A Lightweight AI-Based Intrusion Detection System (IDS) Optimised For Resource-Constrained Networks

Claire Campbell
The University of Cape Town
Cape Town, South Africa
cmpcla004@myuct.ac.za

Sian Caine
The University of Cape Town
Cape Town, South Africa
cnxsia001@myuct.ac.za

Christopher Blignaut
The University of Cape Town
Cape Town, South Africa
blgchr003@myuct.ac.za

ABSTRACT

The rapid growth of Internet of Things (IoT) devices and network interconnectivity resulted in increased network traffic and cyber threat complexity. This has left low-resource networks such as edge devices and embedded systems especially vulnerable. Traditional Intrusion Detection Systems (TIDSs) struggle to scale sufficiently or effectively detect zero-day attacks. AI-based Intrusion Detection Systems (AI-IDSs) can process large volumes of network traffic and boast better detection accuracy and adaptability, however they are too computationally expensive for low-resource environments.

The LAIDS (Lightweight AI-based Intrusion Detection System) project addresses this gap by developing and evaluating lightweight AI-IDS architectures that operate effectively under strict resource limitations similar to a Mini PC: 4GB RAM, 500MB storage, no GPU acceleration, on a 64 bit quad core CPU with a clock speed of 1.5GHz. Studies show that DBN, GAN, Autoencoder, CNN-GRU, CNN-PCA, and CNN-LSTM models offer promising results in terms of performance and resource efficiency. The project will optimise these models through quantisation, pruning, and dimensionality reduction before being compared to a baseline TinyML CNN. The goal of the project is to contribute a real-time IDS framework for low-resource environments while revealing how resource limitations impact IDS effectiveness.

1 INTRODUCTION

The proliferation of digital systems and Internet of Things (IoT) devices has dramatically increased network traffic, resulting in more frequent and sophisticated cyber attacks [47, 50, 62]. Intrusion Detection System (IDS) frameworks are widely implemented to monitor network traffic and detect suspicious activity. However, Traditional Intrusion Detection Systems (TIDSs) struggle with growing data volumes and modern attack complexity [3, 17, 42].

To address these limitations, AI-based Intrusion Detection Systems (AI-IDSs) have emerged and gained widespread adoption for their ability to analyse high volumes of network traffic while offering greater adaptability to zero-day attacks and lower false positive rates [17, 37, 42, 58]. However, their higher computational demands limit their use in resource-constrained networks [20, 24, 37, 40]. This project defines resource-constrained networks as systems with 4GB RAM, 500MB storage, no GPU acceleration, and a 1.5GHz, 64 bit quad-core CPU - typical IoT and embedded device constraints [28, 40].

These limitations demand lightweight, efficient, and intelligent security solutions. The Lightweight AI-based Intrusion Detection System (LAIDS) project aims to develop a lightweight AI-IDS for deployment on network gateways or personal computers to classify and identify attack types in resource-constrained environments. Optimisation techniques such as pruning, which removes redundant neural connections to reduce size and computational costs [34]; quantisation, which lowers the precision of a neural network's weight and activation values to reduce its memory usage [49]; hyperparameter tuning, which involves selecting a learning process configuration to achieve optimal performance [7]; and knowledge distillation, which requires training a smaller model to mimic the behaviour of a larger, more accurate model [55, 56].

Previous research [6, 9, 12, 14, 16, 49, 55] suggests that the following models create ideal lightweight IDSs. Convolutional Neural Networks (CNNs) have shown strong performance as IDS classifiers [43]. A trend has shown that hybrid models typically outperform pure models. CNNs combined with Gated Recurrent Units (GRUs) show improved self-correction [16] and Principal Component Analysis (PCA) aids CNNs with dimensionality reduction [14]. Another promising hybrid model is a CNN with a Long Short-Term Memory (LSTM) model, which combines spatial and temporal pattern recognition [9]. Other models we will look at are Generative Adversarial Models (GANs), which are optimised for lightweight application using knowledge distillation [6]; Autoencoders (AEs) made lightweight through post-training optimisation [49]; and Deep Belief Network (DBN) models, which show promising IDS implementations through hierarchical feature learning [12].

1.1 Problem Statement

The vulnerability of low-resource networks such as IoT networks, edge devices, and embedded systems has become a critical area of research. Existing IDS solutions, TIDS, AI-IDS, and TinyML, either lack zero-day attack detection or are too computationally demanding for low-resource networks [3, 17, 20, 24, 37, 40]. This leaves low-resource networks particularly vulnerable. As such, there is a need for research into a lightweight, AI-driven IDS capable of zero-day attack detection in real time, while operating within a resource-constrained network.

1.2 Research Questions

To address the problem statement, the LAIDS project aims to: develop a series of lightweight AI-IDSs suitable for deployment in a resource-constrained network as well as explore how constrained computational resources influence IDS performance in terms of accuracy, precision and false negative rate. To achieve these aims, we propose the following research questions:

Question 1
How can a Hybrid CNN-GRU or GAN-based IDS outperform a TinyML-based IDS in terms of detection accuracy, false positive rate, and computational resource usage in a resource-constrained network environment?

Question 2
How can a Hybrid CNN-PCA or Autoencoder-based IDS achieve higher resource efficiency than a TinyML-based IDS, while maintaining comparable or better performance metrics in a resource-constrained network environment?

Question 3
How can a Hybrid CNN-LSTM or DBN-based IDS achieve greater resource efficiency, compared to a TinyML-based IDS, while maintaining or improving detection performance in a resource-constrained network environment?

2 BACKGROUND AND RELATED WORK

The following section provides essential context for the LAIDS project. It outlines prior research in lightweight intrusion detection and justifies the selection of the models outlined in the research questions.

2.1 CNN and CNN Hybrid Models

CNNs are deep learning models predominantly used for object recognition and image classification due to their ability to analyse high-dimensional data and recognise complex patterns [43, 51]. CNNs consist of multiple layers, each responsible for learning and extracting relevant features to enhance data classification [51]. These capabilities are well-suited for analysing and classifying network traffic as normal or malicious in IDSs [43].

