Enhancing CNN–LSTM Intrusion Detection with Real-Time Response

Progress Report

University of Moratuwa

Department of Computer Science and Engineering

210144G - D. M. N. D. Dissanayake

1 Introduction

1.1 Background & Context

The growing reliance on interconnected digital infrastructures has made organizations increasingly vulnerable to cyberattacks. These attacks exploit weaknesses in networks, applications, and user practices, often resulting in severe financial, reputational, and operational damage. Cyber threats have evolved from isolated incidents into sophisticated campaigns, employing advanced tactics to bypass security mechanisms and remain undetected for extended periods.

Cyber threats can be classified into several categories, as illustrated in Figure 1. Some target the misuse of system resources, such as man-in-the-middle attacks that intercept communication channels. Others involve the compromise of user or root access privileges, achieved through techniques like SQL injection or cross-site scripting. Attacks on web applications are also prevalent, often providing adversaries with entry points to sensitive data. Malware remains one of the most diverse categories, encompassing viruses, worms, trojans, spyware, and ransomware, each capable of disrupting services or exfiltrating information. Another major category is denial-of-service attacks, which can be host-based, network-based, or distributed, all aimed at overwhelming resources and preventing legitimate access.

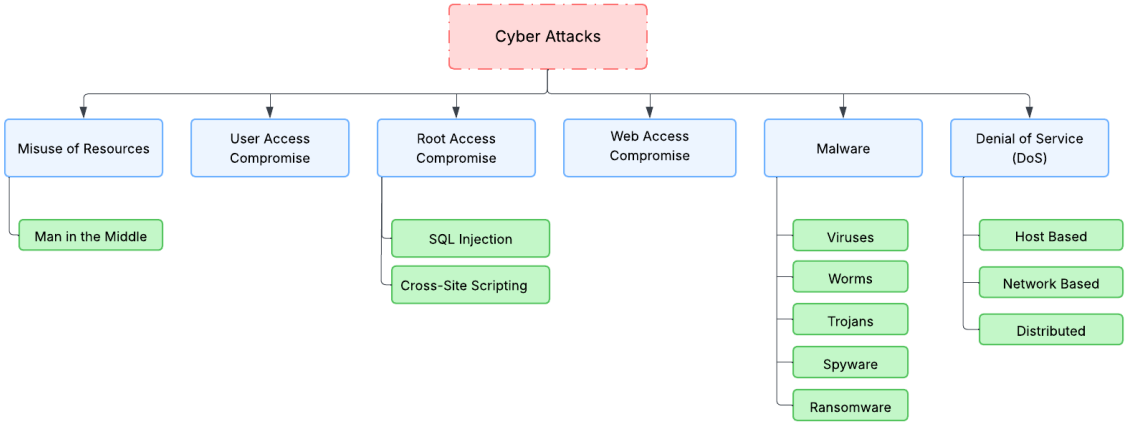

Figure 1: Classification of Cyber Threats

To counter these threats, organizations rely on a broad spectrum of defense mechanisms. These include firewalls that regulate traffic at network boundaries, intrusion detection and prevention systems that monitor activity for signs of compromise, and antivirus software that identifies malicious executables. Encryption protects data confidentiality, while multi-factor authentication strengthens identity verification. Other strategies such as security awareness training, patch management, network segmentation, incident response planning, and endpoint protection contribute to building a layered defense architecture.

Despite these efforts, traditional defense mechanisms are limited in their ability to adapt to new and evolving attack patterns. Rule-based systems, while effective against known threats, often struggle with previously unseen attacks and produce high false alarm rates. This gap has driven the integration of machine learning into cybersecurity. Machine learning models can learn from historical data to distinguish between normal and malicious behavior, thereby improving detection accuracy and reducing false positives. More importantly, they are capable of generalizing to novel attack types, a feature that is

crucial in today's dynamic threat landscape.

