[15] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Future Generation Computer Systems. 2019;100:779-796.
[16] Han D, Wang Z, Zhong Y, Chen W, Yang J, Lu S, Shi X, Yin X. Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors. IEEE Journal on Selected Areas in Communications. 2021;39(8):2632-2647.
[17] Apruzzese, G., Andreolini, M., Ferretti, L., Marchetti, M., & Colajanni, M. (2022). Modeling realistic adversarial attacks against network intrusion detection systems. Digital Threats: Research and Practice (DTRAP), 3(3), 1-19.
[18] Li, J., Fan, L., Wang, X., Sun, T., & Zhou, M. (2024). Product demand prediction with spatial graph neural networks. Applied Sciences, 14(16), 6989.
[19] Ren, S., & Chen, S. (2025). Large Language Models for Cybersecurity Intelligence, Threat Hunting, and Decision Support. Computer Life, 13(3), 39-47.
[20] Sun, T., Wang, M., & Han, X. (2025). Deep Learning in Insurance Fraud Detection: Techniques, Datasets, and Emerging Trends. Journal of Banking and Financial Dynamics, 9(8), 1-11.
[21] Ge, Y., Wang, Y., Liu, J., & Wang, J. (2025). GAN-Enhanced Implied Volatility Surface Reconstruction for Option Pricing Error Mitigation. IEEE Access.
[22] Tan, Y., Wu, B., Cao, J., & Jiang, B. (2025). LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access.
[23] Wang, Y., Ding, G., Zeng, Z., & Yang, S. (2025). Causal-Aware Multimodal Transformer for Supply Chain Demand Forecasting: Integrating Text, Time Series, and Satellite Imagery. IEEE Access.
[24] Chen, S., Liu, Y., Zhang, Q., Shao, Z., & Wang, Z. (2025). Multi-Distance Spatial-Temporal Graph Neural Network for Anomaly Detection in Blockchain Transactions. Advanced Intelligent Systems, 2400898.
[25] Mai, N. T., Cao, W., & Liu, W. (2025). Interpretable knowledge tracing via transformer-Bayesian hybrid networks: Learning temporal dependencies and causal structures in educational data. Applied Sciences, 15(17), 9605.
[26] Cao, W., Mai, N. T., & Liu, W. (2025). Adaptive knowledge assessment via symmetric hierarchical Bayesian neural networks with graph symmetry-aware concept dependencies. Symmetry, 17(8), 1332.
[27] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access.