Lightweight AI based Intrusion Detection Systems (LAIDS) for Resource-Constrained Networks

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

ABSTRACT

This literature review investigates the feasibility of implementing lightweight AI models into intrusion detection systems that will be deployed in resource-constrained networks. The evolution of technology and the growing adoption of IoT devices has helped further propel the cyber attack landscape. Traditional intrusion detection systems have become inefficiently in detecting anomalies in network traffic. Furthermore they are too computationally expensive to implement in resource-constraint networks. AI models are able to achieve higher accuracy in detecting novel threats and adapt to the current dynamic threat landscapes. In particular, deep learning models yield the best performance results in comparison to traditional methods and other machine learning models however, they are also the most computationally intense models. This makes deep learning models unsuitable for deployment in IDSs that work in low-resource networks. Efforts have been made to development lightweight DL models and Hybrid ML-DL models that can maintain DL model precision and accuracy, while minimizing resource use. This allows them to effectively function in resource constrained environments.

KEYWORDS

Intrusion Detection Systems, Artificial Intelligence, Machine Learning, Deep Learning, IoT, Networks,

1 INTRODUCTION

The continuous advancement of digital systems and the increased adoption rate of Internet of Things (IoT) devices has contributed to the rise of cybersecurity threats across networks [10]. Low-resource networks, networks built with limited capacity links and computing devices, are particularly vulnerable to these threats [15]. To mitigate security risks in networks, Intrusion Detection System (IDS) frameworks are implemented to monitor network traffic and flag any suspicious activity [27]. However, the evolution of malicious network activity has made it increasingly difficult for these systems to identify threats [6]. Traditional IDSs that are signature-based or anomaly-based are effective in traditional networks but often require a great amount of computational power to be efficient [19]. This characteristic is undesirable for IDSs implemented in resource-constrained networks. Limitations of traditional IDSs have led to the development of more advanced, Artificial intelligence (AI) based intrusion detection systems. AI based IDSs offer improved accuracy in detecting novel threats as well as adaptability to dynamic threat landscapes [11]. Despite these advantageous qualities, AI based IDSs face concerns surrounding resource usage and complexity [11]. AI based IDSs that make use of machine learning or deep learning models come with high computational costs, which pose a challenge for deployment in low-resource environments [19].

This literature review will investigate the feasibility of implementing a lightweight AI model into an intrusion detection system. By developing a lightweight AI based IDS, resource-constrained networks can benefit from an improved identification of novel threats and attack patterns while minimizing computational overhead costs.

The role of this literature review is to identify problem areas, developments and breakthroughs in IDS frameworks. The findings of this review will provide the foundations on which a lightweight AI IDS (LAIDS) can be designed and developed on.

The scope of this review will focus on academic papers that look at different implementations of intrusion detection systems and their suitability for low-resource networks. AI based IDSs will be the primary focus of discussion. Academic literature published in the last five to ten years will be the main sources considered for this review.

First, Traditional and AI based IDS methods will be explored, particularly looking into deep learning models. Next, existing datasets and evaluation metrics will be analysed. This will be followed by a comparison of traditional IDSs with AI based IDSs as well as Machine learning (ML) IDSs compared with Deep learning (DL) IDSs. The feasibility of a lightweight DL or ML-DL hybrid IDS will be explored, along with the implications of the reviewed literature for the LAIDS project. Finally, conclusions will be drawn from the findings and discussions of this review and a way forward for the LAIDS project will be outlined.

2 TRADITIONAL INTRUSION DETECTION SYSTEMS

Traditional IDS approaches can be classified into signature-based detection and anomaly-based detection.

2.1 Signature-based IDS

Signature-based intrusion detection systems (SIDS) analyses network activities for known threats using predefined patterns or identifiers known as signatures [17]. Signatures are stored in a database that the IDS accesses and contrasts with incoming network traffic [6]. If a match is made between a predefined signature and incoming activity, the activity is flagged by the IDS as suspicious [6]. These signatures are usually developed using string-searching algorithms [17].

Signature-based IDSs have proven to be highly effective at detecting known attack signatures and are able to do so with a relatively low false positive rate [10]. Traditional IDSs typically demand significant computational resources to operate effectively [19]. However,

researchers have developed lightweight signature-based IDSs that can be deployed in resource-constrained environments [19]. S. Tripathi and R. Kumar [22] as well as P. A. M. Hambali et al. [4] proposed lightweight signature-based IDSs that utilised snort, a lightweight SIDS [17], on Raspberry Pi devices to analyse network traffic [19]. These SIDS were able to run on these resource constrained devices efficiently however, they were unable to detect novel threats [19]. A major drawback of SIDSs are their inability to detect unknown attacks or new variants of known attacks [19]. Since their signature databases require constant updates to include the latest attacks, signature-based IDSs struggle to remain effective, especially against evolving threats that span multiple packets or manifest as zero-day attacks [6]. Anomaly-based IDSs have been introduced as a potential solution to SIDS constraints as they classify threat activity based on behaviour instead of signatures [6].