Deep learning approaches have further strengthened this capability. In particular, hybrid architectures such as convolutional neural networks combined with long short-term memory networks (CNN–LSTM) have shown state-of-the-art performance for intrusion detection. [1] [2] CNN layers are adept at capturing spatial correlations in traffic features, while LSTM layers model sequential dependencies across time. Together, they provide a powerful framework for analyzing complex network data.

In this research, the focus is placed on **Intrusion Detection Systems**, which serve as a critical line of defense in identifying malicious activity. By building upon the **CNN–LSTM** baseline proposed by Bamber et al. (2025) [1], this project aims not only to enhance detection accuracy through mechanisms such as focal loss, SMOTE, and attention layers but also to explore the integration of real-time response capabilities. This direction ensures that the proposed system is both academically rigorous and practically relevant in addressing the modern challenges of cybersecurity.

1.2 Research Objectives

1. Reproduce the CNN-LSTM baseline on NSL KDD[1] to establish a performance benchmark.
2. Enhance model performance through:
    - Focal loss for imbalance handling
    - SMOTE oversampling of minority attack classes
    - Incorporation of an attention mechanism for feature focusing
    - Hyperparameter tuning
3. Integrate real-time response simulation, measuring inference latency and throughput.
4. Deliver a reproducible, well-documented system with ablation studies and GitHub repository.

2 Literature Review

Machine learning models fundamentally reshape cybersecurity strategies by enabling adaptive, proactive, and predictive defense postures [3]. By continuously processing and analyzing immense volumes of network traffic data, these models can identify subtle anomalies and complex patterns in real time, thereby allowing for the detection of novel threats before they can inflict significant damage [3, 4]. This approach moves beyond the limitations of a closed-world system, which can only identify what it has been explicitly trained to recognize, toward an open-world model capable of generalizing from learned patterns to new data [5]. Furthermore, the application of ML extends beyond mere detection, driving the automation of threat response. Machine learning-based systems can automatically initiate predefined mitigation actions such as isolating affected systems or blocking malicious IP addresses within seconds of a threat being identified, minimizing potential harm and expediting incident response [3]. This fundamental change in methodology, from a reactive, rule-based defense to a predictive, behavior based one, represents a critical advancement in the field of network security. Here I will be focusing on recent work on **Intrusion Detection** and **Real Time Responses**

[1] `https://web.archive.org/web/20150205070216/http://nsl.cs.unb.ca/NSL-KDD/`

2.1 Intrusion Detection Systems

Intrusion Detection Systems (IDS) are essential components of a layered defense architecture, functioning as a device or software application that continuously monitors a network or system for signs of malicious activity or policy violations [6]. Their primary purpose is to identify potential threats and provide an early warning to system administrators, capturing and logging critical information that can be used for later investigation of a data breach [7]. The field of IDS has historically been dominated by two principal methodologies: signature-based detection and anomaly-based detection [8, 9].

Signature-based IDS, also known as misuse detection, operates on a principle similar to traditional antivirus software. It maintains a database of known intrusion signatures, and any network traffic that matches a signature is flagged as an attack [8]. This method is highly accurate and yields a low rate of false alarms for attacks whose signatures are contained within the database [8]. However, its dependency on a finite database renders it ineffective against new and evolving attacks for which no signatures exist, leading to a high rate of false negatives for novel threats [8]. In contrast, **Anomaly-Based IDS** establishes a baseline of normal network behavior by profiling the activities of users and systems [8]. Any deviation from this established baseline is identified as anomalous and, therefore, potentially malicious. This approach is powerful because it can detect previously unknown intrusions, making it effective against zero-day attacks [8]. The main drawback of anomaly-based systems is their potential for a higher rate of false alarms, as they may misclassify legitimate but unusual behavior as a threat [8]. The inherent trade-off between the precision of signature-based systems and the generality of anomaly-based systems has created a demand for more sophisticated, hybrid solutions that can mitigate the weaknesses of each approach. This need has been the catalyst for the development of advanced deep learning models that can learn to identify both known attack patterns and general behavioral anomalies.