To address the specific challenges of resource-constrained environments, researchers have explored lightweight CNN implementations. Sun and Zhao [55] proposed TinyNIDS, a CNN-based IDS optimised for edge devices in 6G networks through post-training quantisation and pruning to reduce latency and increase throughput. The LAIDS project will be primarily targeting networks operating in 4G, 5G, or LTE environments, which often have stricter resource limitations in terms of CPU power and memory.

Further enhancing CNN performance for IDS involves integrating them with dimensionality reduction technqiues such as PCA. PCA is an unsupervised technique that transforms high-dimensional data into a lower-dimensional space using *principal components*, while retaining the most significant variance [2, 53]. By reducing noise and complexity, PCA can improve a CNN's intrusion detection accuracy and efficiency [32, 53, 64].

Awotunde et al. [14] demonstrated how a hybrid Kernal PCA (KPCA) and CNN IDS for IoT environments can achieve high accuracy and precision. The model achieved a 99.35% accuracy rate as well as a 98.57% precision rate. Despite, the demonstration of strong detection capabilities, the research did not directly address resource usage or its optimisation for deployment in low-resource environments. The benefits of the dimensionality reduction through PCA in conjunction with CNNs will be explored.

Another effective approach to enhancing CNN-based IDSs is through hybrid architectures combining CNNs' feature extraction abilities with LSTMs to capture temporal dependencies in network traffic [52]. LSTMs, a type of Recurrent Neural Networks, excel at mapping temporal features in sequential data, mitigating the vanishing gradient problem and proving effective in modelling network traffic flows for IDS [25, 43]. This is beneficial for detecting time-based attack patterns [44].

Altunay and Albayrak [9] proposed a high-accuracy CNN-LSTM hybrid IDS, however the paper can be critiqued for its limited focus on resource-constrained environments. Similarly, Du et al. [18] found strong classification performance with a CNN-LSTM model for IoT security. Its ability to capture both spatial and temporal patterns in network traffic will be explored.

GRUs offer an alternative to LSTMs, providing a more computationally efficient solution to the vanishing gradient problem by simplifying the gating mechanism [16, 63]. Combining the temporal pattern recognition of GRUs with the spatial feature extraction of CNNs enables the detection of both malicious packets and sequence-based attacks.

Cao et al [16]'s CNN-GRU hybrid model achieved high accuracy, demonstrating its potential. However, optimisations need to be implemented for low-resource environments. Kilichev et al. [33] proposed a CNN-LSTM-GRU model for IoT electric vehicle charging stations that achieved particularly high accuracies for both binary and multi-class classifications. Little insight is provided on the model's resource usage and furthermore, the authors note that further optimisations are required [33].

2.2 Alternative Single Models

Research has also demonstrated the success of single DL IDS models. GANs consist of a generator that creates synthetic network traffic and a discriminator that distinguishes between real (normal) and generated (malicious) traffic. This approach allows for zero-day attack detection without relying on known attack signatures [43]. Only the discriminator component of the model needs to be deployed to the network, reducing the IDS size. Further optimisations such as pruning, quantisation, and activation functions such as LeakyReLu can be applied [35].

Ali et al. [6] proposed a lightweight IoT IDS using GANs and knowledge distillation. The GAN's generator augmented data for enhanced training and detection accuracy on uncommon attacks. Knowledge distillation compressed the model for resource-constrained deployment, enabling the model to achieve strong performance. Additionally, Park et al. [41] found that GANs, performed particularly well at detecting zero-day attacks. The factors listed above demonstrate the potential for lightweight GAN-based IDS.

DBNs are generative deep neural networks composed of stacked Restricted Boltzmann Machines (RBMs), enabling hierarchal feature learning and complex pattern detection [13]. An RBM is a generative stochastic neural network that learns a probability distribution over its input using a two-layered architecture: a hidden and a visible layer. By stacking RBMs, DBNs allow the hidden layer of one RBM to serve as the visible layer of the next, enabling the network to learn increasingly abstract representations and generate new data samples resembling the training set. This layered structure also makes DBNs particularly useful for filtering out noise during feature extraction and classification in IDSs [13, 65].

Balakrishnan et al. [13]'s DBN-based IDS showed high accuracy for common attacks but struggled with complex threats and lacked detailed resource usage analysis, such as CPU time and memory consumption. Similarly, Alom et al. [8] achieved high detection accuracy with a DBN on the NSL-KDD dataset. Further supporting DBN IDS, Wei et al. [61]'s optimised DBN achieved high detection speed and accuracy. That being said, their model was not concerned with resource usage requiring long training times.

AEs are unsupervised DL models that learn to reconstruct their input by efficiently compressing and decompressing it [21]. In the context of IDS, an AE attempts to reconstruct network traffic data after encoding it, after which a high reconstruction error for unseen traffic can indicate malicious activity [21]. Sharmila and Nagapadma [49] proposed lightweight AE models for IoT anomaly detection, focusing on balancing accuracy with resource efficiency through post-training optimisation such as pruning and quantisation. The model QAE-u8 demonstrated the lowest resource usage and outperformed the baseline AE in terms of resource efficiency, although it showed slightly reduced accuracy.

2.3 Limitations of Existing Work

While previous research [6, 9, 12, 14, 33, 49, 55, 61, 63] has explored various models for intrusion detection, a significant limitation is the underdeveloped focus on model optimisation for resource usage, specifically for deployment in low-resource environments. This prior research [8, 9, 12, 14, 16, 41, 49, 55, 61] prioritised accuracy over practical constraints such as memory, computation time, and energy consumption on edge devices. In contrast, the LAIDS project directly addresses this gap by explicitly designing and optimising models for resource-constrained environments. Building on previous work [6, 9, 12, 14, 33, 49, 55, 61, 63], it aims to develop even more lightweight models that balance resource efficiency with detection performance, enabling practical deployment of AI-powered security solutions in real-world IoT networks.

3 PROCEDURES AND METHODS

To design a Lightweight AI-IDS optimised for deployment on resource-constrained devices, this project will follow a modular and iterative design methodology. The project will involve the development, optimisation, and evaluation of multiple AI models through a structured development pipeline to ensure consistency across models, facilitate comparative evaluation, and align with the computational constraints of edge environments.