2.2 Anomaly-based IDS

Anomaly-based IDSs (AIDS) utilise techniques such as statical models, machine learning or knowledge-based methods to detect threats in network traffic [6]. An AIDS discovers attacks through identifying network activities that deviate from predefined normal activity patterns on its network [6]. This predefined set of normal patterns is known as the baseline [25]. Any activity that doesn't match the baseline is flagged as suspicious [25]. For example, if there is an unexpected increase in network activity in the middle of the night and the baseline states that there should little to no activity, then the IDS will raise an alarm [25]. This anomaly detection method allows for detection of known and unknown attacks, including zero-day attacks [25][6]. Building and maintaining these anomaly-based IDSs come with high computational overhead costs that are unsuitable for low-resourced networks [10]. Furthermore, anomaly-based IDSs are prone to high false positive rates due to inherent difficulty in defining a baseline for normal network behaviour [25]. For example, changes in network traffic behaviour may be flagged as a potential threats, even if it's legitimate activity [6]. A statistical-based AIDS uses distribution based models to map normal network traffic behaviour [6]. It is simple to build and allows real-time detection however, it outputs subpar accuracy rates [6]. Knowledge-based AIDS have a lower false positive rate as they create a knowledge base that contains the profiles of all normal traffic in a network. However, this knowledge base needs constant updating, making knowledge-based AIDS difficult to maintain in a changing cyber attack environment [6]. Machine learning AIDS will be looked at in the next section of this review.

3 AI BASED INTRUSION DETECTION SYSTEMS

AI based IDSs have shown to be a promising alternative to traditional intrusion detection systems. Their adaptability and enhanced detection accuracy make them particularly effective in dynamic and evolving threat environments [11]. AI based IDSs can be classified into Machine learning (ML) based IDSs and Deep learning (DL) based IDSs. These categories can be further segmented into various other models, as seen in Figure 1.

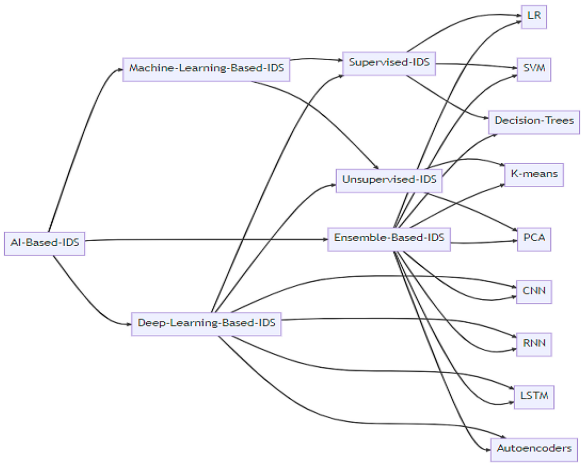

Figure 1: Classification of AI based IDS [18]

3.1 Machine Learning Based Intrusion Detection Systems

Machine Learning (ML) is a subsection of AI that involves training models using algorithms that learn from historical data [8]. Based on the data, models can enhance their judgments or projections - without being directly programmed to do so [8]. In the context of cybersecurity, machine learning based IDSs utilise models that have been trained on large volumes of past network activity data [8]. This allows machine learning IDSs to improve and adapt their detection abilities to identify known or unknown abnormal network traffic patterns [11].

3.1.1 Supervised Machine Learning Intrusion Detection. Supervised machine learning is the process of training models using algorithms and labelled datasets [25]. These labelled datasets enable models to learn how to map a given input to the correct output [25]. In the case of IDSs, an input would be some sort of network activity, and the output would be whether the activity was an attack or not. Once a model is trained, it can apply the knowledge it has learnt to interpret unseen data [25]. A main advantage of supervised ML models are their ability to refine themselves based on their accuracy rate feedback during the training phase [25]. A drawback in training supervised ML models is that they require large amounts of computational power and precise input data to achieve effective results. This challenge is further magnified when big data is needed in training, as it can be difficult for supervised ML models to decipher [25]. Supervised machine learning models can be segmented into either classification or regression models [18]. Classification models group data points in datasets based on common features while regression models predict continuous outcomes based on the given data [25]. As intrusion detection at its core aims to distinguish between normal and malicious network traffic, we will focus on classification models such as Support Vector Machines, K-Nearest Neighbours, Decision Trees and Random Forests in this review.

