Enhanced Intrusion Detection in Robot Operating Systems via Grid Search Based Multi-Head Attention Stacked Convolutional Network

Muhammad Hamza Zafar[1] and Even Falkenberg Langås[1] and Muhammad Faisal Aftab[1] and Filippo Sanfilippo[1]

***Abstract*— This study presents a novel intrusion detection system (IDS) for Robot Operating Systems (ROS), utilising a hybrid neural network combining 1D Convolutional Neural Networks (CNNs) with Multi-head Attention (MHA). This approach effectively captures both local and global data features, essential for detecting security threats in ROS. The model architecture includes layers of 1D-CNNs for detailed temporal feature extraction, complemented by MHA to identify complex intrusion patterns. Extensive hyperparameter optimisation through grid search ensures optimal model performance. A key aspect of this research is the use of the recently introduced ROSIDS23 dataset, which provides a comprehensive and realistic benchmark for testing. The model demonstrated exceptional accuracy, achieving 99% in training and greater than 97% in testing, highlighting its efficacy in ROS security enhancement. These results and the utilisation of ROSIDS23 dataset mark significant advancements in the field of robotic security.**

Intrusion Detection System, Robot Operating System, Multi-head Attention, Convolutional Neural Network

I. Introduction

In an era where robotics are becoming integral to various sectors such as manufacturing, healthcare, and service industries, their security has emerged as a paramount concern. The integration of advanced robotics in these sectors not only enhances efficiency and productivity but also introduces new vulnerabilities. As these robots gain autonomy and connectivity, ensuring their secure operation becomes imperative. Intrusion Detection Systems (IDS) have thus become a crucial component of robotic security, designed to detect unauthorised access or anomalous behaviour, thereby safeguarding these sophisticated machines [1]. Robotic systems present security challenges distinct from traditional IT systems. These challenges stem from their real-time processing requirements, mobility, and direct interaction with the physical world. Unlike static IT networks, robots operate in dynamic environments and are subject to both cyber and physical threats [2]. This dual vulnerability necessitates a more comprehensive approach to security, one that can promptly detect and respond to a wide range of intrusion types. Machine Learning (ML) and Deep Learning (DL) technologies offer promising solutions to these security challenges. These technologies can analyse complex and voluminous data from various sensors, learning to detect patterns indicative of intrusions. The adaptability of ML and DL proves advantageous in identifying emerging intrusion attempts that traditional rule-based systems may overlook.

*This research is supported by the Artificial Intelligence, Biomechatronics and Collaborative Robotics research group at the Top Research Center Mechatronics (TRCM), University of Agder (UiA), Norway.

[1]Authors are with Dept. of Engineering Sciences, University of Agder, Grimstad, Norway muhammad.h.zafar@uia.no, even.falkenberg.langas@uia.no, faisal.aftab@uia.no,filippo.sanfilippo@uia.no

A. Literature Review

Intrusion detection using ML for robots has gained significant attention in recent years. ML techniques have been widely applied in the development of intrusion detection systems, showcasing their potential to enhance security measures in robotic systems [3]. The application of advanced ML methods for network intrusion detection has been a subject of extensive research, with a focus on developing scalable and adaptive learning approaches. Furthermore, the use of different ML algorithms, such as Support Vector Machine, Naive Bayes, Decision Tree, and ensemble learning, has been explored for building effective intrusion detection models [4]. These studies highlight the versatility of ML in addressing security challenges in interconnected systems, including robotics and cyber-physical systems.

The vulnerability of robotic platforms to cyber-attacks has been a growing concern, prompting researchers to explore the application of ML techniques for intrusion detection in robotics [5]. The need to address cyber-security threats in robotic systems has led to the investigation of hybrid ML techniques for intrusion detection, emphasising the significance of leveraging advanced ML methods to ensure the security of robotic platforms. Additionally, the use of computational intelligence in intrusion detection systems has been reviewed, providing insights into the effectiveness of different ML algorithms in detecting and mitigating security threats [6].

In the context of robotics, the development of intrusion detection systems using ML has been crucial in ensuring the security and integrity of robotic networks. The use of ML algorithms for intrusion detection in Software Defined Networks (SDNs) has been explored, highlighting the potential for leveraging ML and DL techniques in complex network environments [7]. Furthermore, the application of ML approaches for anomaly detection and classification in robotic systems has been investigated, emphasising the importance of developing robust intrusion detection models to safeguard robotic platforms from cyber-attacks [8].