2.2 Advanced Hybrid Deep Learning Architectures for Intrusion Detection

The limitations of traditional IDS and the complexity of modern network data have driven the adoption of deep learning, with hybrid architectures proving to be exceptionally effective. The combination of convolutional neural networks (CNNs) and long short-term memory (LSTM) networks is particularly prominent due to its unique ability to process the bi-faceted nature of network traffic.

2.2.1 The Efficacy of CNN-LSTM Models

The success of the hybrid CNN-LSTM architecture for intrusion detection stems from its design, which mirrors the intrinsic structure of network traffic data. Network traffic can be conceptualized as having both static and dynamic characteristics. The CNN component is uniquely suited to detect spatial patterns of network traffic data by acting as a feature extractor [1]. It analyzes the relationships and hierarchies among features in a single data record, identifying local patterns indicative of malicious activity.

Following this initial feature extraction, the LSTM component takes over. Network traffic is not a static collection of records but a continuous stream of sequential data where dependencies exist across time. The LSTM, a specialized type of recurrent neural network (RNN), is designed to model sequential dependencies across time and analyze the dynamic aspects of this data stream [1]. By processing the

sequence of features extracted by the CNN, the LSTM can identify temporal patterns that may indicate a sophisticated attack unfolding over a longer duration. This synergy provides a comprehensive and robust framework for distinguishing between benign and malicious network behavior, leading to higher accuracy and a reduced false positive rate than either model could achieve in isolation [1,8]. This hybrid approach has demonstrated superior performance on diverse datasets like UNSW-NB15 and X-IIoTID, achieving high accuracies such as 93.21% and 99.84% respectively for binary classification [8].

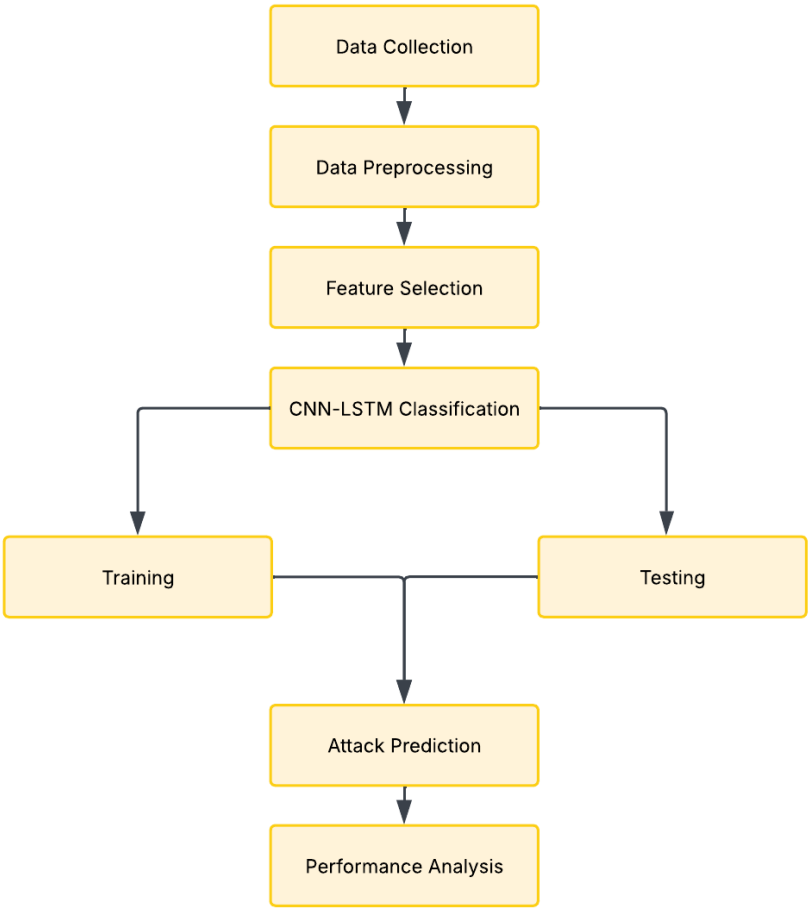

Figure 2: CNN-LSTM Flow

2.2.2 Enhancing Performance with Attention Mechanisms

While the CNN-LSTM architecture provides a strong foundation for intrusion detection, its performance can be further refined by incorporating an attention mechanism [10]. This enhancement addresses a key challenge in network data analysis: the presence of vast amounts of heterogeneous and potentially irrelevant data that can obscure the subtle patterns of an attack.