Support Vector Machines (SVM)
Support vector machines are classification models that sort data

into different groups based on shared features. They do this by finding an optimal boundary line (a hyperplane) that best separates the data into distinct categories [18]. SVMs have proven to have high accuracy rates and aren't susceptible to overfitting data [11]. SVMs are also able to work with large datasets however, they become more computationally expensive when performing real-time detection on big data [25][11]. Techniques such as dimensionality reduction have been introduced to reduced the computational costs of SVM IDSs. This has increased the feasibility of using SVM IDSs for real-time threat detection in resource-constrained environments [11].

K-Nearest Neighbour (KNN)
K-Nearest neighbour models are widely used in IDSs due to their simplicity and high accuracy [11]. KNNs classify new data points by comparing them to a K number of similar data points that it's already categorised [18]. For example, if K is set to five then a new data point will be compared to five other similar (nearest) data points [18]. The five nearest data points are computed using Euclidean distance measurement [18]. The new data point will then be classified according to what the majority of the other five data points are classified as [18]. In the context of IDSs, if four out of five nearest data points are classified as threats, then so will the new data point. They work best in small to medium sized databases with predictable network activity patterns [11]. Due to the complexity of computing distances between data points, KNNs are unsuitable for real-time intrusion detection in dynamic threat environments [11]. Like SVMs, KNNs are resource intense when handling big data - making them difficult to implement in low-resource networks [11].

Random Forest (RF)
Random forest models are a combination of several Decision Tree (DT) models that can be used for both regression and classification [18]. DTs have a tree-like structure where branches represent decision paths that the model can take and the leaves at the end of the branches are its final output [13]. In comparison to SVMs and KNNS, decision trees have a shorter decision-making computation time however, they are prone to overfitting data [11]. Each decision tree in a RF is trained individually on their own subset of data from a larger database [18][23]. RFs are ensemble models, meaning their prediction is based off of each individual decision tree's prediction [18][23]. A RFs final outcome is determined by the majority vote from all its trees [18][23]. Random Forests have a higher threat-detection accuracy in comparison to a singular DT due to RFs examining the predictions of multiple DTs [11]. Another advantage of RFs are their ability to minimize the overfitting of data that decision trees suffer from [11]. RFs are able to process large datasets however, they bear computational times similar to SVMs and KNNs [25]. That being said, according to research [9], RFs tend to show better accuracy rates than SVMs and KNNs. Further research [23] suggests that RFs can outperform multiple ML models with accuracy and precisions rates higher than 90%. Nevertheless, long computation times make RFs undesirable for real-time intrusion detection in low-resource networks [18].

3.1.2 Unsupervised Machine Learning Intrusion Detection. Unsupervised machine learning is the process of training models using algorithms and unlabelled datasets [25]. The model improves itself through interpreting input data and finding patterns [25]. A key advantage of unsupervised ML is that no human intervention is required to validate its results [25]. It is also capable of discovering patterns in large datasets quicker than a manual analysis [25]. This swift pattern recognition is ideal for detecting novel security threats [25]. Similar to supervised ML models, unsupervised ML models incur high computational costs and additionally, requires large amounts of data for effective training [25]. As there is no human intervention, unsupervised ML models are susceptible to higher risks of inaccurate predictions in comparison to supervised ML models [25].
Unsupervised learning can be categorized into clustering, association and dimensionality reduction algorithms [25]. This review will focus on models that use clustering and dimensionality reduction algorithms. Clustering is the grouping of unlabeled data points based on shared features - a similar concept to KNNs [25]. Dimensionality reduction is the process of reducing features on input data to make it easier to handle [25].

K-Mean Clustering
K-Mean clustering is a popular unsupervised ML algorithm that groups data into a K number of clusters based on similarities between the data points. Initially, random data points called centroids are assigned to the center of each cluster [3]. New data point are then assigned to a cluster based on a similarity measurement calculated using Euclidean distance [2] [14][3]. In the context of IDSs, a clustering algorithm such as k-mean clustering could be used to group network traffic into clusters of abnormal and normal activity [3]. K-means algorithms are relatively simple to build however, a common implementation challenge is determining the optimal number of clusters that will be initialised [6]. Multiple k values often need to be tested before a suitable one is found [6]. The effectiveness of clustering is also heavily dependent on the random initialisation of clusters [3]. According to research [3] randomised cluster centers can lead to the production of empty clusters and slow computation times. While methods exits to improve this dilemma, they come with increased computation costs [3]. K-mean clustering algorithms work best with spherical data (uniformly distributed) [14] - a uncommon quality in network traffic data. These limitations of k-means clustering make it unideal for implementation in IDSs that operate in low-resource networks.