B. Problem Statement and Contributions

While previous research has laid foundational work in the realm of Intrusion Detection Systems (IDS) for various networked and cyber-physical systems, a notable gap exists in the application of these systems specifically within the Robot Operating System (ROS).Several current IDS methods, although successful in traditional IT networks or general cyber-physical systems, fail to sufficiently handle the distinct complexities and operational dynamics inherent in ROS. These systems often lack the capability to efficiently process the high-volume, high-velocity data streams generated by robotic platforms, and thus may not effectively adapt to the continuously evolving patterns of cyber threats in robotic environments. Moreover, the integration of advanced ML techniques in IDS for ROS has been limited, with many models failing to simultaneously capture both the detailed temporal features and the broader, more complex intrusion patterns specific to ROS. This study addresses these shortcomings by introducing a hybrid neural network model that not only fills the gap in IDS research for ROS but also overcomes these weaknesses. By effectively combining 1D Convolutional Neural Networks (CNNs) with Multi-head Attention (MHA) mechanisms, this work not only tailors the intrusion detection process to ROS but also significantly enhances the accuracy and adaptability of the IDS, marking a substantial advancement in the field of robotic security. The contributions of this work are listed below:

1) Developed a unique hybrid model combining 1D CNNs and MHA, optimising feature extraction capabilities for both local and global aspects of ROS data, enhancing the detection of complex intrusion patterns.
2) Achieved exceptional performance with 99% accuracy in training and greater than 97% accuracy in testing, demonstrating the model's effectiveness in identifying a wide range of security threats in ROS environments on ROSIDS23 dataset provided recently.
3) Employed a comprehensive grid search mechanism for hyperparameter tuning, ensuring the model is finely tuned to the specificities of ROS, thereby maximising its detection capabilities and reliability.
4) Advanced the field of robotic security by introducing a cutting-edge IDS tailored for ROS, setting a new benchmark for intrusion detection in robotic operating systems and contributing valuable insights into the application of neural networks in cybersecurity.

II. Dataset Description and Pre-Processing

In this section detailed elaboration of ROSIDS23 benchmark dataset is carried out with the data pre-processing technique used for handling the missing data as well as NaN values.

A. Dataset Description

The ROSIDS23 dataset is a collection of network traffic data, specifically acquired from a network of autonomous robots [9]. This data, in pcap format, was analysed using the CICFlowMeter tool to extract relevant traffic features. The dataset is notable for encompassing four types of security breaches: unauthorised publishing, unauthorised subscribing, subscriber flooding, and a generic DoS (Denial of Service) attack. The first three attacks are unique to ROS environments, while the last is a common network security threat. ROSIDS23 is a diverse dataset, classified into multiple categories. Each record in the dataset is timestamped and contains eighty-three distinct features, along with a classification label. This label identifies the nature of the traffic, which could be one of five types: benign, DoS, unauthorised publishing, unauthorised subscribing, or subscriber flooding.

In terms of volume, the dataset is extensive and varied. The majority of the records, 62,511 in number, are classified as benign, indicating they are safe and non-malicious. The dataset also includes a significant number of malicious instances: 31,000 records of DoS attacks and 30,064 records of subscriber flood attacks. Additionally, it highlights two other critical security issues: unauthorised subscribing with 5289 instances and unauthorised publishing with 7817 instances. These figures emphasise the prevalence of each category within the dataset. Fig. 1 provides a graphical representation of these statistics, offering a comprehensive view of the dataset's composition.

TABLE I
THE DETAILS OF THE PROPOSED DATASET: ROSIDS23 DATASET.

Name of Dataset	ROSIDS23
Type of Dataset:	Multi-class
Total Features:	83
Total Classes:	5
Class Labels:	Benign, DoS, Unauthorised Publish, Unauthorised Subscribe, Subscriber Flood

B. Dataset Pre-Processing

In the data pre-processing section, we explore the application of K-Nearest Neighbors (KNN) imputation, an instance-based or lazy learning method, for handling missing data in a dataset [10]. This technique is particularly effective when data is missing completely at random (MCAR) or missing at random (MAR).

The core principle of KNN imputation is the assumption that similar data points are located near each other in the feature space. To determine the 'closeness' of data points, various distance metrics such as Euclidean, Manhattan, or Minkowski distances are employed. The fundamental strategy involves identifying 'k' nearest neighbours to a data point with missing values and utilising these neighbours to estimate the missing values.

