A Literature Review of Lightweight AI-Based Intrusion Detection Systems (LAIDS) for Resource-Constrained Networks

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

ABSTRACT

The widespread adoption of Internet of Things (IoT) devices has introduced major cyber security risks. This adoption has amplified the need for effective Intrusion Detection Systems (IDS) that can operate efficiently in resource-constrained environments. This literature review proposes that Lightweight AI-Based Intrusion Detection Systems (LAIDS) for low resource networks are a viable solution. Making use of a hybrid models, LAIDS aims to optimise the trade-off of security performance for computational efficiency, offering a viable solution for improved network security in environments with constrained resources. The IDS does this by exploring various Artificial Intelligence (AI)-enhanced IDS approaches, with a focus on hybrid models that combine lightweight machine learning techniques with deep learning algorithms to improve detection accuracy and performance. Techniques such as Support Vector Machines (SVM), Logistic Regression (LR), and K-means clustering are examined for their potential to preprocess data and extract important features, minimizing the computational overhead of the IDS. Deep learning models like Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs) are then employed to identify complex patterns in network traffic, improving the detection of disguised malicious activity. The study highlights the challenges of balancing security performance with computational efficiency, particularly in low-resource environments. Finally the study emphasises the critical need for security solutions for resource-constrained networks.

KEYWORDS

Artificial Intelligence, Machine Learning, Deep Learning, Resource-Constrained Networks, Cybersecurity, Internet of Things, Intrusion Detection System, Low-resource Environments

1 INTRODUCTION

The introduction of the connectivity of various devices through the Internet of Things (IoT) has created a large demand for miniature, low-resource networks [31]. Cisco predicts that the number of devices connected with the IoT will reach 500 billion devices by 2030 [44]. This has expanded the proliferation of low-resource networks significantly in the past decade. This can be seen with the introduction of various inexpensive micro controllers such as the: Arduino, Raspberry Pi and ESP32 as well as various others [36]. These low-resource networks range from providing internet access to rural areas, to automating production lines, to ensuring that you can play music from your phone to your wireless earphones.

Despite being computationally expensive, the implementation of IDS is becoming commonplace as corporate entities and governing bodies enforce more stringent compliance regulations [11]. As such an IDS is becoming a vital component of any network. This increase exacerbates the challenges of implementing robust IDS in resource constrained networks. For context; most micro-controllers have limited memory. A typical Raspberry Pi 4 micro-controller has between 1-8 Gb of memory — quite a large amount of memory compared to other commercial micro-controllers like the Arduino offering 1MB of memory or ESP32 offering 520 KB of memory — this provides unique challenges to the implementation of AI-IDS as the AI model has to be able to reliably diagnose incoming traffic, store potentially harmful packets as well as continually train itself all while ensuring its network performance impact is minimal, and that the micro-controller has enough memory to continue its functions. The purpose of this literature review is to enlighten the reader on the various approaches and algorithms used to create an IDS while comparing them using various benchmarks. This literature review aims to highlight and compare the performance of existing approaches while identifying gaps that warrant further research in the endeavor of Lightweight AI-Based Intrusion Detection Systems for Resource-Constrained Networks.

2 TRADITIONAL INTRUSION DETECTION SYSTEMS VS AI-BASED INTRUSION DETECTION SYSTEMS

With advances in the fields of ML and Deep Learning (DL), IDSs have transformed from using traditional approaches to modern self-improving AI-powered systems. While TIDS relies on a set of signature, anomaly and rule-based techniques, AI-IDS are able to incorporate a range of adaptive techniques to fine tune their ability to detect threats, as well as process larger volumes in real time than TIDS [30].

2.1 Traditional IDS Implementations

TIDS have various implementations, however the three major implementations are: Anomaly-based, Signature-based and Rule-based IDS. Anomaly-based IDS detects anomalies by comparing current network traffic to a predefined 'normal' network traffic [30]. Signature-based IDS is an IDS where network activity is compared to a database of known attack patterns to determine if it is malicious or not [30]. Rule-based IDS is an IDS in which a set of rules —typically defined by the network administrator— determine whether network activity is malicious or not [41].

Despite most TIDS implementing a combination of the three, they lack the flexibility to handle innovative attack techniques [2]. Consequently a large amount of resources are required to ensure that

the TIDS is up to date for the current cyber security landscape. This is due to the inflexible nature of the signature and rule-based IDSs. Additionally, while any TIDS implementing some form of anomaly-based IDS is able to adapt and thus require less maintenance [7]. These IDS have a much larger false positive rate [7], which in turn requires constant supervision from a network admin or a blue team.

2.2 ML-Based IDS Implementations

The implementation of AI introduced adaptive learning capabilities to the IDS. This has had the effect of not only enhancing IDS threat detection abilities, but also future-proofing, and in turn, extending the lifespan of the IDS as a whole [30]. The implementation of AI in IDS can be split into two categories: ML-based and DL-based. These categories can be further divided into supervised, unsupervised and semi-supervised learning.