3.1 Dataset Acquisition and Preprocessing

The initial phase involves sourcing a publicly available benchmark dataset for training, testing, and validation. The project will utilise the IoT-23 dataset [23], a new collection of network traffic from IoT devices comprising of twenty malware captures and three benign traffic captures stored in pcap files. This dataset provides real, labelled IoT malware and benign traffic designed for machine learning (ML) algorithm development, making it suitable for the LAIDS project [23]. Its selection is based on its IoT-specific network traffic, aligning with the project's lightweight requirements. The availability of both PCAP files and lighter Zeek logs also offer flexibility for experimentation.

The IoT-23 dataset will undergo standard preprocessing: data normalisation, categorical feature encoding for handling non-numeric data such as the Protocol categorical value, and handling of missing or anomalous data, which are common practices in lightweight IDS research [22, 29, 38, 54]. The label distribution, particularly the prevalence of PartOfAHorizontalPortScan flows, indicates a class imbalance requiring handling, which will be addressed through oversampling [1, 11]. The temporal nature of models like GRU and LSTM will be addressed by treating network flows as sequences or using time-based features. The dataset includes network features categorised into flow features, basic features, time features, and content features [19]. Relevant features will be selected from the dataset such as Port Numbers, Protocol, and Packet Size. New features may potentially be engineered to enhance model performance and efficiency [5, 45, 48]. To ensure robust model development and evaluation, the IoT-23 dataset will be divided into training, validation, and testing sets using a 70/15/15 split.

3.2 Model Development

3.2.1 Baseline Model.

Building on the success of recent studies [10, 27, 31], this project will develop a baseline CNN-based IDS incorporating the TinyML framework which is tailored for deploying ML models on embedded hardware [27]. TinyML's lightweight architecture makes it an ideal benchmark for performance under limited computational resources [30]. The model will serve as a reference point for detection accuracy, model size, inference latency, and overall resource consumption, against which all subsequent models will be compared.

3.2.2 CNN-GRU Model.

The project will implement a hybrid CNN-GRU IDS. The IDS will combine spatial extraction via the CNN, and temporal sequence modelling via the GRU to detect both localised anomalies and sequential attacks.

CNNs are well suited to recognising spatial features in data [43, 59]. The IDS will use 1D CNN layers to extract spatial features and infer patterns from network packets to detect localised threats, outputting a high-level feature map. These outputs will then be pooled and normalised before being passed to the GRU to model temporal patterns in the network traffic. GRUs are often chosen over LSTM models as they are more computationally efficient and require less memory due to their simplified gating mechanisms [39].

3.2.3 GAN Model.

The project will implement a GAN-based IDS. The GAN will undergo unsupervised training using normal traffic, improving the generator's output's realism and, in turn, the discriminator's pattern reignition. This approach allows for zero-day attack detection without relying on known attack signatures [43]. When deployed, any deviations from normal network traffic will be detected by the discriminator. The discriminator will also include an auxiliary classifier categorise the traffic into types of attacks based on the extracted features. Only the discriminator will be deployed to the network, reducing the size of the IDS. Further optimisations can be applied through techniques such as pruning, quantisation and activation functions such as LeakyReLu [35].

3.2.4 PCA-CNN Model.

This project will implement a hybrid CNN-PCA based IDS. PCA will first reduce the dimensionality of the input data to minimise noise and lower the CNN's resource demands [65]. The transformed data will then be classified by the CNN as either malicious or benign. This combination allows PCA to handle basic feature extraction, enabling the CNN to focus on complex pattern recognition, improving both performance and efficiency [36, 64]. PCA's limitation becomes apparent in capturing non-linear patterns [46, 64]. If needed, Kernel PCA (KPCA) will be explored as a non-linear alternative [14].

3.2.5 Autoencoder Model.

This project will implement an autoencoder based IDS. The AE will be trained solely on unlabelled normal network traffic. The AE's encoder component will perform dimensionality reduction, compressing the given preprocessed input data and the decoder component will then reconstruct the encoder's output data back into it's original form [4]. When the model is exposed to malicious traffic, the decoder is will poorly reconstruct the encoder's output data resulting in a high reconstruction error. This error will indicate that the network traffic may be malicious [49].

Once developed, further optimisation techniques such as quantisation, clustering and pruning shall be applied to improve the efficiency of the model.

3.2.6 CNN-LSTM Model.

The CNN layers will capture spatial hierarchies and local patterns in the traffic data. The LSTM layers will process the resulting feature maps over time to model temporal dependencies. The network traffic data will be converted into sequences suitable for time-series analysis. These will be passed through 1D convolutional layers for feature extraction. The resulting output will be passed through LSTM layers which will model temporal dependencies [26]. A final dense layer will classify the traffic as benign or malicious.

The hybrid model will be subjected to the same optimisation and evaluation procedures, including pruning, quantisation and deployment on a Mini PC.

3.2.7 DBN Model.

The training process will follow a two-phase approach. First, each RBM will undergo unsupervised pre-training to reconstruct its input and pass meaningful representations forward, followed by supervised fine-tuning using back-propagation and stochastic gradient descent [15].

The implementation will need to include thorough data preprocessing involving normalisation and PCA to reduce dimensionality [13]. The DBN architecture will mirror Alom et al's [8], using a five-layer RBM stack followed by a softmax output layer for multi-class classification [15]. The five-layer depth was proven to be effective in learning increasingly complex representations of network traffic data, but the number of layers will potentially be tuned during the experimentation phase to find an efficient balance.

Post-training, the model will be optimised via pruning, quantisation, and hyperparameter tuning in line with the procedures applied to other models in the project.

3.3 Model Training and Evaluation

Each model will be trained, tested, and validated independently, with their performance evaluated across a range of metrics such as memory usage, model size, accuracy, precision, false negative rate, and inference latency. Given the project's emphasis on developing a lightweight IDS, resource-related metrics will be the focus when assessing each model's suitability for deployment in constrained environments. The false negative rate will also be closely monitored, as it is one of the primary indicators of the models' effectiveness in detecting threats. In addition to evaluating the models' ability to distinguish between benign and malicious network traffic, their effectiveness in identifying specific types of malicious activity will also be examined. This added interpretability is useful for providing more actionable security insights compared to traditional binary classification.