An attention mechanism is a powerful technique that allows a model to highlight the utmost informative input features [10]. In the context of a hybrid IDS, this mechanism enables the model to dynamically assign a weight to each feature and time step based on its significance to the detection

task [10]. This is a fundamental improvement over traditional architectures where all features contribute equally, a design that can dilute the relevance of a few critical attack indicators that are often buried within a large volume of non-malicious data [10]. The attention layer acts as a computational focusing lens, enabling the model to pay more attention to the elements of the input that are most indicative of a threat [11]. By directing the model's focus to important attack features while effectively ignoring unnecessary data, the attention mechanism significantly enhances the system's ability to identify malicious activities, particularly in high-dimensional or noisy datasets [10]. Empirical studies confirm the value of this addition, showing that Attention-CNN-LSTM models can achieve 94.8–97.5% accuracy on datasets like NSL-KDD and Bot-IoT [10]. An ablation study has further confirmed that the attention layer alone contributed to a 3-4% improvement in F1-score and MCC [10], demonstrating its substantial impact on a model's discriminative power.

2.2.3 Real-Time Response for Operational Viability

Beyond the academic pursuit of high detection accuracy, the operational value of an IDS is fundamentally tied to its ability to perform real-time inference with minimal latency and high throughput. An IDS is only effective if its packet processing rate must exceed the expected maximum packet arrival rate [5]. This requirement is particularly acute in modern environments characterized by high-volume network traffic and the proliferation of resource-constrained edge devices [12]. The focus of research is shifting from static, offline performance on a test set to dynamic, real-world deployability.

Research demonstrates that advanced deep learning models can be optimized to meet the stringent demands of real-time deployment. For example, a hybrid Attention-CNN-LSTM model has been shown to support real-time inference with sub-35ms latency and a throughput of over 1200 records per second, making it suitable for high-traffic environments [10]. Even more significant advancements have been made through optimization techniques specifically designed for resource-constrained hardware. A lightweight deep learning model, the DNN-KDQ framework, achieved an ultra-low inference time of 0.07 ms per sample while maintaining a high prediction test accuracy of 99.43% [12]. This was accomplished by combining techniques such as knowledge distillation and quantization, which drastically reduce the model's size from 196.77 KB to 20.18 KB [12]. The ability to simultaneously optimize for high performance, low latency, and a reduced memory footprint represents a maturation of the field, where researchers are increasingly focused on creating models that are not only effective but also practically implementable in real-world operational scenarios.

2.3 NSL-KDD: Structure and Features

The NSL-KDD dataset is a benchmark widely used in NIDS research and is an improved version of the KDD Cup 99 dataset [13]. It is a structured dataset comprised of 125,973 training samples and 22,544 test samples, with each record characterized by 41 distinct features [13]. These features are logically categorized into four groups, providing a comprehensive view of network connections [13]:

1. **Basic Features (1–9):** This category includes fundamental packet information such as connection duration, protocol type, service, and flag [13].
2. **Content Features (10–22):** These features are derived from the content of the packets themselves

and are particularly useful for identifying attacks that target specific application-level behaviors, such as R2L and U2R attacks which may involve multiple failed login attempts [13].

3. **Time-based Features (23–31):** These features capture patterns within a short time window, typically 2 seconds, and are used to provide a temporal context for network connections (e.g., `count` and `srv count`) [13].

4. **Host-based Features (32–41):** These features are calculated over longer time windows, spanning more than 2 seconds, and describe attack patterns originating from a destination host (e.g., `dst host count` and `dst host srv count`) [13].