ML is a method in which a model is trained using algorithms that learn from historical data and, as such, can adapt itself without manual intervention [39]. In the field of IDS, these models are trained on large datasets of network traffic in order to enhance their threat detection capabilities [41]. These models are able to detect abnormal patterns which are unable to be detected by regular humans. As such novel attack vectors and zero-day attacks are more likely to be determined and combated swiftly reducing potential damage [41]. These models can be divided into the categories of supervised and unsupervised learning. The difference between the two is found in the training data, where in supervised learning, the model is fed datasets where each input has a mapped output [35]. This training process is supervised by the user and has the goal of teaching the model to map specific inputs to specific outputs [41]. Unsupervised learning differs from supervised learning in both purpose and training data. The aim of unsupervised learning is to train the model to discover inherent patterns and categories within the data thus improving itself on its own [18]. To do this, the training data does not have a mapping from input to output unlike supervised learning. Rather, the model itself must determine the patterns and adapt itself to improve its performance [18].

In the concept of IDS, ML supervised learning algorithms such as k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), Random Forest, Linear and Logistic Regression as well as unsupervised learning algorithms such as K-means clustering, Apriori and Principal Component allow the system to recognize patterns in network traffic and enact appropriate steps to limit if not entirely mitigate the effects of a cyber attack [41] [30].

A. k-Nearest Neighbor (k-NN)

k-NNs are a supervised learning algorithms which are quite popular in IDS. This is because of its achieving high performance, while maintaining its simplicity through comparing data points to their k-nearest neighbors [18]. That being said, the computational complexity of calculating the distance between each neighbors rises rapidly. This results in a computational complexity of $O(kn)$[1] for the algorithm [4]. This is not feasible in real time [17] and as such, k-NNs are unsuited for larger datasets. That being said, they are highly effective in processing datasets which can be entirely held in RAM — those being small to medium-sized datasets [30]. With these limits in mind, k-NNs may be used in resource-constrained environments when combined with other techniques such as data reduction or sampling.

Sowmya and Anita [35] provided a survey of the literature on ML implementations of IDS and found that the K-NN implementation achieved an accuracy of 99.89%, the highest of all the ML-based IDSs that they analyzed. Similarly, Peng et al. [26] proposed an IDS model which specialized in processing large volumes of data in fog computing. In doing so, they found that the K-NN model achieved a higher accuracy in classifying probe attacks compared to other, existing models. Moving away from prioritizing accuracy, Alippi and Roveri [4] identified the shortcomings of K-NN's when applied to resource constrained environments. To overcome these flaws, they implemented optimization techniques on the algorithm such as instance selection, KD-trees and Principal Component Analysis. In doing so, they created a lightweight K-NN based IDS with performance comparable to unoptimised models.

B. Support Vector Machine (SVM)

SVMs are supervised learning algorithms that are commonly applied in IDS as they demonstrate exceptional performance in to binary and multi-class classification [30]. They are able to effectively employ binary classification (determining whether network traffic activity is malicious or not) as well as employing multi-class classification in the form of determining what type of malicious activity is present e.g: DDos, Portscan, Privilege escalation attempts etc... Additionally, SVMs have been widely adopted as classifiers due to their ability to remain exceptionally accurate while preventing overfitting [35]. However SVMs are binary classifiers and as such, most successful IDS implementations incorporate a multi layered architecture to distinguish between classifications [41]. As such, the implementation of a SVM can be a memory intensive operation. That being said, SVMs are able to employ technique such as dimensionality reduction which may be used to reduce the computational power required to employ an SVM [18], thus making the utilization of SVMs in low-resource environments more practical.

Kabir et al. [13] proposed a novel SVM-based IDS implementation that classifies in two stages. They implemented a least-squares SVM. The SVM first divides the input data into subgroups and extracts features from these subgroups using a optimum allocation scheme. The SVM was then employed to detect if any intrusions had taken place. Their model was able to achieve an accuracy of 99.67% when tasked with multi-class classification. Jha and Ragha [12] also employed a SVM while evaluating the feasibility of SVMs in IDS. They found that SVMs. In doing so, they proposed a novel hybrid approach which combined filter and wrapper modules to select relevant features from the inputted data. They then employed a SVM to classify the processed dataset which in turn achieved an exceptional degree of accuracy compared to other implementations. Mohammadi et al. [21] conducted a survey of SVM implementation in IDS. They highlighted that SVM performance is acceptable in resource-constrained environments when the data was pre-processed. However they

[1] k represents the number of neighbors taken into account for the classification decision

highlighted that further research should be conducted into the optimization of SVMs for low-resource environments such as IoT.

C. Random Forest

Random Forest is another supervised ML algorithm that serves as an ensemble classifier. Random Forests have been adopted in a wide range of IDS as they excel at computing large datasets while remaining computationally cheap [30]. Random Forests have also been proven to detect complex patterns while maintaining a high degree of accuracy on large datasets [1]. That being said, while running a Random Forest model remains a computationally cheap option, training such a model requires a large amount of computational resources [1]. This in turn shall result in network performance suffering significant degradation during periods of training.

Chen and Yuan [8] created an IDS using a Random Forest implementation. Their system utilized a multi-layered architecture in which they had a module to detect signals and extract the important features of said signals. They then used a model to classify the traffic as malicious or normal. Making use of reinforcement learning and static feature fusion, they were able to obtain an accuracy of 96.93%. Similarly Abdelaziz et al. [1] investigated the effectiveness of implementing a Random Forest model in an IDS with regards to classifying network traffic. They found that by applying a custom prediction function, their Random Forest classifier achieved an f1-score of 93.31%. They further noted the system's exceptional ability to detect day-zero attacks. That being said, neither study attempted to employ their solution in a low-resource environment. Further research to investigate the optimization of a Random Forest model should be done.