3.4 Model Optimisation

After the initial training and evaluation phase, each model will undergo targeted optimisation to enhance their suitability for resource-constrained environments. Our aim is to improve resource utilisation without significantly compromising performance. These procedures will focus on reducing computational overhead and model size. One key technique to be explored is model pruning, which aims to eliminate redundant connections within the trained neural networks, thereby decreasing the model's memory footprint and inference time. Both structured pruning, removing entire neurons, and unstructured weight pruning, eliminating less significant individual weights, will be explored [34, 57]. Another critical optimisation method is quantisation, which reduces the numerical precision of the model's weights and activations. By using lower-bit representations, significant reductions of memory usage can be achieved as well as accelerated computations, as demonstrated by the Quantised Autoencoder developed by Sharmila and Nagapadma [49].

The project will also investigate parameter tuning. Techniques such as the SPO method used by Aljehane et al. [7] will be considered to fine-tune hyper-parameters for optimal performance within the resource constraints. Additionally, offline knowledge distillation will be explored as a means of compressing larger, potentially more accurate models into smaller, more efficient ones by transferring their learned knowledge [57, 60]. Building upon recent advancements in the optimisation of TinyML models, this study will assess the potential for further improvements of the baseline by combining techniques such as pruning, quantisation, and knowledge distillation [55, 56] to achieve a lightweight, effective AI-IDS suitable for resource-constrained devices.

3.5 Model Testing and Evaluation

Once optimised the models will be converted usising TensorFlow Lite and integrated into a Python-based IDS pipeline deployed on a Mini PC. The Mini PC will be connected to an isolated virtual network, this setup will simulate a real-world edge device providing a realistic context to evaluate responsiveness, memory consumption, and detection performance under constrained conditions.

The IDS will be tested within this virtual network where it will analyse traffic from the IoT-23 dataset, with its performance in detecting and classifying intrusions continuously monitored. To support testing, a virtual Software Defined Network will be set up using MiniNet to simulate the network topology and the ONOS controller to dynamically configure flow rules. Open vSwitch bridges will direct traffic through monitored ports to the IDS on the Mini PC. TCP Replay will inject public PCAP datasets into the network, maintaining original packet timing and structure to simulate realistic traffic.

Key evaluation metrics include CPU usage, memory footprint, accuracy, and latency to ensure the system operates effectively in resource-constrained environments. The system must also provide real-time intrusion detection while and delivering reliable outputs to aid in intrusion diagnosis.

4 ETHICAL, PROFESSIONAL AND LEGAL ISSUES

4.1 Legal Considerations

The LAIDS project anticipates no direct legal issues as it avoids human data in AI model development. Compliance with privacy regulations like POPIA is ensured by using monitored network data only with explicit organisational consent. The project will use the publicly available IoT-23 dataset to avoid any legal concerns with processing sensitive information. All outputs (code, models, documentation) will be open-source to promote transparency and lawful distribution.

4.2 Ethics clearance

Ethical clearance is not required as this project does not involve human participants or sensitive personal data. The primary ethical consideration, dataset usage, is addressed by using the thoroughly anonymised and ethically sound, IoT-23 dataset which has been used widely in research. All data processing occurs within isolated virtual networks on local machines, preventing harm to real-world systems. The AI development will not involve profiling nor will it make generalisations based on the origin of network traffic within the anonymised dataset. By using only publicly available, anonymised data, the AI-IDS operates without interacting with real individuals or live systems, mitigating common AI ethical concerns.

5 ANTICIPATED OUTCOMES

5.1 System

The final system outcome will be a functional, lightweight and modular LAIDS. Its modular architecture will allow for the separation of preprocessing and classification components for improved flexibility, maintainability, and targeted optimisation. The system will monitor network traffic in real time, classifying benign and malicious activity with minimal resource usage.

5.2 Research

The research aims to evaluate which model architectures are the most resource-efficient under the defined constraints and identify the most favourable trade-off between performance and computational efficiency. The project will also investigate how certain optimisation techniques affect the overall system resource requirements. Finally, through a comparison of the proposed models against the baseline model, the research aims to answer which model would be most suitable LAIDS.

5.3 Expected impact

The project aims to prove that AI-powered IDS solutions can be effective and accessible in low-resource environments. A successful Mini PC deployment would underscore the feasibility of practical, low-resource cybersecurity solutions. This could benefit communities with limited computational infrastructure, such as rural or underserved areas in South Africa. The final system could also serve as a benchmark for future lightweight AI-IDS implementations in academic and industry settings.

5.4 Key success factors

Success will be measured by the system's ability to meet the defined resource constraints while operating in real-time and achieving both a high detection rate and a low false negative rate. Another key success factor is the ability to provide a clear analysis and justification of the trade-offs made during model selection and optimisation. This includes a comparison based on the evaluation metrics outlined across different models, illustrating how decisions regarding architecture, preprocessing, and tuning influenced the balance between efficiency and accuracy.

6 PROJECT PLAN

6.1 Risks and Risk Management

See Appendix A: Project Risks and Management Strategies

6.2 Timeline

See Appendix B: Gantt Chart

6.3 Resources Required

People

The LAIDS project team will consist of Christopher Blignaut, Sian Caine and Claire Campbell. They will be supervised by Dr Josiah Chavula.

Software and Hardware Equipment

The project will use Google Colab for model prototyping, training, and code sharing via a cloud-based Jupyter interface. Models will be developed in Python using TensorFlow/Keras, and additional workflows will run on local machines with at least 8GB RAM. SmartSDN will integrate software-defined networking (SDN) with ML to capture packets via Open vSwitch for input to the models. To simulate deployment in constrained environments, the system will also run on a Mini PC.

6.4 Deliverables

The project will generate several key deliverables throughout its lifecycle. These include: literature reviews, a project proposal, a demonstration showcasing running prototypes, a final paper detailing research findings, the final project code, a poster summarising research findings, and a website for updates and documentation. The project will culminate in a final demonstration providing a comprehensive overview of the completed system and participation in the UCT School of IT Showcase.