Principal Component Analysis (PCA)
Principal component analysis is a dimensionality reduction technique that has proven to be extremely effective network traffic analysis [5]. It transforms high-dimensional data into a low-dimensional form using linear combinations to generate a new subset of features from the data's original feature set [5]. These new features, known as principal components, are the most significant pieces of information in the data [20] [1]. High dimensional data often creates unnecessary noise as many dimensions are redundant or irrelevant [28]. Noisy data increases a classification model's computation complexity and can diminish their ability to detect intrusions in network traffic in real-time [28]. By reducing dimensionality of input network traffic data, PCAs improve a classification models' intrusion detection accuracy while reducing its processing time [5][20]. PCAs

are only able to perform linear combination procedures [14] however, they are able to do so with a low-complexity level and without requiring extensive training [27]. High computational complexity is a common problem deep learning-based intrusion detection systems face [27]. Utilising a PCA as a pre-processing step in a deep learning IDS, improves the system and enables it to become lightweight and efficient [27]. A PCA's ability to enhance an intrusion detection systems performance, reduce it complexity and its overhead costs makes it a viable implementation option in a low-resource network IDSs.

3.2 Deep learning Intrusion Detection

Deep Learning is a subsection of Machine Learning [18]. It is considered more advanced than traditional machine learning due to its complex architecture [18]. Deep learning architecture consists of multiple layers of interconnected neurons; each layer of neurons processes and refines the output of the neuron layer that precedes them [18]. DL models are able to detect underlying patterns in large databases and have been proven to yield high intrusion detection rates [8]. They require substantial volumes of training data and are computationally expensive to train [8]. In addition, they are complex to build and use a significant amount of computational resources to be effective [8]. Deep Learning models that are popular for IDSs are Convolutional Neural Networks (CNNs), recurrent Neutral Networks (RNNs), Generative Adversarial Networks (GANs) [11].

Convolutional Neural Networks (CNNs)
Convolutional neural networks are supervised deep learning models that consists of an input layer, several hidden layers and an output layer [25]. Within these hidden layers are convolution and pooling layers [13]. The convolution layer identifies significant features in the input data while the pooling layer reduces dimensionality of the data [23]. Due to its architecture, CNNs are commonly used for intrusion detection [13]. CNNs are able to evaluate large amounts of high-dimensional data such as network traffic activity and find underlying patterns [11]. Research [11] has shown that CNNs display high accuracy rates for detection intrusion. Further research [18] suggests that CNNs yield higher accuracy rates for detection intrusion than most DL-based models. There is a concern that CNNs maybe be too computationally intense to operate in a resource-constraint environment such as an IoT network [11]. To solve this issue, a multitude of effective lightweight CNNs models have been designed for implementation into intrusion detection systems in low-resource networks. An example of such is TinyNIDS, a lightweight CNN-based model capable of intrusion detecting in real-time in 6G environments [21].
TinyNIDS employs compression methods such as pruning and quantisation to reduce its size and complexity [21]. A comparison of TinyNIDS against CNN and RNN models revealed that TinyNIDS achieved a slightly higher accuracy rate while maintaining a lower latency than these models[21].
Additionally, Hybrid ML-DL models have been developed that combine the most desirable qualities from both model types [11]. According to Reddy et al. [11] , CNN-SVM hybrid models can showcase high accuracy and true positive rates as well as a low false positive rate. Furthermore, the hybrid model combines SVMs interpretability and the efficiency of CNNs, making them suitable for IoT networks.

Recurrent Neural Networks (RNNs)
Recurrent neural networks are supervised DL models that consist of an input layer, hidden layer and an output layer [26]. They are commonly used models in network analysis due to their ability to model sequential data, meaning they can analyse network traffic flows [11]. RNNS evaluate network traffic by looping neuron outputs back into the neural network to be processed again [8]. This architecture allows for the identifying of complex underlying patterns and anomalies in real-time network traffic [8]. RNN-models show promising results, achieving higher accuracy rates and lower false positives in intrusion detection than traditional models when tested with the NSL-KDD dataset [26]. Despite this, research [18] comparing RNN and CNN models suggests that CNNs are slightly superior. Furthermore, RNNs are susceptible to extensive training times and vanishing gradients [26]. Long short-term models (LSTMs), a variant of RNNs, mitigate this vanishing gradient problem however, they have a higher computational complexity [11]. Despite their effectiveness, RNNs are known to be resource hungry therefore making them unsuitable for operation in low-resource environments [21]. Ullah et al.[24] proposed three lightweight RNN-based models designed to operate in low-resource environments such as IoT networks. A performance analysis was conducted comparing these models with Yin et al.'s [26] RNN model using the NSL-KDD dataset [24]. The results demonstrated that the models developed by Ullah et al. [24] outperformed Yin et al.'s [26] model, achieving accuracy rates almost 30% higher and a precision rate almost 15% higher [24]. Additionally, the lightweight models maintained a false positive rate of approximately 0.06%, whereas Yin et al.'s [26] model had a notably higher false positive rate of 13% [24].