D. Logistic Regression (LR)

LR is a supervised ML algorithm which is used for binary or multi-class classification. It is a model which is simple and computationally cheap and thus useful for large datasets [35]. Additionally they demonstrate exceptional accuracy in classifying network traffic, so much so that they are able to compete, and in some cases, outperform existing complex supervised ML algorithms [37]. That being said, a major limit of LR models is that their primary purpose binary or multi-class classification and as such are unable to detect more advanced patterns, limiting their used for novel attack vectors. While not robust enough a model to serve as an IDS alone, it shows potential through being incorporated into a multilayered architecture as a first-line defence of classifying network traffic as either normal or malicious before employing other more advanced and computationally expensive models to analyse flagged and potentially malicious data.

Ponmalar and Dhanakoti [28] proposed a novel approach to IDS in the form of the genetic algorithm and logistic regression. The genetic algorithm was used to preprocess the NSL-KDD dataset, reducing the number of features. The LR algorithm would then classify the pre-processed data. They found that their model achieved a F1-score of 95.98% and accuracy of 96.29% — results better than their baseline implementation. Subba et al. [37] implemented a LR-based IDS implementation. They compared this implementation to other ML-IDS implementations such as a Native-Bayes-based and a SVM-based IDS implementation. They found that their model performed as well, if not better than the other ML-based implementations achieving an accuracy of 95.31% and a detection rate of 98.74%. They also noted that their LR model was more computationally efficient than the other ML-implementations. This in turn shows that there is potential for LR-based IDS implementations in resource-constrained enviroments.

E. K-means Clustering

K-means clustering is an unsupervised ML technique in which partitions data into K number of different clusters. It is one of the most employed clustering analysis techniques [15]. One of the major benefits of K-means clustering is that is is an unsupervised learning algorithm. In the context of IDS implementation, the model can identify malicious traffic by flagging any traffic that falls far from established cluster centres as potentially malicious [41]. This in turn allows for the model to train and enhance itself without the need for network admin supervision [41]. K-means clustering has been widely adopted among IDS namely for its accuracy while remaining a relatively lightweight model, even when processing large datasets. However, while it is an unsupervised learning algorithm, K-means clustering still requires a predefined K value [9]. Additionally the model has to undergo a period of oversensitivity while it trains itself [9]. This may result in the model having to be pretrained on the specific network's previous data before being deployed into the IDS of the network.

Eslamnezhad and Varjani [9] attempted to optimise K-means clustering for IDS. In doing so, they proposed a model which performed quite well in detecting certain attacks. The model used the NSL-KDD dataset to evaluate its performance. It was noted however that the model demonstrated particularly poor performance when detecting Remote-to-Local and User-to-Root attacks. That being said Ravale et al. [29] proposed a hybrid IDS combining K-means Clustering and SVMs. They used the SVMs to simplify the KDDCup99 dataset using feature extraction and a K-means Clustering model to classify the network traffic. In doing so, they were able to overcome the shortfalls observed by Eslamnezhad and Varjani. Furthermore, the use of their hybrid model resulted in a reduced computational complexity compared to a standard K-means Clustering-based IDS showing the poential for a lightweight implementation of a K-means Clustering IDS.

2.3 DL-Based IDS Implementations

Deep Learning is a subset of algorithms in the realm of ML. The difference between DL and ML is that in ML, the training data fed into the model must be sanitized and formatted before being used. This is not the case with DL models they are able to learn from unstructured data [6]. As such, deep learning models are able to analyse, classify and train themselves on the raw network traffic [41] rather than formatting the data which can be computationally expensive. The adoption of DL in IDS has greatly enhanced IDS detection capabilities, allowing it to detect more intricate patterns in the network data and revealing more unusual attacks [30]. DL adoption into IDS has greatly enhanced the scalability and adaptability of IDS. The following are DL models commonly implemented in

IDS:

A. Convolutional Neural Networks (CNN)
CNNs are supervised DL models which are widely applied in IDS for their ability to recognize complex patterns in high dimensional data [30]. By applying convolutional layers, CNNs are able to establish local dependencies used to distinguish patterns [40]. These models are able to process high volumes of data as well as use reasoning from previous data points to establish patterns. For these reasons, CNNs are suitable for implementation IDS as they have proven to accurately distinguish normal from malicious network traffic [30]. As they are DL models, they are also able to analyse raw data, which in the case of IDS would be raw packet data. This provides an opportunity for the model to detect suspicious patterns that other traditional or ML IDS may miss. The issue with CNN implementation in a resource-constrained environment is that the computations required by CNNs are quite expensive which may result in the degradation of the performance of the network. That being said, the implementation of a hybrid architecture incorporating CNNs as well as other, less resource intensive DL models may mitigate this shortcoming [30][17].

Reddy et al. [30] noted that CNNs particularly excel at processing high dimensional data and recognising complex patterns. They also noted that CNNs are suited to process large volumes of data. Capitalizing on these features of CNNs, Udurume et al. [40] compared a range of AI models in their effectiveness of being implemented as an IDS. Their study focused on binary and multi-class classification. They found that CNN model achieved an accuracy of 92.692% and 94.845% for binary and multi-class classification respectively. Pham et al. [27] implemented a lightweight IDS leveraging a CNN model. They were able to implement a lightweight CNN model through preparing the network data into an image. By doing this, they were able to minimize the computational strain on the network while maintaining a detection time of 3.23 microseconds. Thus showing that there is potential for CNN IDS implementation in resource-constrained environments.