6.5 Milestones

The project's progression is marked by several key stages. These stages are marked with milestones including: completing literature reviews, drafting, presenting and revising of the project proposal, and the building, training and testing of both the baseline and proposed models culminating in a project progress demonstration. After which the final milestones include: evaluation and comparisons of the models, completing the final paper and final project code, showcasing our completed system in a final project demonstration, preparation of a project poster and website and participation in the School of IT Showcase.

6.6 Work Allocation

The LAIDS team will collaboratively acquire and preprocess datasets and jointly develop and evaluate a baseline model. Each member will address one research question by developing, evaluating and implementing two IDS models: Sian will build a PCA-CNN and an Autoencoder model; Claire will create a CNN-LSTM and a DBN model; Christopher will develop a CNN-GRU and a GAN model. The team will also build a virtual SDN for model testing.

REFERENCES

[1] Ahmed Abdelkhalek and Maggie Mashaly. 2023. Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning. *The journal of Supercomputing* 79, 10 (2023), 10611–10644.

[2] Razan Abdulhammed, Miad Faezipour, Hassan Musafer, and Abdelshakour Abuzneid. 2019. Efficient Network Intrusion Detection Using PCA-Based Dimensionality Reduction of Features. *2019 International Symposium on Networks, Computers and Communications (ISNCC)* (06 2019). https://doi.org/10.1109/isncc.2019.8909140

[3] Moorthy Agoramoorthy, Ahamed Ali, D. Sujatha, Michael Raj. T F, and G. Ramesh. 2023. An Analysis of Signature-Based Components in Hybrid Intrusion Detection Systems. In *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*. 1–5. https://doi.org/10.1109/ICCEBS58601.2023.10449209

[4] Ghada AL Mukhaini, Mohammed Anbar, Selvakumar Manickam, Taief Alaa Al-Amiedy, and Ammar Al Momani. 2023. A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks. *Journal of King Saud University - Computer and Information Sciences* 36 (11 2023), 101866–101866. https://doi.org/10.1016/j.jksuci.2023.101866

[5] Mohammed M Alani and Ali Miri. 2022. Towards an explainable universal feature set for IoT intrusion detection. *Sensors* 22, 15 (2022), 5690.

[6] Tarek Ali, Amna Eleyan, Tarek Bejaoui, and Mohammed Al-Khalidi. 2024. Lightweight Intrusion Detection System with GAN-Based Knowledge Distillation. *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)* (05 2024), 1–7. https://doi.org/10.1109/smartnets61466.2024.10577682

[7] Nojood O Aljehane, Hanan A Mengash, Siwar BH Hassine, Faiz A Alotaibi, Ahmed S Salama, and Sitelbanat Abdelbagi. 2024. Optimizing intrusion detection using intelligent feature selection with machine learning model. *Alexandria Engineering Journal* 91 (2024), 39–49.

[8] Md Zahangir Alom, VenkataRamesh Bontupalli, and Tarek M Taha. 2015. Intrusion detection using deep belief networks. In *2015 National Aerospace and Electronics Conference (NAECON)*. IEEE, 339–344.

[9] Hakan Can Altunay and Zafer Albayrak. 2023. A hybrid CNN+LSTM-based intrusion detection system for industrial IoT networks. *Engineering Science and Technology, an International Journal* 38 (02 2023), 101322. https://doi.org/10.1016/j.jestch.2022.101322

[10] Mattia Antonini, Miguel Pincheira, Massimo Vecchio, and Fabio Antonelli. 2023. An adaptable and unsupervised TinyML anomaly detection system for extreme industrial environments. *Sensors* 23, 4 (2023), 2344.

[11] Sikha Bagui and Kunqi Li. 2021. Resampling imbalanced data for network intrusion detection datasets. *Journal of Big Data* 8, 1 (2021), 6.

[12] Nagaraj Balakrishnan, Arunkumar Rajendran, Danilo Pelusi, and Vijayakumar Ponnusamy. 2019. Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things. *Internet of Things* 14 (09 2019), 100112. https://doi.org/10.1016/j.iot.2019.100112

[13] Nagaraj Balakrishnan, Arunkumar Rajendran, Danilo Pelusi, and Vijayakumar Ponnusamy. 2021. Deep Belief Network enhanced intrusion detection system to prevent security breach in the Internet of Things. *Internet of things* 14 (2021), 100112.

[14] Awotunde Bamidele, Ranjit Panigrahi, Biswajit Brahma, and Akash Kumar Bhoi. 2024. CNN-KPCA: A hybrid Convolutional Neural Network with Kernel Principal Component Analysis for Intrusion Detection System for the Internet of Things Environments. *Information Technology and Nanotechnology* Vol-3584 (02 2024), 74–83.

[15] Othmane Belarbi, Aftab Khan, Pietro Carnelli, and Theodoros Spyridopoulos. 2022. An intrusion detection system based on deep belief networks. In *International Conference on Science of Cyber Security*. Springer, 377–392.

[16] Bo Cao, Chenghai Li, Yafei Song, Yueyi Qin, and Chen Chen. 2022. Network intrusion detection model based on CNN and GRU. *Applied Sciences* 12, 9 (2022), 4184.

[17] Swathi Ch and Suresh Babu Kare. 2024. A Comprehensive Analysis of Network Intrusion Detection in Internet of Things and Wireless Networks. In *2024 International Conference on Data Science and Network Security (ICDSNS)*. 01–05. https://doi.org/10.1109/ICDSNS62112.2024.10691047

[18] Jiawei Du, Kai Yang, Yanjing Hu, and Lingjie Jiang. 2023. NIDS-CNNLSTM: Network intrusion detection classification model based on deep learning. *IEEE Access* 11 (2023), 24808–24821.

[19] Vibekananda Dutta, Michal Choras, Marek Pawlicki, and Rafal Kozik. 2020. Detection of Cyberattacks Traces in IoT Data. *J. Univers. Comput. Sci.* 26, 11 (2020), 1422–1434.

[20] Alexander G Eustis. 2019. The Mirai Botnet and the importance of IoT device security. In *16th International Conference on Information Technology-New Generations (ITNG 2019)*. Springer, 85–89.