Generative Adversarial Networks (GANs)
Generative adversarial networks are supervised DL models that comprises of generator and discriminator components [11]. The generator produces synthetic data, while the discriminator is required to classify the data [11]. For example, in terms of an IDS, the generator would create synthetic network traffic data and the discriminator would have to classify which network activity is normal and which is malicious [11]. According to Reddy et al. [11], GANs are useful in IDSs for anomaly detection - particularly in identifying zero-day attacks. Due to GANs creating their own data, GAN-based IDSs improve their detection of novel attack patterns without needing to be trained on enormous amounts of labelled data [11]. Samples of malicious data are difficult to acquire [10]. This has resulted in the use of imbalanced datasets when training models that will be implemented into IDSs [10]. GANs contribution to data augmentation, which allows it to fill the gaps that training datasets currently face [11]. Research [11] has shown that GANs have high accuracy rates when detecting intrusions in networks however, their performance can decline when applied in real-world environments that differs from their training phase's setting.

Auto-encoders (AEs)
Auto-encoders are unsupervised deep learning models that comprises of encoder and decoder components [18][25]. They are commonly used for dimensionality reduction as well as feature extraction and transformation, making them practical in intrusion detection [13][25]. An auto-encoders model allows a IDS to efficiently detect anomalies in network traffic as well as adapt to changes in network traffic behaviour [8]. This is possible as AEs are only trained to learn features of unlabelled regular network traffic data, allowing them to identify activity that does not match normal traffic behaviour when deployed [8]. The encoder performs dimensionality reduction on the input data and the decoder reconstructs the encoders output [2]. The decoder's main objective while reconstructing is to reduce reconstruction error – a measure of how well a model can replicate its input [2]. When the model tries to reconstruct anomaly data a significant error will arise and tell the IDS that the data is a potential threat [16]. Traditional AEs are computationally expensive and thus are unsuitable for implementations into an IDS used in low-resource environments such as IoTs [16]. Sharmila and Nagapadma [16] proposed a lightweight Quantized Auto-encoder (QAE-u8) that addresses traditional AE limitations and is a viable option for intrusion detection in resource constrained networks. This QAE incorporates quantisation, clustering and pruning methods to increases the model's efficiency while conserving detection accuracy. Experimental results on a Raspberry Pi device demonstrated that a QAE-u8 model can outperformed a traditional auto-encoder, consuming significantly less CPU power, memory, and processing time. Research further reveals [16] that while this optimized AE model is practical for deployment in low-resource environments, it experiences a slight trade-off in performance in comparison to a traditional AE.

4 DATASETS AND EVALUATION METRICS

4.1 Datasets

Selecting an appropriate datasets to train and evaluate an IDS is a critical step in the development of an effective system. This section will look at widely used intrusion detection databases and evaluate their suitability to train a model that will be used in a Lightweight AI based Intrusion Detection System (LAIDS).

4.1.1 NSL-KDD. A refined version of the KDD'99 dataset consisting of normal network traffic data, along with Probe, DDoS, R2L and U2R attack pattern data [10]. The dataset is primarily used for benchmarking models used in IDSs [10]. Drawbacks of this dataset are its lack of IoT-specific data and its class imbalance [10]. This dataset is also outdated and does not accurately represent real-world network traffic [10]. These issues can limit the effectiveness of models trained on the dataset [10]. Studies have shown [2] that models in IDSs meant for deployment in IoT networks perform better on datasets that specifically contains IoT network activity. Therefore, the NSL-KDD is not an ideal dataset to use to train a model that will be used in a LAIDS.

4.1.2 UNSW-NB15. A widely used dataset that contains over 2.5 million network packets and nine different types of attack patterns [23]. Despite this variety of attack data, the dataset has a high imbalance of data classes with over 87% of its data being normal network activity [23]. Moreover, the dataset lacks IoT-specific data, making it less desirable to use in training models that will be deployed on IoT networks. [10].