B. Generative Adversarial Network (GAN)
GANs are more niche of an IDS implementation than the other models listed above. They are classified as a semi-supervised learning model, however they are primarily an unsupervised learning model which may be modified into a supervised learning model. The model is made up of a generator and a discriminator component [32]. In the context of IDS the generator generates synthetic network patterns and the discriminator classifies the inputted network traffic as real or generated [30]. The discriminator can be employed to distinguish normal network behaviour from abnormal network behaviour. This approach benefits from a lack of reliance on known attack signatures [30]. This makes GANs particularly effective for detecting zero-day attacks — a feature paramount to quality of the IDS. GANs can also be optimised for low-resource environments with a variety of techniques such as only deploying the discriminator component to the network or using efficient activation functions such as LeakyReLU [19].

Shahriar et al. [32] compared the performance of a GAN-based IDS implementation to a stand alone IDS. Both models were trained on the KDDCup99 dataset. They found that the Gan-based IDS yielded a higher F1 score than the stand alone IDS. Similarly Park et al. [25] explored the use of GANs in IDS, finding that GANs can learn new attack signatures through synthetic data generation. This in turn improves the robustness of the model against novel attack vectors. Ali et al. [3] further analysed the incorporation of GANs in IDS, particularly in a resource-constrained environment for IoT application. Their findings showed that the merging of adversarial training and knowledge distillation resulted in a highly accurate IDS which was more resource efficient than typical GAN-based IDS implementations. However, they did note that more research must be conducted into the lightweight optimization of GAN-based IDS.

C. Autoencoder
Autoencoders are a group of unsupervised DL models. They are made up from a three-phase architecture in which the autoencoder takes an input, encodes it and then decodes it into an output [10]. In the context of IDS, the autoencoder tries to reconstruct the network traffic after encoding it. If an certain error threshold is crossed during reconstruction, then the traffic is flagged as potentially malicious [10] . This has the advantage of not relying on knowing specific attack signatures and as such is highly adapted to innovative attack vectors. That being said, there are many attack vectors that disguise themselves as normal network traffic such as Man-in-the-middle attacks and DNS Tunnelling which may evade detection. However, this model trains itself to enhance these processes without user supervision. One of the processes that it enhances is extracting meaningful data from packets which may aid its performance against disguised attacks [16]. In addition to this, the model may enhance its ability to eliminate noise from the dataset [10] — a common feature of network traffic. Another benefit of autoencoders is that there exist lightweight models for IDS which are optimised for resource-constrained environments [20].

Faber et al. [10] identified the need of a lightweight IDS in both cloud and mobile device security. They proposed an autoencoder-based IDS in which they were able to run the neural network and consequently, the intensive computations required on a server. This allowed for resource constrained devices to offload the work while ensuring security on the network. In addition to this, Reddy et al. [30] also noted that the cloud networks can have be beneficial for network security as it allows for the network to tolerate larger loads as well as enhance its resilience towards DDos attacks. Kunang et al. [16] also employed an autoencoder model for IDS while employing measures such as dimensional reduction to enhance the performance of the model while optomizing it for lower-resource enviroments. They found that using a ReLu activation with a cross-entropy loss function resulted in an accuracy of 99.947% on the KDDCup99 dataset. While the model was accurate, it still required a resource intensive IDS. Doitshman et al. [20] proposed their own autoencoder-based IDS called Kitsune. It makes use of a distributed architecture in the form of an ensemble of smaller autoencoders which they dubbed the KitNet. Their IDS was capable of achieving a false positive rate of 0.001. They also found that increasing the

number of smaller autoencoders yielded better results.

D. Deep Belief Network (DBN)
DBNs are unsupervised DL models which are made up of multiple stacked Restricted Boltzmann Machines (RBM)s, each RBM serving as a layer of the DBN. Using back-propagation, the models are able to fine tune themselves [34]. When being trained, the lower level RBMs capture basic features while higher level RBMs will capture more complex patterns [34]. In the context of an IDS implementation, the lower levels of the DBN will capture basic features such as packet size an destination port numbers. Higher levels will learn to recognize complex patterns which may resemble attack signatures. DBNs outperform most other machine learning models in regards to both accuracy as well as detection rates when implemented as an IDS [5]. While DBNs demonstrate a high degree of accuracy and are able to process large volumes of data [5], they are quite memory intensive DL models. Additionally the training process of DBNs is computationally expensive with multiple rounds of training for each layer of the DBN [34]. When implemented into a low-resource environment, the training of the DBN shall have a negative impact on network performance. That being said, the utilization of techniques such as feature selection, dimensionality reduction and network pruning which reduce the computational expense of the model [24] may result in DBNs being a viable model to be implemented.

While reviewing the academic literature of DBNs, Sohn et al. [34] noted the trend of most DBNs models using 3 hidden layers while applying a back-propagation algorithm to train the system. They recommended that further research should be done in regards to these fine-tuning algorithms. Nivaashini et al. [23] implemented a federated DBN model which achieved an accuracy of up to 99% This model outperformed other DBN implementations. They also noted that a larger number of devices used in learning resulted in a higher performance. In addition to this, Sohn et al. [34] also noted that the use of DBNs as IDS is still in its infancy and there were a large number of research opportunities.