[21] Kamil Faber, Lukasz Faber, and Bartlomiej Sniezynski. 2021. Autoencoder-based IDS for cloud and mobile devices. In *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 728–736. https://doi.org/10.1109/CCGrid51090.2021.00088

[22] Samir Fenanir, Fouzi Semchedine, and Abderrahmane Baadache. 2019. A Machine Learning-Based Lightweight Intrusion Detection System for the Internet of Things. *Revue d'Intelligence Artificielle* 33, 3 (2019).

[23] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. [n. d.]. *IoT-23: A labeled dataset with malicious and benign IoT network traffic*. https://doi.org/10.5281/zenodo.4743746

[24] Mohammed Sayeeduddin Habeeb and T. Ranga Babu. 2022. Network intrusion detection system: A survey on artificial intelligence-based techniques. *Expert Systems* 39, 9 (2022), e13066. https://doi.org/10.1111/exsy.13066 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/exsy.13066

[25] Mohammed Sayeeduddin Habeeb and T Ranga Babu. 2022. Network intrusion detection system: a survey on artificial intelligence-based techniques. *Expert*

Systems 39, 9 (2022), e13066.

[26] Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. 2022. CNN-LSTM: hybrid deep neural network for network intrusion detection system. *IEEE Access* 10 (2022), 99837–99849.

[27] Hyungchul Im and Seongsoo Lee. 2024. TinyML-Based Intrusion Detection System for In-Vehicle Network Using Convolutional Neural Network on Embedded Devices. *IEEE Embedded Systems Letters* (2024).

[28] Riku Immonen and Timo Hämäläinen. 2022. Tiny Machine Learning for Resource-Constrained Microcontrollers. *Journal of Sensors* 2022, 1 (2022), 7437023.

[29] Sana Ullah Jan, Saeed Ahmed, Vladimir Shakhov, and Insoo Koo. 2019. Toward a lightweight intrusion detection system for the internet of things. *IEEE access* 7 (2019), 42450–42471.

[30] Abbas Javed, Amna Ehtsham, Muhammad Jawad, Muhammad Naeem Awais, Ayyaz-ul-Haq Qureshi, and Hadi Larijani. 2024. Implementation of lightweight machine learning-based intrusion detection system on IoT devices of smart homes. *Future Internet* 16, 6 (2024), 200.

[31] Iyad Katib, Emad Albassam, Sanaa A Sharaf, and Mahmoud Ragab. 2025. Safeguarding IoT consumer devices: Deep learning with TinyML driven real-time anomaly detection for predictive maintenance. *Ain Shams Engineering Journal* 16, 2 (2025), 103281.

[32] K. Keerthi Vasan and B. Surendiran. 2016. Dimensionality Reduction Using Principal Component Analysis for Network Intrusion Detection. *Perspectives in Science* 8 (09 2016), 510–512. https://doi.org/10.1016/j.pisc.2016.05.010

[33] Dusmurod Kilichev, Dilmurod Turimov, and Wooseong Kim. 2024. Next–generation intrusion detection for iot evcs: Integrating cnn, lstm, and gru models. *Mathematics* 12, 4 (2024), 571.

[34] Mingjian Lei, Xiaoyong Li, Binsi Cai, Yunfeng Li, Limengwei Liu, and Wenping Kong. 2020. P-DNN: An effective intrusion detection method based on pruning deep neural network. In *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[35] Stamatis Mastromichalakis. 2020. ALReLU: A different approach on Leaky ReLU activation function to improve Neural Networks Performance. *arXiv preprint arXiv:2012.07564* (2020).

[36] Amir Mehrabinezhad, Mohammad Teshnehlab, and Arash Sharifi. 2024. A comparative study to examine principal component analysis and kernel principal component analysis-based weighting layer for convolutional neural networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization* 12 (07 2024). https://doi.org/10.1080/21681163.2024.2379526

[37] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. 2018. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089* (2018).

[38] Xuan-Ha Nguyen, Xuan-Duong Nguyen, Hoang-Hai Huynh, and Kim-Hung Le. 2022. Realguard: A lightweight network intrusion detection system for IoT gateways. *Sensors* 22, 2 (2022), 432.

[39] Ogobuchi Daniel Okey, Demostenes Zegarra Rodriguez, and João Henrique Kleinschmidt. 2024. Enhancing IoT Intrusion Detection with Federated Learning-Based CNN-GRU and LSTM-GRU Ensembles. In *2024 19th International Symposium on Wireless Communication Systems (ISWCS)*. 1–6. https://doi.org/10.1109/ISWCS61526.2024.10639159

[40] Uneneibotejit Otokwala, Andrei Petrovski, and Harsha Kalutarage. 2024. Optimized common features selection and deep-autoencoder (OCFSDA) for lightweight intrusion detection in Internet of things. *International Journal of Information Security* 23, 4 (2024), 2559–2581.

[41] Cheolhee Park, Jonghoon Lee, Youngsoo Kim, Jong-Geun Park, Hyunjin Kim, and Dowon Hong. 2023. An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. *IEEE Internet of Things Journal* 10, 3 (2023), 2330–2345. https://doi.org/10.1109/JIOT.2022.3211346

[42] Vignesh Reddy, Sunitha R, M. Anusha, S Chaitra, and Abhilasha P Kumar. 2024. Artificial Intelligence Based Intrusion Detection Systems. In *2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)*. 1–6. https://doi.org/10.1109/ICMNWC63764.2024.10872055

[43] Vignesh Reddy, Sunitha R, M. Anusha, S Chaitra, and Abhilasha P Kumar. 2024. Artificial Intelligence Based Intrusion Detection Systems. *2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC)* (12 2024), 1–6. https://doi.org/10.1109/icmnwc63764.2024.10872055

[44] Bipraneel Roy and Hon Cheung. 2018. A deep learning approach for intrusion detection in internet of things using bi-directional long short-term memory recurrent neural network. In *2018 28th international telecommunication networks and applications conference (ITNAC)*. IEEE, 1–6.

[45] Amiya Kumar Sahu, Suraj Sharma, Mohammad Tanveer, and Rohit Raja. 2021. Internet of Things attack detection using hybrid Deep Learning Model. *Computer Communications* 176 (2021), 146–154.