4.1.3 CICIDS2017. The dataset was built in 2017 and comprises of real-world normal and attack network traffic data [25]. It offers a detailed set of data, making it ideal for training models that will be used in detection intrusion systems [10]. The drawback of this dataset is that it does not encompass current unseen threats[10].

4.1.4 KDDcup99. Similar to NSL-KDD, the KDDcup99 dataset consists of Probe, DDoS, R2L and U2R attack pattern data and is used to benchmark a model's performance [10]. The dataset was released in 1999 and therefore is outdated and does not contain current attack patterns [10]. The dataset is also imbalanced and contains more instances of normal network traffic than malicious[10].

4.1.5 IoT-23. A labelled IoT network traffic focused dataset that contains both malicious and normal traffic data [10]. It was created in 2020 and contains 760 million data packets that were captured between 2018 and 2019 by the Stratosphere Laboratory at Czech Technical University[10]. It is a suitable dataset to train models that will be used in IDSs but there is a concern that the dataset does not accurately represent the current IoT threat landscape [10].

4.2 Evaluation Metrics

IDSs are evaluated using a confusion matrix. True positives (TP) represent the number of abnormal network traffic correctly identified, while true negatives (TN) represent the amount of normal traffic correctly identified by the IDS. False positives (FP) are normal traffic activities that are classified as abnormal by an IDS and false negatives (FN) are abnormal traffic reported as normal by the IDS [26].

4.2.1 Accuracy. The overall correctness of an IDS and a common evaluation of an IDS's performance [10][26]. It is the proportion of false positives and true positives over the total number of scenarios examined[10].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}\,[26] \tag{1}$$

4.2.2 Precision. The amount of true positive results out of all positive results the IDS reported [10]. It measures an IDS's ability to pinpoint actual threats [10].

$$\text{Precision} = \frac{TP}{TP + FP}\,[10]. \tag{2}$$

4.2.3 Detection Rate. The proportional of true positive results the IDS reported to the total number of actual intrusion data points in the dataset [10]. It is another way to measure an IDS's ability to pinpoint actual threats [10].

$$\text{Detection Rate} = \frac{TP}{TP + FN}\,[10]. \tag{3}$$

4.2.4 False Positive Rate/False Alarm rate. The proportion of false positive results reported by the IDS to the total amount of data points the IDS incorrectly classified [10]. It is a measure of an IDS's reliability [10].

$$\text{False Positive Rate/False Alarm rate} = \frac{FP}{FP + TN}\,[26] \tag{4}$$

5 DISCUSSION

The following section will compare and contrast traditional and AI based IDSs, looking at factors such as accuracy, efficiency, and resource consumption. Furthermore, ML and DL based IDSs well be compared, highlighting feasible lightweight and Hybrid models. Finally, the implications of these findings for the development of a lightweight AI based IDS will be discussed, addressing a way forward for the project as well as any expected implementation challenges.

5.1 A Comparison of Traditional and AI based Intrusion Detection Systems

Traditional intrusion detection systems that are signature or anomaly based have failed to adapt to the current dynamic cyber attack landscape [23][19]. Furthermore, these traditional IDSs are often computationally expensive and thus, unsuitable for low-resource environments such as IoT networks [19]. While signature-based IDSs have the potential to be adapted into lightweight IDSs that can efficiency detect known anomalies in network traffic, they are unable to identify new variants of known attacks or new attacks [19]. This is an undesirable property to have as day-zero attacks become more prominent. Anomaly-based IDSs (AIDS) are able to detect unknown threats but are prone to high false positive rates and high computational overhead costs [10][25]. Knowledge-based AIDSs' high maintenance requirements and statistical-based AIDs' lack of accuracy make them unsuitable for implementation in an IDS meant for deployment in low-resource networks [6]. This is likely one of the reasons that machine learning AIDS have become more prominent .

There has been an ongoing trend in researchers moving away from traditional IDS methods and focusing on AI based IDSs . This may be due to AI based IDSs' ability to adapt to dynamic threat landscapes as well as achieving higher accuracy and precision rates [11].

Compared to signature-based IDSs, AI based IDSs are able to detect unknown attack patterns as well as adapt to changing network behaviour [11]. Furthermore, AI based IDSs have reduced false positives in comparison to traditional IDSs [11][18]. AI-based IDSs offer superior accuracy and efficiency than traditional IDSs. Despite their high resource consumption, they are better suited for implementation in intrusion detection systems.