3 MEASURING THE PERFORMANCES OF INTRUSION DETECTION SYSTEMS

3.1 Datasets To Be Used

Datasets are used to gauge a model's effectiveness in determining network-based intrusion. These datasets provide a large amount of normal network traffic as well as malicious network traffic. This allows us to quantify various measurements on how certain models perform. That being said, due to various security and privacy issues most datasets used to train IDS models are not made public [43]. The datasets which are publicly available have undergone anonymization which may have impacted the integrity of the data [43]. The following datasets are publicly available and commonly used to train IDS models.

A. UNSW-NB15
UNSW-NB15 is a dataset created by The cyber security research team of Australian Centre for Cyber Security. It is widely referenced in literature. The dataset comprises of over 2.5 million network packets [40] and boasts a collection of the following 9 attack types: Dos, Exploits, Generic, Analysis, Backdoors, Reconnaissance, Shellcode, Fuzzers and Worms [41]. The dataset is skewed towards non-anomalous packets which make up 87% of the packets [40].

B. CICIDS2017
CICIDS2017 is a dataset which was created by The Canadian Institute for Cybersecurity in 2017. It was created using real-time traffic data which was captured over the course of 5 days [41]. The dataset contains both normal and malicious network activity. Containing over 100000 pieces of network data in which the following attack vectors are found: Brute Force FTP, Brute Force SSH, Dos, Heartbleed, Web Attack, Infiltration, Botnet and DDoS [43].

C. KDDCup99
KDDCup99 is one of the most widely used datasets when training IDS models [42]. It was created by the Lincoln Laboratory of the Massachusetts Institute of Technology. It is the result of the processing the data of the 1998 DARPA Intrusion Detection Challenge dataset [43]. It is a robust dataset containing 4.9 million samples, each sample having 41 features [41]. Additionally 39 types of network attacks are found within the dataset. A flaw in this dataset is that it is highly skewed towards DoS [43]. Nevertheless the dataset is still widely used.

D. NSL-KDD
The NSL-KDD dataset was developed by Tavallaee et al [38] to address the shortcomings of the KDDCup99 dataset. It retains the essence of the KDDCup99 dataset while filtering it to remove redundancies[16]. It is modified to be less biased and as such is less likely to train biased models off of this dataset than the KDDCup99 dataset [16]. This reduction in size means that the entire dataset can be used when training the model rather than using statistical sampling [14]. In short, it is a modified version of the KDDCup99 dataset which aims to address its predecessor's flaws.

E. Kyoto 2006
This dataset was created by Kyoto University. It was developed through deploying various security measures such as honeypots and web crawlers to collect network traffic [41]. The dataset consists of twenty four statistical features to help analyze the data. Fourteen of which are from KDDCup99 [42]. Despite the name, this dataset has been continually updated and the most recent version holds network traffic from 2006 to 2015 [42].

3.2 Evaluation Metrics

Various evaluation metrics are used to determine the effectiveness of the IDS implementation of the models. In order to understand the purpose of the metric as well as what it is evaluating, an explanation of each metric is mentioned below:

Accuracy
Accuracy measures the efficiency of an IDS. It is the ratio of correctly

identified classes to all samples [41].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision
Precision is the ratio of all correctly classified attack samples to all attack samples [41]. This measures the IDS's ability to detect real threats.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

Detection Rate
Also known as recall, this measure is the ratio of all correctly classified attack samples to the total number of attack samples [35]. This measures how well an IDS captures attacks compared to the number of attacks that actually happen.

$$\text{Detection Rate} = \frac{TP}{TP + FN} \tag{3}$$

False Positive Rate/False Alarm rate
This measure is the ratio of samples that are wrongly classified as attacks compared to the total number of samples that are normal [35].

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

F1-Score
This measure is also known as the F-measure. It is the mean of the IDS's Precision and Detection Rate. It is used to evaluate the system by showing the difference between the two metrics to ensure that the IDS is balanced [41].

$$\text{F1-Score/F Measure} = 2 \cdot \frac{\frac{TP}{TP+FP} \cdot \frac{TP}{TP+FN}}{\left(\frac{TP}{TP+FP}\right) + \left(\frac{TP}{TP+FN}\right)} \tag{5}$$

4 DISCUSSION

A large amount of research has been done into the enhancement of IDS through AI. That being said, a few key trends have emerged from the academic literature. What follows is a discussion of these trends. Then the performance/resource trade off shall be discussed before finally explaining how the LAIDS project shall contribute to the academic discussion on lightweight IDS implementations.

4.1 Observations

As discussed above, there are two major categories when referring to IDS implementations: TIDS and AI-IDS. With the ever-evolving cybercrime landscape, it is essential for security systems to be able to adapt and detect day-zero attacks [7][2]. Sharif et al. [33] highlighted the potential increasing financial loss due to cybercrime further emphasizing the economic notion for robust IDS implementations. There is a general consensus among the literature that AI-IDS perform better than TIDS and as such, the implementation of AI into IDS is paramount[7][30][41].