The dataset also contains a five-class target for classification, including Normal, DoS (Denial of Service), Probe, R2L, and U2R attacks [13].

Table 1: Overview of NSL-KDD Dataset Features

Feature Category	**Description and Example Features**
Basic Features (1–9)	Fundamental packet information (e.g., `duration`, `protocol_type`, `service`, `flag`) [13]
Content Features (10–22)	Derived from packet content, useful for R2L and U2R attacks (e.g., multiple failed logins) [13]
Time-based Features (23–31)	Patterns within a 2-second time window (e.g., `count`, `srv_count`) [13]
Host-based Features (32–41)	Attack patterns from destination to host over longer time windows (e.g., `dst_host_count`, `dst_host_srv_count`) [13]

3 Proposed Plan

3.1 Dataset and Preprocessing (NSL-KDD)

We use NSL-KDD due to its corrected design over KDD'99 and its standardized splits (KDDTrain+, KDDTest+), which enable fair comparisons across IDS studies [14]. Each record has 41 features spanning basic, content, time-based, and host-based categories with a five-class target (Normal, DoS, Probe, R2L, U2R); the severe imbalance in R2L/U2R motivates imbalance-aware training and metrics [15].

Preprocessing pipeline (reproducible):

1. Map attack types to the 5-class target (Normal/DoS/Probe/R2L/U2R).

2. One-hot encode the three categorical features (`protocol_type`, `service`, `flag`); standardize the remaining numeric features (e.g., z-score).

3. Use `KDDTrain+` exclusively for training/validation; `KDDTest+` for final held-out evaluation to avoid leakage and to mirror how NSL-KDD was intended to be used. []

4. Export and version the preprocessing as a deterministic `scikit-learn` Pipeline to guarantee identical transforms in both offline and streaming (real-time) modes.

3.2 Baseline Reproduction (CNN–LSTM)

Reimplement the CNN LSTM baseline of Bamber *et al.* in TensorFlow and verify reported performance on NSL-KDD [1]. The hybrid structure CNN feature extraction followed by LSTM aggregation and a dense softmax head is consistent with contemporary IDS literature. Train on KDDTrain+, early-stop on a validation fold, and report final metrics on KDDTest+; store the exact random seeds, splits, and commit hashes.

3.3 Enhancements

Focal Loss. To address class imbalance, cross-entropy is replaced with Focal Loss, which concentrates the gradient on hard/minority examples and has been shown to improve minority recall and macro-F1 in IDS studies [16, 17].

Plan

Use the standard focal loss,

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t)$$

with class-specific α. Tune $\gamma \in \{1, 2, 3\}$ and optionally blend with label smoothing (small ϵ) to stabilize training.

SMOTE. Over-sampling via SMOTE generates synthetic minority instances that expand the decision boundary for rare classes; multiple IDS works (including on NSL-KDD) find improved minority recall and overall F1 when pairing SMOTE with modern learners. Variants like SMOTE-IPF or cluster-SMOTE further reduce noise [18].

Plan

Apply SMOTE after train/val split and before scaling to avoid leakage.

Compare: no resampling vs **SMOTE** vs **class-weighted focal loss** vs both. Report per-class metrics.

Attention. Self-attention/adaptive attention highlights the most informative latent features/time steps, improving discriminative power in hybrid CNN–LSTM IDS and boosting F1/MCC across NSL-KDD and other datasets. Recent peer-reviewed work shows Attention-CNN-LSTM outperforms non-attention hybrids. [10].

Plan

Insert a lightweight additive or scaled dot-product attention block on top of LSTM outputs; compare to the baseline (no attention). Keep parameter count modest to preserve latency.

Hyperparameter Optimization. IDS performance is sensitive to LR, batch size, filter widths, LSTM units, and dropout. Bayesian optimization/pruning frameworks (e.g., Optuna) consistently outperform manual or random search for IDS models on NSL-KDD. [19].