5.2 A Comparison of Machine Learning and Deep learning Intrusion Detection Systems

Since deep learning is a subset of machine learning, ML and DL models share some similarities. ML and DL models require sizable amounts of training data [8] and can both be computationally expensive to train and run [8][25]. In contrast, DL models have demonstrated better performance than ML models in intrusion detection [8]. T. Sowmya and E.A. Mary Anita's [18] comparison of the accuracy rates of ML-based IDSs and DL-based IDSs demonstrated percentages reaching into the 90s for ML models such as K-NNs, random forests as well as DL models such as CNNs, RNNs and AEs. However, T. Sowmya and E.A. Mary Anita [18] noted that, in general, DL-based IDSs achieved higher accuracy rates than their ML counterparts, with a CNN-based IDS developed by Pengju Liu [7] achieving the highest accuracy rate of 99.95%. This implies that a CNN could be a useful baseline model in the development of a lightweight AI based IDS for low-resource networks.

Performance comparisons of M. Uduremen et al.'s [23] DL models with ML models such as KNN, SVM, and RF reveal that their models outperformed these ML models, yielding higher accuracy and precision rates and further proves that DL models generally outperform ML models. The enhanced performance of DL models in comparison to ML models comes at a cost. DL models are often more effective but also more resource-intensive than ML models due to their complex neural network architecture [8][2]. To mitigate intense resource consumption while retaining detection accuracy and precision, lightweight DL models and Hybrid ML-DL models have been developed for implementation in IDSs in resource-constrained environments [11].

Hybrid model IDSs are frequently superior to single model IDSs as they can display better accuracy rates [11]. Hybrid models such as the PCA-DL or CNN-SVM models previously discussed can be successfully implemented into IoT networks with high detection rates and low computational overheads [3][12] [11] A drawback of these models are that they have intricate builds and training procedures, meaning they could be difficult to implement [11]. Lightweight models such as TinyNIDS, QAE-u8 are also able to be deployed in low-resource environments. TinyNIDs can achieve higher accuracy rates with lower latency times than CNN and RNN models [21]. QAE-u8 consumes significantly less resources on a Raspberry Pi device than a traditional auto-encoder however, it exhibits slightly less accuracy in intrusion detection [16]. Lightweight RNNs such as Ullah et al.'s [24], are also a feasible model option for IDSs in low-resource environments.

5.3 Implications for Lightweight AI based Intrusion Detection System Development Project

Developing an AI based Intrusion Detection System (IDS) for low-resource networks presents several challenges and trade-offs. A lightweight deep learning model or a hybrid model that combines machine learning and deep learning (Ml-DL) models appear to be the most viable options for lightweight AI based IDS (LAIDS). Research [16] indicates that lightweight models, such as QAE-u8, can effectively reduce resource consumption, however detection performance may be compromised in order to do so. Hybrid ML-DL models can operate effectively with limited computational resources without sacrificing performance [11]. This being said, they might prove challenging to implement due to their complex structure and training process [11].

CNN-based models have consistently demonstrated powerful performance and have strong potential to be the baseline model for the LAIDS project. Furthermore, optimizing CNN-based models for constrained networks is feasible, as observed with TinyNIDS [21].

The LAIDS project may face significant challenges in finding suitable network traffic databases to train and test models on. Reliable and accurate databases for training models used in IoT intrusion

detection are needed to ensure effective training and testing. Many widely used datasets such as the NSL-KDD and UNSW-NB15 lack network traffic behaviour typically found in low-resource environments [10]. Recent efforts have led to the development of more IoT-focused datasets, such as IoT-23 [10]. Additionally, these datasets hold heavy class imbalances. For example, only 13% of the UNSW-NB15 dataset is threat activity [10]. This imbalance in data can negatively impact model performance [10]. Furthermore, many publicly available databases, such as KDDcup99 and NSL-KDD, are outdated and do not contain modern attack patterns [10].

Selecting the best model and datasets for the LAIDS project can greatly impact the efficiency of its intrusion detection capabilities in low-resource network environments.

6 CONCLUSIONS

The evolution of technology and the increased adoption of IoT devices has enabled increases in cyber attacks in networks. Due to the sophistication of these attacks, traditional IDS methods have become obsolete. Furthermore, they are often too computationally expensive to implement into IDSs that operate in resource-constraint environments. This has created the need for a more efficient and reliable IDS and led to the development of AI based IDSs that make use of machine learning and deep learning models. These AI models are able to achieve higher detection accuracy rates than their traditional counterparts.

In particular, deep learning models such as CNNs and Auto-encoders show superior performance due to their neural network architecture that allows them to pick up even more subtle threats. DL models are one of the more computationally expensive machine learning models, requiring large amounts of network traffic data to be trained as well as significant resource usage for operation. This makes them unsuitable for deployment in IDSs that work in low-resource networks. To enable use of DL models in resource constrained environments, efforts have been made to develop lightweight DL models like QAE-u8 and TinyNIDS as well as Hybrid ML-DL models such a PCA-DL or CNN-SVM. These models maintain the precision and accuracy of deep learning models while minimizing resource use, making them feasible baseline models from which the LAIDS project can build on.