The AI models to be implemented into IDS fall into two categories: ML and DL. Across all the ML and DL models that have been examined this paper, a common trend has shown itself. DL implementations tend to out-perform ML implementations. This can be seen through Doitshman et al. [20] and their IDS proposal Kitsune which performed better than most ML algorithms, despite being a lightweight IDS approach. Another example was Shahriar et al. [32] which found that their GAN-based IDS yielded a higher F1 score than other ML-based IDSs. Most ML-IDS implementations require preprocessing of data in order to achieve the same performance as DL models. This is due to ML's innate attribute of requiring the data prepared into a specific format before the model can process said data. Unlike DL which is able to read in and automatically extract features from the raw network data. The literature discusses a general trend of DL models outperforming their ML counterparts [41][20][30]. This is seen through DL models continuously scoring a higher accuracy [30][41], precision, and F1-score [35]. DL models are also more capable of detecting zero-day attacks [20][41] and as such are more robust. That being said, DL models are more computationally expensive than their ML counterparts. In a low-resource environment, DL models, while providing a robust defence in the form of the IDS, may degrade network performance due to their computational complexity [24].

An interesting trend found among the literature is that hybrid IDS models outperformed most of the single-model-based IDS models [12][29][17]. These models were able to leverage the strengths of one model to make up for the weaknesses of another and vice-versa. By doing this, these models were able to enhance the detection capabilities of the overall system. Examples of which can be found in the works of Jha and Ragha [12] which implemented a hybrid SVM-based IDS which yielded exceptional results. Or Ravale et al. [29] which proposed a hybrid K-means Clustering IDS which was both lightweight and outperformed standard K-means Clustering IDS implementations. Another benefit of these hybrid models is that they were able to demonstrate this enhanced performance in resource constrained environments. An implementation of a hybrid DL model possibly incorporating a ML model or data preprocessing will be able to make use of a DL model's performance, while reducing its computational complexity. Other measures to reduce AI-IDS model complexity such as feature reduction, principle component analysis and dimensional reduction are also becoming more common place in the lightweight optimization of IDSs [18] [1] [41]. SVMs have been implemented with other AI models, primarily for the purpose of reducing the complexity of the data being fed into these other models thus reducing the computational complexity of the IDS overall [12] [41]. Autoencoders and GANs are DL algorithms which have shown promise in being adapted into lightweight hybrid models while maintaining a high degree of performance [30] [20]. Further study into the optimisation of these models should be conducted.

4.2 The Performance/Resource Trade Off

When analysing the performance of most computing algorithms, the relationship between performance and resource requirements comes into focus. This has been touched upon earlier in the discussion when mentioning that DL models, which are more complex than their ML counterparts, are more resource intensive. In the context of the LAIDS project, this relationship is all the more imperative. As such the balance between the performance of the IDS and the performance of the network node or even the network as a whole must be taken into account when developing an AI-IDS. While some papers have mentioned the computational expense required by these AI models such as Alippi and Roveri [4] defining the algorithmic complexity of K-NN. Most papers do not mention the memory or computational expense of these AI models unless they are proposing their own novel approaches to The Performance/Resource Trade Off problem for IDS. Two examples of this are Doitshman et al. [20] with their proposal of Kitsune or Nguyen et al. [22] and their implementation of Realguard. While this relationship can be a hindrance, it does present the unique opportunity of being able to employ less computationally expensive ML models which require preprocessing of their data rather than employing a higher performing DL model. Hybrid approaches mitigate these trade-offs by integrating multiple models in order to enhance the performance of the IDS while maintaining computational feasibility in low-resource environments [30] [12]. This can be seen through Ravale et al [29] and their use of a hybrid SVM and K-means Clustering IDS implementation which was able to achive much better results than a typical K-means Clustering-based IDS. or Additional methods do exist to try and ensure that DL models can be implemented as lightweight IDS. This has been seen with Doitshman et al. [20] with their proposal of Kitsune, a lightweight autoencoder-based IDS. Similarly, it has been seen through Mastromichalakis [19] and their implementation of a GAN-based IDS where it was optimised for low resource environments using a LeakyReLU activation function.

4.3 The LAIDS Project

This paper has been leading up to the discussion of the LAIDS project. The increasing adoption of IoT devices has led the proliferation of low-resource microcontrollers [44]. This coupled with the evolving cyber threat landscape has resulted in a heightened demand for low-resource IDSs. As discussed in this paper, TIDS are no longer feasible with the current requirements of an IDS. They are unable to process the large volume of network traffic and they are not robust enough to detect zero-day attacks. As such the demand for AI-IDS is becoming evermore present. With this in mind, most microcontrollers do posses the resources to implement the computationally expensive models which are required for these AI-IDSs without impacting the performance of both the node, or the network as a whole [20]. With this we arrive at the Performance/Resource Trade off problem where users have to make the decision of choosing security over performance. The aim of the LAIDS project is to reduce the impact of this decision allowing users to implement an effective IDS into their low-resource network while experiencing a minimal degradation in network performance.