The algorithm for KNN Imputation involves several key steps:

1) **Selection of 'k'**: The number of nearest neighbours, 'k', is chosen carefully, as it significantly influences the accuracy of the imputation.
2) **Distance Metric**: A suitable distance metric is selected to measure the distances between data points, aiding in the identification of nearest neighbours.

3) **Finding Nearest Neighbours**: For each data point that has missing values, the algorithm identifies 'k' nearest neighbours from the subset of data points that do not have missing values.
4) **Imputation**: The missing values are imputed using the mean, median, or mode of the 'k' nearest neighbours' corresponding values, depending on the data type (mean is used in this study).

From a mathematical perspective, let X represent the dataset with missing values, and x_i be a data point within a d-dimensional space that has missing values. To calculate the distance between two points x_j and x_k when not all attributes are available, we use a chosen metric, such as the Euclidean distance, defined as:

$$D(x_j; x_k) = \sqrt{\sum_{l=1}^{} w_l \quad (x_{jl} \quad x_{kl})^2} \tag{1}$$

Where $D(x_j; x_k)$ represents the distance between x_j and x_k; x_{jl} and x_{kl} are the l-th attributes of x_j and x_k respectively; and w_l is a weight that is 1 if both x_{jl} and x_{kl} are present and 0 otherwise. This modification ensures that only the non-missing attributes contribute to the distance calculation. After identifying the k nearest neighbours, the missing value m in x_i is imputed as the mean:

$$m = \frac{1}{k} \sum_{j=1}^{} x_{jn} \tag{2}$$

Where x_{jn} represents the values of the n-th attribute (the one that's missing in x_i) from the 'k' nearest neighbours. This approach assumes that the missing data point can be reasonably estimated by the average of the similar non-missing data points.

After applying KNN imputation to handle missing data in the dataset, the next critical step in data preprocessing is normalisation, which is essential for scaling numeric data to a specific range. Min-Max normalisation, a commonly used normalisation technique, rescales the feature values to a fixed range, typically between 0 and 1. Min-Max normalisation is particularly useful when the dataset features have different scales and ranges, as it ensures that each feature contributes equally to the analysis. This is crucial for improving the performance and accuracy of many ML algorithms.

The formula for Min-Max Normalisation is as follows:

$$x' = \frac{x \quad \min(x)}{\max(x) \quad \min(x)} \tag{3}$$

where x is the original value, $\min(x)$ is the minimum value of the feature across the dataset, and $\max(x)$ is the maximum value of the feature. The result, x', is the normalised value, rescaled to the range [0, 1].

III. Proposed Technique

A. Convolutional Neural Network

A 1-dimensional Convolutional Neural Network (1D-CNN) is a specialised neural network model that excels in processing spatial data [11]. Its architecture is specifically designed to identify patterns in sequences, making it particularly useful for applications such as audio signal analysis and other forms of sequential data. The 1D-CNN comprises various layers, each serving a unique function in the data processing pipeline. The input layer receives one-dimensional data, such as audio signals, which are then processed through several convolutional layers. These layers are composed of filters or kernels, applied across the input data to produce feature maps. The convolution operation in these layers is mathematically represented as:

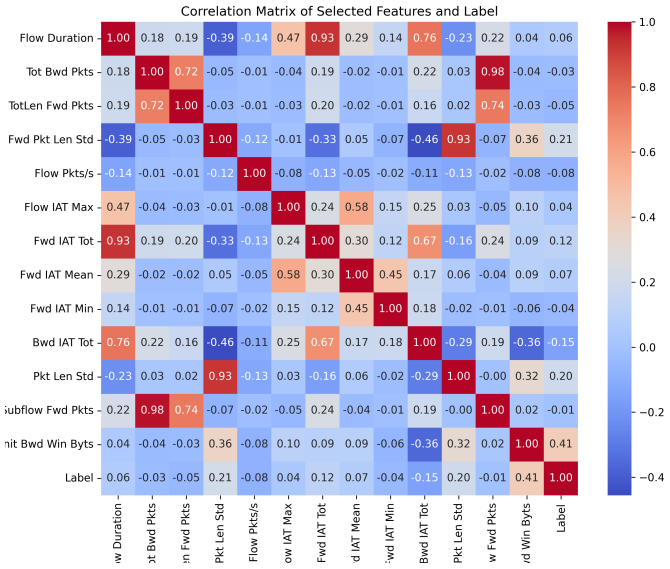

Fig. 1. Correlation of Best Features.

$$y[k] = \sum_{n=-\infty}^{} x[n] \quad h[k \quad n] \tag{4}$$

where $x[n]$ is the input data, $h[m]$ is the filter, and $y[k]$ is the output.