[46] T. Saranya, S. Sridevi, C. Deisy, Tran Duc Chung, and M.K.A.Ahamed Khan. 2020. Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review. *Procedia Computer Science* 171 (2020), 1251–1260. https://doi.org/10.1016/j.procs.2020.04.133

[47] M.H.U. Sharif and M.A. Mohammed. 2022. A literature review of financial losses statistics for cyber security and future trend. *World Journal of Advanced Research and Reviews* 15, 1 (2022), 138–156. https://doi.org/10.30574/wjarr.2022.15.1.0573

[48] Taveesh Sharma. 2023. Investigating optimal internet data collection in low resource networks. http://hdl.handle.net/11427/38141

[49] B S Sharmila and Rohini Nagapadma. 2023. Quantized autoencoder (QAE) intrusion detection system for anomaly detection in resource-constrained IoT devices using RT-IoT2022 dataset. *Cybersecurity* 6 (09 2023). https://doi.org/10.1186/s42400-023-00178-5

[50] T. Sowmya and E.A. Mary Anita. 2023. A comprehensive review of AI based intrusion detection system. *ScienceDirect* 28 (06 2023), 100827–100827. https://doi.org/10.1016/j.measen.2023.100827

[51] T. Sowmya and E.A. Mary Anita. 2023. A comprehensive review of AI based intrusion detection system. *Measurement: Sensors* 28 (2023), 100827. https://doi.org/10.1016/j.measen.2023.100827

[52] Ralf C Staudemeyer and Eric Rothstein Morris. 2019. Understanding LSTM–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586* (2019).

[53] Basant Subba, Santosh Biswas, and Sushanta Karmakar. 2016. Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis. *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* (11 2016). https://doi.org/10.1109/ants.2016.7947776

[54] Belal Sudqi Khater, Ainuddin Wahid Bin Abdul Wahab, Mohd Yamani Idna Bin Idris, Mohammed Abdulla Hussain, and Ashraf Ahmed Ibrahim. 2019. A lightweight perceptron-based intrusion detection system for fog computing. *applied sciences* 9, 1 (2019), 178.

[55] Bin Sun and Yu Zhao. 2024. TinyNIDS:CNN-Based Network Intrusion Detection System on TinyML Models in 6G Environments. *Internet Technology Letters* (12 2024). https://doi.org/10.1002/itl2.629

[56] Thanaphon Suwannaphong, Ferdian Jovan, Ian Craddock, and Ryan McConville. 2025. Optimising TinyML with quantization and distillation of transformer and mamba models for indoor localisation on edge devices. *Scientific Reports* 15, 1 (2025), 10081.

[57] Sunil Vadera and Salem Ameen. 2022. Methods for pruning deep neural networks. *IEEE Access* 10 (2022), 63280–63300.

[58] Patrick Vanin, Thomas Newe, Lubna Luxmi Dhirani, Eoin O'Connell, Donna O'Shea, Brian Lee, and Muzaffar Rao. 2022. A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning. *Applied Sciences* 12 (01 2022), 11752. https://doi.org/10.3390/app122211752

[59] Patrick Vanin, Thomas Newe, Lubna Luxmi Dhirani, Eoin O'Connell, Donna O'Shea, Brian Lee, and Muzaffar Rao. 2022. A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning. *Applied Sciences* 12, 22 (2022), 11752. https://doi.org/10.3390/app122211752

[60] Zhendong Wang, Zeyu Li, Daojing He, and Sammy Chan. 2022. A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning. *Expert Systems with Applications* 206 (2022), 117671.

[61] Peng Wei, Yufeng Li, Zhen Zhang, Tao Hu, Ziyong Li, and Diyang Liu. 2019. An optimization method for intrusion detection classification model based on deep belief network. *Ieee Access* 7 (2019), 87593–87605.

[62] Chuanlong Yin, Yuefei Zhu, Jinlong Fei, and Xinzheng He. 2017. A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access* 5 (10 2017), 21954–21961. https://doi.org/10.1109/access.2017.2762418

[63] S Zargar. 2021. Introduction to sequence learning models: RNN, LSTM, GRU. *Department of Mechanical and Aerospace Engineering, North Carolina State University* (2021).

[64] Ruijie Zhao, Guan Gui, Zhi Xue, Jie Yin, Tomoaki Ohtsuki, Bamidele Adebisi, and Haris Gacanin. 2021. A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things. *IEEE Internet of Things Journal* 9 (2021), 9960–9972. https://doi.org/10.1109/jiot.2021.3119055