Plan

Objective: macro-F1 on validation (to reflect imbalance).
Early pruning on plateaued trials; retain top-k configs for retraining.

3.4 Real-Time Response (Streaming Simulation)

An IDS is only useful operationally if it can keep up with traffic and act (alert, block, quarantine) quickly. Prior work demonstrates online intrusion detection with low-latency models on commodity or edge hardware, motivating latency-aware design and measurement. [5, 20]

Stream Simulation	Feed normalized records sequentially (or in micro-batches of 32–128) at controlled rates.
Model Export	Save the TensorFlow model as a `SavedModel` and run inference including preprocessing latency.
Attack Response	On predicted attack, log + console alert, and emit JSON policy to a stub firewall.
Metrics	Record latency (ms/sample) as median and p95, and throughput (records).

4 Time Line

Time line is planned as per the given instructions targeting completion of the research before the 12th week.

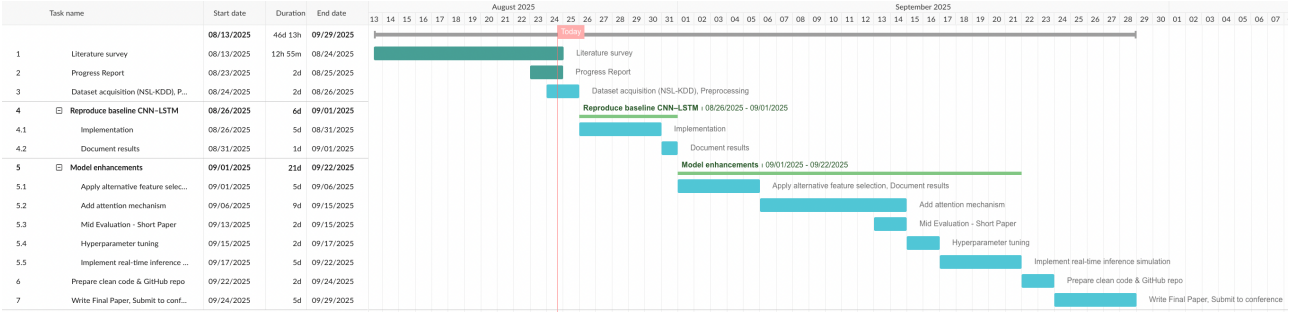

Figure 3: Time Line

5 Conclusion

This project set out to explore the design and enhancement of a hybrid CNN–LSTM based intrusion detection system using the NSL-KDD dataset. The motivation stems from the increasing sophistication

of cyber threats and the limitations of traditional defense mechanisms, which struggle to adapt to evolving attack patterns. By reproducing the baseline CNN–LSTM model and incrementally enhancing it with focal loss, SMOTE, attention layers, and automated hyperparameter optimization, this work aims to improve both the accuracy and robustness of intrusion detection.

Beyond detection accuracy, the inclusion of a real-time response simulation ensures the system is evaluated not only as a research prototype but also as a practical defense mechanism, capable of operating under latency and throughput constraints. The methodology balances theoretical rigor with applied relevance, grounding each enhancement in established research while tailoring the pipeline to the realities of class imbalance and streaming data.

Ultimately, this project aspires to deliver an intrusion detection framework that advances the state of the art in both detection effectiveness and operational readiness. While focused on NSL-KDD for benchmarking, the findings are expected to provide generalizable insights for deploying deep learning-based IDS solutions in modern networked environments.