Following convolution, the output is typically passed through a non-linear activation function, such as the Rectified Linear Unit (ReLU), defined as:

$$f(x) = \max(0; x) \tag{5}$$

Pooling layers follow, which reduce data dimensionality and computational complexity. A common approach in pooling is max pooling, where the maximum value within a certain neighbourhood is selected.

The network also includes fully connected layers, where every input is linked to each output by a weight, represented as:

$$y_j = \quad \left(\sum_{i=1}^{} W_{ji} x_i + b_j \right) \tag{6}$$

Here, W_{ji} denotes the weight from input i to output j, b_j is the bias, and is the activation function. The final layer, the output layer, typically utilises a softmax function

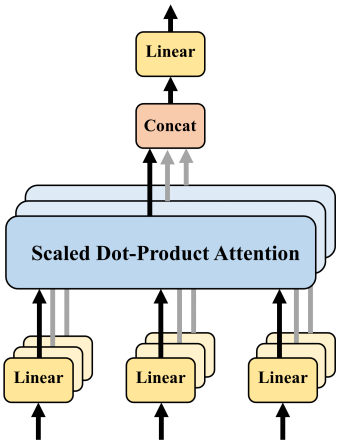

Fig. 2. Architecture of a Multi Head Attention Module.

in classification tasks to transform outputs into probability scores.

B. Multi-head Attention (MHA)

The MHA mechanism is a pivotal component of transformer models, widely used in natural language processing [12]. This mechanism enables the model to simultaneously process information from different representation subspaces at different positions, leading to a more nuanced understanding of the context. In the MHA mechanism, an attention function is applied multiple times in parallel. The attention function maps a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values. Each weight assigned to a value is computed based on the query and the corresponding key. Mathematically, the attention weights are computed using the scaled dot-product attention, which is defined as:

$$\text{Attention}(Q; K; V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (7)$$

where Q, K, and V are matrices representing queries, keys, and values, respectively, and d_k is the dimension of the key vectors. In the MHA, the above attention mechanism is applied multiple times in parallel, with each 'head' projecting the queries, keys, and values with different, learned linear projections. This parallel application of attention allows the model to jointly attend to information from different representation subspaces. The outputs of each head are then concatenated and linearly transformed into the expected dimension. This is mathematically represented as:

$$\text{Multi-head}(Q; K; V) = \text{Concat}(\text{head}_1; \dots; \text{head}_h)W^O \quad (8)$$

$$\text{where head}_i = \text{Attention}(QW_i^Q; KW_i^K; VW_i^V) \quad (9)$$

Here, W_i^Q, W_i^K, W_i^V, and W^O are parameter matrices, and h is the number of heads.

C. Multi-head Attention-based Stacked Convolutional Network

The proposed hybrid neural network model, integrating 1D CNNs with MHA (MHA-SConv), emerges as a highly effective approach for Intrusion Detection Systems (IDS). This model adeptly balances the extraction of local features through 1D-CNNs with the global contextual understanding afforded by the MHA mechanism, making it particularly suitable for the subtle demands of intrusion detection.

In the field of IDS, the initial layers of the model, composed of 1D convolutional layers followed by max-pooling, play a pivotal role in extracting local and temporal features from input data. This could include network traffic patterns, system logs, or other relevant sequential data. These convolutional layers are particularly skilled at identifying specific patterns, like sequences of network packets indicative of a scan or signatures of known attacks, providing a granular view of network activities. Following the convolutional layers, the model employs the MHA mechanism, a crucial component for understanding broader context and dependencies within the data. Intrusion detection often involves identifying complex attack patterns that unfold over time, with relevant indicators scattered throughout the network traffic or logs. The MHA mechanism enables the model to assess these distant but interconnected segments of data, weighing them appropriately when predicting potential threats. This aspect is vital for recognising sophisticated cyber threats that may not present immediate or obvious signatures.