Throughout this review, various AI models have been presented. Each model has its benefits and flaws. As such a viable LAIDS would be a hybrid approach, making use of multiple models which would be able to cover the shortfalls of each other. Determining which models to implement and balancing their advantages and shortfalls such as adversarial attacks and overfitting is an expected challenge of the LAIDS project. However, multi-staged approaches have been shown particular promise when optimizing AI models for low resource environments [8] [37]. Therefore a possible approach would involve a multi-staged approach in which the data is first preprocessed by a lightweight model such as a SVM or LR. This would have the effect of extracting important data. After which a more computationally complex DL model would be employed to detect more complex patterns, identifying malicious activity disguising itself as normal traffic. The literature reviewed shows particular promise for both K-means Clustering, SVMs and LR's as they are highly accurate classifiers [37] [41]. While LR and SVM models are much less computationally complex compared to K-means Clustering models, they are supervised learning models which may pose an issue to the longevity of the IDS considering the constantly evolving cyber threat landscape. However an argument can be made that the primary purpose of the ML techniques in this system would be merely to extract useful features rather than identifying the malicious traffic. After this preprocessing, a DL model would be implemented to analyse the preprocessed network traffic. CNNs and GANs have demonstrated effective performance when classifying network traffic [30] [41]. Additionally lightweight implementations of both GANs and CNNs have resulted in IDS models that were able to uphold their performance despite consuming fewer resources [3] [25]. Similarly, as discussed earlier, autoencoders have found success in being adapted for lightweight IDS as miniature models used for specialized singular tasks [20]. Making use of these approaches shall result in the implementation of these computationally expensive DL models in a low-resource environment feasible.

The aim of the LAIDS project would be to design a system which would minimize the Performance/Resource Trade off problem faced by most resource constrained IDSs. While other works have strived to solve the same problem and they have made headway, the Performance/Resource Trade off problem is still a major factor in IDS. Balancing performance while reducing computational overhead will be the main challenge faced by the LAIDS project. Ideally the system will be able to detect intrusions while having a negligible effect on network performance using the resources on a microcontroller.

5 CONCLUSIONS

In our increasingly connected world it is clear that IDS is going to remain as a tool integral to network security. Furthermore, with the evolution of cyber crime and increased network volume, TIDS are rapidly becoming obsolete and as such are being enhanced with AI models. This rapid AI enhancement of TIDS has highlighted the trade-off between IDS performance and network performance.

With the increasing adoption of IoT devices, optimizing AI-IDS solutions for low-resource environments is becoming increasingly

important. From the literature analysis conducted above, hybrid models have proven to improve IDS performance by combining models and techniques which compensate for each other's limitations and vise-versa. The literature further suggests that utilizing less computationally expensive ML models such as SVMs, LRs, and K-means clustering to preprocess the data prior to deploying more computationally complex DL models like Autoencoders, CNNs, and GANs to classify the network traffic as malicious or normal can enhance detection performance while maintaining efficiency.

The LAIDS project aims to address the security needs faced by the growing number of low-resource networks by developing a hybrid IDS that leverages the discussed models and techniques to achieve optimal performance in a low-resource environment. Future research should continue to focus on optimising DL models, exploring hybrid architectures, and refining resource-efficient AI techniques to enable effective intrusion detection in resource-constrained environments. The practical implementation of the IDS from the LAIDS project will ideally be deployable on low-resource microcontrollers, providing security for both IoT devices and low-resource networks.

REFERENCES

[1] M.T. Abdelaziz, A. Radwan, H. Mamdouh, et al. 2025. Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance. *Journal of Network and Systems Management* 33, 2 (2025). https://doi.org/10.1007/s10922-024-09874-0

[2] 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

[3] Tarek Ali, Amna Eleyan, Tarek Bejaoui, and Mohammed Al-Khalidi. 2024. Lightweight Intrusion Detection System with GAN-Based Knowledge Distillation. In *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*. 1–7. https://doi.org/10.1109/SmartNets61466.2024.10577682

[4] Cesare Alippi and Manuel Roveri. 2007. Reducing Computational Complexity in k-NN based Adaptive Classifiers. In *2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. 68–71. https://doi.org/10.1109/CIMSA.2007.4362541

[5] Md. Zahangir Alom, VenkataRamesh Bontupalli, and Tarek M. Taha. 2015. Intrusion detection using deep belief networks. In *2015 National Aerospace and Electronics Conference (NAECON)*. 339–344. https://doi.org/10.1109/NAECON.2015.7443094

[6] Hadjer Benmeziane. 2020. *Master Thesis - Comparison of Deep Learning Frameworks and Compilers*. Ph. D. Dissertation. https://doi.org/10.13140/RG.2.2.15094.22085

[7] 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

[8] Y. Chen and F. Yuan. 2022. Dynamic detection of malicious intrusion in wireless network based on improved random forest algorithm. In *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC)* (Dalian, China). 27–32. https://doi.org/10.1109/IPEC54454.2022.9777557

[9] Mohsen Eslamnezhad and Ali Yazdian Varjani. 2014. Intrusion detection based on MinMax K-means clustering. In *7'th International Symposium on Telecommunications (IST'2014)*. 804–808. https://doi.org/10.1109/ISTEL.2014.7000814

[10] 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

[11] J. Henriques, F. Caldeira, T. Cruz, and P. Simões. 2024. A survey on forensics and compliance auditing for critical infrastructure protection. *IEEE Access* 12 (2024), 2409–2444. https://doi.org/10.1109/ACCESS.2023.3348552

[12] Jayshree Jha and Leena Ragha. 2013. Intrusion detection system using support vector machine. *International Journal of Applied Information Systems (IJAIS)* 3 (2013), 25–30.

[13] Enamul Kabir, Jiankun Hu, Hua Wang, and Guangping Zhuo. 2018. A novel statistical technique for intrusion detection systems. *Future Generation Computer Systems* 79 (2018), 303–318.

[14] Ansam Khraisat and Ammar Alazab. 2021. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* 4 (2021), 1–27.