REFERENCES

[1] 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

[2] 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

[3] Mohsen Eslamnezhad and Ali Yazdian Varjani. 2014. Intrusion detection based on MinMax K-means clustering. *7'th International Symposium on Telecommunications (IST'2014)* (09 2014). https://doi.org/10.1109/istel.2014.7000814

[4] Pajar Abdul Malik Hambali, Syamsuddin, Mufid Ridlo Effendi, and Eki Ahmad Zaki Hamidi. 2020. Prototype Design of Monitoring System Base Tranceiver Station (BTS) Base on Internet of Things. *2020 6th International Conference on Wireless and Telematics (ICWT)* (09 2020), 1–6. https://doi.org/10.1109/icwt50448.2020.9243661

[5] 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

[6] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* 2 (07 2019), 1–22. https://doi.org/10.1186/s42400-019-0038-7

[7] Pengju Liu. 2019. An Intrusion Detection System Based on Convolutional Neural Network. *Proceedings of the 2019 11th International Conference on Computer and Automation Engineering* (02 2019), 62–67. https://doi.org/10.1145/3313991.3314009

[8] Salman Muneer, Umer Farooq, Atifa Athar, Muhammad Ahsan Raza, Taher M. Ghazal, and Shadman Sakib. 2024. A Critical Review of Artificial Intelligence Based Approaches in Intrusion Detection: A Comprehensive Analysis. *Journal of Engineering* 2024 (01 2024). https://doi.org/10.1155/2024/3909173

[9] Vasudeva Pai, Devidas Bhat, and Adesh N.D. 2021. Comparative Analysis of Machine Learning Algorithms for Intrusion Detection. *IOP Conference Series: Materials Science and Engineering* 1013 (01 2021), 012038. https://doi.org/10.1088/1757-899x/1013/1/012038

[10] Md Mahbubur Rahman, Shaharia Al Shakil, and Mizanur Rahman Mustakim. 2025. A Survey on Intrusion Detection System in IoT Networks. *Cyber Security and Applications* 3 (12 2025), s. https://doi.org/10.1016/j.csa.2024.100082

[11] 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

[12] Muhammad Sajid, Kaleem Razzaq Malik, Ahmad Almogren, Tauqeer Safdar Malik, Ali Haider Khan, Jawad Tanveer, and Ateeq Ur Rehman. 2024. Enhancing intrusion detection: a hybrid machine and deep learning approach. *Journal of Cloud Computing Advances Systems and Applications* 13 (07 2024). https://doi.org/10.1186/s13677-024-00685-x

[13] Aya Salem, Safaa Azzam, O Emam, and Amr Abohany. 2024. Journal of Big Data Advancing cybersecurity: a comprehensive review of AI-driven detection techniques. *Journal of Big Data* 11 (2024), 105. https://doi.org/10.1186/s40537-024-00957-y

[14] 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

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

[16] 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

[17] Nazim Uddin Sheikh, Hasina Rahman, Shashwat Vikram, and Hamed AlQahtani. 2018. A Lightweight Signature-Based IDS for IoT Environment. https://doi.org/10.48550/arXiv.1811.04582

[18] 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

[19] Charles Stolz, Fuhao Li, and Jielun Zhang. 2024. Implementing Lightweight Intrusion Detection System on Resource Constrained Devices. *2024 Cyber Awareness and Research Symposium (CARS)* (10 2024), 1–6. https://doi.org/10.1109/cars61786.2024.10778716

[20] 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

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

[22] Shyava Tripathi and Rishi Kumar. 2018. Raspberry Pi as an Intrusion Detection System, a Honeypot and a Packet Analyzer. , 80–85 pages. https://doi.org/10.1109/CTEMS.2018.8769135

[23] Miracle Udurume, Vladimir Shakhov, and Insoo Koo. 2024. Comparative Evaluation of Network-Based Intrusion Detection: Deep Learning vs Traditional Machine Learning Approach. *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)* (07 2024), 520–525. https://doi.org/10.1109/icufn61752.2024.10625037

[24] Imtiaz Ullah and Qusay H. Mahmoud. 2022. Design and Development of RNN Anomaly Detection Model for IoT Networks. *IEEE Access* 10 (04 2022), 62722–62750. https://doi.org/10.1109/access.2022.3176317

[25] 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

[26] 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

[27] 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

[28] 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