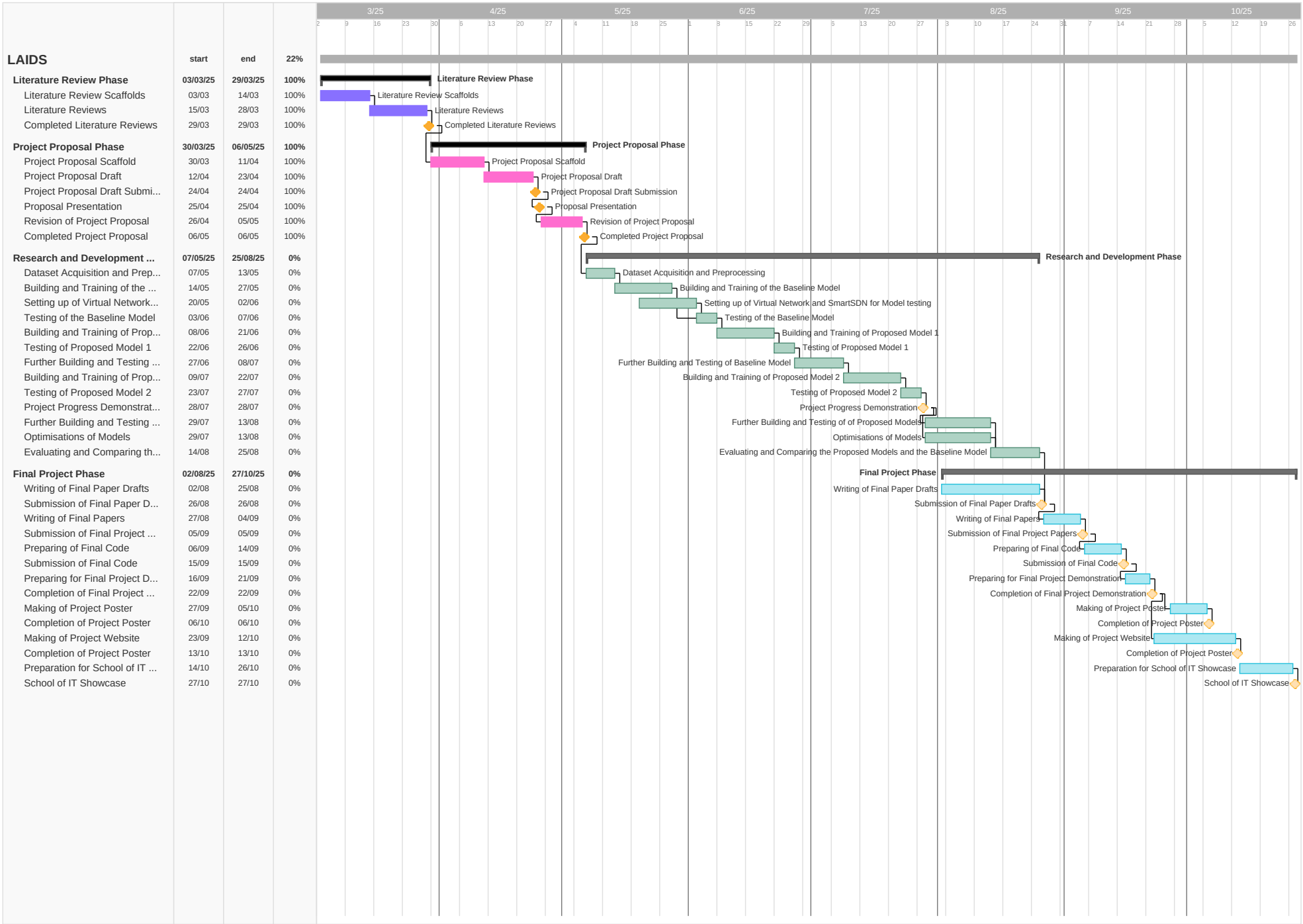
[65] Shengchu Zhao, Wei Li, Tanveer Zia, and Albert Y. Zomaya. 2017. A Dimension Reduction Model and Classifier for Anomaly-Based Intrusion Detection in Internet of Things. *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)* (11 2017). https://doi.org/10.1109/dasc-picom-datacom-cyberscitec.2017.141

APPENDIX A: PROJECT RISKS AND MANAGEMENT STRATEGIES

Risk	Impact	Likelihood	Mitigation Strategy
Insufficient IoT-specific traffic in datasets	High	Medium	Supplement datasets with synthetic IoT traffic. Use IoT-focused datasets such as BoT-IoT or IoT-23.
Limited computing power of our team's devices	Medium	Medium	Use Google Colab for ML and DL tasks.
Model underfitting: the model is too simple to capture the underlying patterns in the network traffic data.	High	Medium	Decrease regularisation. Increase the amount of training data. Add complexity to the model through additional neurons.
Model overfitting: the model cannot generalise the training data and instead learns it too well, including noise and irrelevant details.	High	Medium	Increase the amount of training data. Use early stopping. Simplify the model with pruning. Apply feature selection and ensemble techniques.
Inadequate model performance	High	High	Apply optimisation techniques. Tune hyperparameters. Focus on obtaining quality training and testing data. Ensembling
Team coordination challenges	Medium	Low	Set weekly meetings. Maintain clear communication on roles and deliverables.
Misinterpreting the project's objectives	High	Low	Schedule weekly meetings with the supervisor. Clearly define deliverables. Provide regular updates and demonstrations.
Lack of experience with machine learning	High	High	Allocate early project time for focused learning on ML methods. Prototype with lightweight models.

Table 1: Project Risks and Management Strategies

APPENDIX B: GANTT CHART



LAIDS	start	end	22%
Literature Review Phase	**03/03/25**	**29/03/25**	**100%**
Literature Review Scaffolds	03/03	14/03	100%
Literature Reviews	15/03	28/03	100%
Completed Literature Reviews	29/03	29/03	100%
Project Proposal Phase	**30/03/25**	**06/05/25**	**100%**
Project Proposal Scaffold	30/03	11/04	100%
Project Proposal Draft	12/04	23/04	100%
Project Proposal Draft Submi...	24/04	24/04	100%
Proposal Presentation	25/04	25/04	100%
Revision of Project Proposal	26/04	05/05	100%
Completed Project Proposal	06/05	06/05	100%
Research and Development ...	**07/05/25**	**25/08/25**	**0%**
Dataset Acquisition and Prep...	07/05	13/05	0%
Building and Training of the ...	14/05	27/05	0%
Setting up of Virtual Network...	20/05	02/06	0%
Testing of the Baseline Model	03/06	07/06	0%
Building and Training of Prop...	08/06	21/06	0%
Testing of Proposed Model 1	22/06	26/06	0%
Further Building and Testing ...	27/06	08/07	0%
Building and Training of Prop...	09/07	22/07	0%
Testing of Proposed Model 2	23/07	27/07	0%
Project Progress Demonstrat...	28/07	28/07	0%
Further Building and Testing ...	29/07	13/08	0%
Optimisations of Models	29/07	13/08	0%
Evaluating and Comparing th...	14/08	25/08	0%
Final Project Phase	**02/08/25**	**27/10/25**	**0%**
Writing of Final Paper Drafts	02/08	25/08	0%
Submission of Final Paper D...	26/08	26/08	0%
Writing of Final Papers	27/08	04/09	0%
Submission of Final Project ...	05/09	05/09	0%
Preparing of Final Code	06/09	14/09	0%
Submission of Final Code	15/09	15/09	0%
Preparing for Final Project D...	16/09	21/09	0%
Completion of Final Project ...	22/09	22/09	0%
Making of Project Poster	27/09	05/10	0%
Completion of Project Poster	06/10	06/10	0%
Making of Project Website	23/09	12/10	0%
Completion of Project Poster	13/10	13/10	0%
Preparation for School of IT ...	14/10	26/10	0%
School of IT Showcase	27/10	27/10	0%