[15] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* 2 (12 2019). https://doi.org/10.1186/s42400-019-0038-7

[16] Yesi Novaria Kunang, Siti Nurmaini, Deris Stiawan, Ahmad Zarkasi, Firdaus, and Jasmir. 2018. Automatic Features Extraction Using Autoencoder in Intrusion Detection System. In *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. 219–224. https://doi.org/10.1109/ICECOS.2018.8605181

[17] Mohammed A. Mahdi. 2024. Secure and Efficient IoT Networks: An AI and ML-based Intrusion Detection System. In *2024 3rd International Conference on Artificial Intelligence For Internet of Things (AIIoT)*. 1–6. https://doi.org/10.1109/AIIoT58432.2024.10574789

[18] Batta Mahesh et al. 2020. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]* 9, 1 (2020), 381–386.

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

[20] 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).

[21] Mokhtar Mohammadi, Tarik A Rashid, Sarkhel H Taher Karim, Adil Hussain Mohammed Aldalwie, Quan Thanh Tho, Moazam Bidaki, Amir Masoud Rahmani, and Mehdi Hosseinzadeh. 2021. A comprehensive survey and taxonomy of the SVM-based intrusion detection systems. *Journal of Network and Computer Applications* 178 (2021), 102983.

[22] 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.

[23] M. Nivaashini, E. Suganya, and S. Sountharrajan. 2024. FEDDBN-IDS: Federated Deep Belief Network-Based Wireless Network Intrusion Detection System. *EURASIP Journal on Information Security* 2024, 8 (2024). https://doi.org/10.1186/s13635-024-00156-5

[24] 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.

[25] 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

[26] Kai Peng, Victor CM Leung, Lixin Zheng, Shangguang Wang, Chao Huang, and Tao Lin. 2018. Intrusion detection system based on decision tree over big data in fog environment. *Wireless Communications and Mobile Computing* 2018, 1 (2018), 4680867.

[27] Vinh Pham, Eunil Seo, and Tai-Myoung Chung. 2020. Lightweight Convolutional Neural Network Based Intrusion Detection System. *J. Commun.* 15, 11 (2020), 808–817.

[28] A. Ponmalar and V. Dhanakoti. 2022. An intrusion detection approach using ensemble Support Vector Machine based Chaos Game Optimization algorithm in big data platform. *Applied Soft Computing* 116 (2022), 108295. https://doi.org/10.1016/j.asoc.2021.108295

[29] Ujwala Ravale, Nilesh Marathe, and Puja Padiya. 2015. Feature Selection Based Hybrid Anomaly Intrusion Detection System Using K Means and RBF Kernel Function. *Procedia Computer Science* 45 (2015), 428–435. https://doi.org/10.1016/j.procs.2015.03.174 International Conference on Advanced Computing Technologies and Applications (ICACTA).

[30] 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

[31] Muhammad Shafiq, Zhaoquan Gu, Omar Cheikhrouhou, Wajdi Alhakami, and Habib Hamam. 2022. The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks. *Wireless Communications and Mobile Computing* 2022, 1 (2022), 8669348. https://doi.org/10.1155/2022/8669348 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/8669348

[32] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso. 2020. G-ids: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 376–385.

[33] 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

[34] Insoo Sohn. 2021. Deep belief network based intrusion detection techniques: A survey. *Expert Systems with Applications* 167 (2021), 114170.

[35] 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

[36] D. Strobel, D. Oswald, B. Richter, F. Schellenberg, and C. Paar. 2014. Microcontrollers as (In)Security Devices for Pervasive Computing Applications. *Proc. IEEE* 102, 8 (2014), 1157–1173. https://doi.org/10.1109/JPROC.2014.2325397

[37] Basant Subba, Santosh Biswas, and Sushanta Karmakar. 2015. Intrusion Detection Systems using Linear Discriminant Analysis and Logistic Regression. In *2015 Annual IEEE India Conference (INDICON)*. 1–6. https://doi.org/10.1109/INDICON.2015.7443533

[38] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 1–6. https://doi.org/10.1109/CISDA.2009.5356528

[39] Ankit Thakkar and Ritika Lohiya. 2021. A review on machine learning and deep learning perspectives of IDS for IoT: recent updates, security issues, and challenges. *Archives of Computational Methods in Engineering* 28, 4 (2021), 3211–3243.

[40] Miracle Udurume, Vladimir Shakhov, and Insoo Koo. 2024. Comparative Evaluation of Network-Based Intrusion Detection: Deep Learning vs Traditional Machine Learning Approach. In *2024 Fifteenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 520–525.

[41] 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

[42] R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. 2019. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access* 7 (2019), 41525–41550. https://doi.org/10.1109/ACCESS.2019.2895334

[43] Zhendong Wang, Yong Zeng, Yaodi Liu, and Dahai Li. 2021. Deep Belief Network Integrating Improved Kernel-Based Extreme Learning Machine for Network Intrusion Detection. *IEEE Access* 9 (2021), 16062–16091. https://doi.org/10.1109/ACCESS.2021.3051074

[44] Yousaf Bin Zikria, Rashid Ali, Muhammad Khalil Afzal, and Sung Won Kim. 2021. Next-Generation Internet of Things (IoT): Opportunities, Challenges, and Solutions. *Sensors* 21, 4 (2021). https://doi.org/10.3390/s21041174

Received 28 March 2025; revised TBC; accepted TBC