References

[1] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," vol. 148, p. 104146. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404824004516

[2] M. Aljanabi and N. Kumaran, "Effective intrusion detection through hybrid CNN-LSTM and grey wolf optimization for feature selection in complex network environments," vol. 1, no. 1, pp. 22–32. [Online]. Available: https://www.gkijaret.com/index.php/gkijaret/article/view/3

[3] SailPoint, "How ai and machine learning are improving cybersecurity," 2025. [Online]. Available: https://www.sailpoint.com/identity-library/how-ai-and-machine-learning-are-improving-cybersecurity

[4] Comparitech, "Machine learning enhances threat detection by analyzing network traffic, identifying anomalies, and improving security with adaptive, real-time responses," 2025. [Online]. Available: https://www.comparitech.com/net-admin/machine-learning-threat-detection/

[5] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," in *Proceedings 2018 Network and Distributed System Security Symposium*. Internet Society. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-3_Mirsky_paper.pdf

[6] Wikipedia, "Intrusion detection system," 2025. [Online]. Available: https://en.wikipedia.org/wiki/Intrusion_detection_system

[7] DHS.gov, "Intrusion detection and prevention systems," 2025. [Online]. Available: https://www.dhs.gov/publication/intrusion-detection-and-prevention-systems

[8] A. M. Alashjaee, "A hybrid cnn+lstm-based intrusion detection system for industrial iot networks," *ResearchGate*, 2023. [Online]. Available: https://www.researchgate.net/publication/366919487_A_hybrid_CNN_LSTM-based_intrusion_detection_system_for_industrial_IoT_networks

[9] M. Aljanabi, "Effective intrusion detection through hybrid cnn-lstm and grey wolf optimization," *MDPI*, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/18/7856

[10] A. M. Alashjaee, "Deep learning for network security: an attention-CNN-LSTM model for accurate intrusion detection," vol. 15, no. 1, p. 21856, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41598-025-07706-y

[11] K. T. V. Nguyen, A. V. T. Le, and K. M. T. Vo, "attention mechanism in cnn-lstm for ids," *arXiv*, 2025. [Online]. Available: https://arxiv.org/html/2501.13962v1

[12] U. I. Umar, M. A. Abdullahi, and H. Danjuma, "Energy-efficient deep learning-based intrusion detection system for edge computing: a novel dnn-kdq model," *Journal of Cloud Computing*, vol. 14, no. 1, p. 32, 2025. [Online]. Available: https://www.researchgate.net/publication/393249275_Energy-efficient_deep_learning-based_intrusion_detection_system_for_edge_computing_a_novel_DNN-KDQ_model

[13] J. Jang, Y. An, D. Kim, and D. Choi, "Feature importance-based backdoor attack in nsl-kdd," *ResearchGate*, 2025. [Online]. Available: https://www.researchgate.net/publication/376446254_Feature_Importance-Based_Backdoor_Attack_in_NSL-KDD

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/5356528/

[15] J. Jang, Y. An, D. Kim, and D. Choi, "Feature importance-based backdoor attack in NSL-KDD," vol. 12, no. 24, p. 4953, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2079-9292/12/24/4953

[16] S. Khanam, I. Ahmedy, M. Y. I. Idris, and M. H. Jaward, "Towards an effective intrusion detection model using focal loss variational autoencoder for internet of things (IoT)," vol. 22, no. 15, p. 5822, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/22/15/5822

[17] M. Mulyanto, M. Faisal, S. W. Prakosa, and J.-S. Leu, "Effectiveness of focal loss for minority classification in network intrusion detection systems," vol. 13, p. 4, ADS Bibcode: 2020Symm...13....4M. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020Symm...13....4M

[18] T. B. Shana, N. Kumari, M. Agarwal, S. Mondal, and U. Rathnayake, "Anomaly-based intrusion detection system based on SMOTE-IPF, whale optimization algorithm, and ensemble learning," vol. 27, p. 200543. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305325000699

[19] M. Masum, H. Shahriar, H. Haddad, M. J. H. Faruk, M. Valero, M. A. Khan, M. A. Rahman, M. I. Adnan, and A. Cuzzocrea, "Bayesian hyperparameter optimization for deep neural network-based network intrusion detection," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 5413–5419. [Online]. Available: http://arxiv.org/abs/2207.09902

[20] Energy-efficient deep learning-based intrusion detection system for edge computing: a novel DNN-KDQ model | journal of cloud computing | full text. [Online]. Available: https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-025-00762-9