The synergy between the localised feature detection capabilities of the CNNs and the global perspective offered by the MHA results in a robust and refined intrusion detection tool. This hybrid model offers better accuracy as compared to the methods that rely solely on one of these approaches. It effectively reduces false positives and negatives, a critical factor in IDS where the cost of misidentifying threats can be high. This model's adaptability to different network environments and attack vectors is an added advantage. It learns from both the specific characteristics of certain attacks (local features) and broader attack strategies or unusual activities that manifest over time (global patterns). This adaptability is crucial in an ever-evolving cybersecurity landscape. The efficiency of this hybrid model in processing sequential data makes it a viable candidate for real-time intrusion detection. It can quickly analyse streaming data, enabling timely identification and response to potential network threats. In conclusion, the combination of 1D-CNNs and MHA in a single model offers a comprehensive and dynamic approach to intrusion detection, addressing both immediate and complex, context-dependent cyber threats with high accuracy and reliability.

D. Grid Search based MHA-SConv

The incorporation of a grid search mechanism for hyperparameter tuning significantly enhances the performance and efficacy of the proposed hybrid neural network model. Grid search is a structured approach to hyperparameter tuning that methodically builds and evaluates a model for each combination of algorithm parameters specified in a grid. This

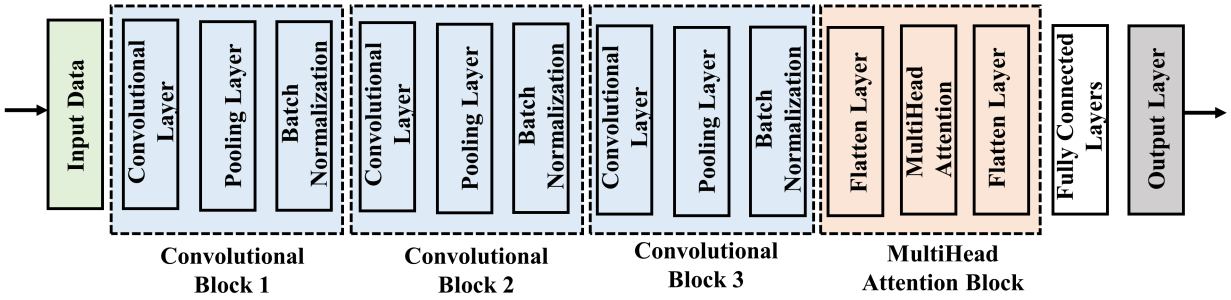

Fig. 3. Architecture of Proposed Multi-head Attention Stacked Convolutional Network for IDS.

technique is instrumental in identifying the most optimal set of parameters for the model, ensuring that it performs at its best. The proposed model employs grid search to explore varied hyperparameter combinations (e.g., convolutional layers, kernel size, attention heads). Each combination represents a distinct configuration, trained and evaluated on a validation set. Performance metrics such as accuracy, precision, recall, and F1-score are employed to gauge intrusion detection effectiveness. Despite being time-consuming, this exhaustive search identifies the optimal configuration for peak performance.

IV. Results and Discussion

Fig. 4 presents the training and validation accuracy curves for an Intrusion Detection System (IDS) developed using a MHA Stacked Convolutional Network architecture within a Robot Operating System (ROS) environment. The dataset used is divided into 60-10-30 meaning 60% training, 10% validation, and 30% testing data. The x-axis represents the number of epochs, ranging from 0 to 30, which denotes the iterations over the entire dataset during the learning process. The y-axis quantifies the accuracy, ranging from 0 to 1, which is a measure of the model's performance in correctly identifying security breaches. The red dashed line depicts the training accuracy, indicating how well the model learns from the dataset it was trained on, achieving a peak accuracy of 99.1%. The blue dashed line shows the validation accuracy, which measures the model's ability to generalise to new, unseen data, with a maximum accuracy of 97.43%. The high accuracy values suggest a well-fitted model, and the proximity of the two curves indicates that the model generalises well and is not overfitting. However, the slight divergence between the curves could be an area to investigate for potential overfitting as the epochs increase. Overall, the system demonstrates excellent learning capabilities and robustness in detecting intrusions in the ROS environment.

TABLE II
Comparative analysis of competing techniques for IDS in ROSIDS23.

Technique	Accuracy	Precision	Recall	F1-Score
MHA-SConv	0.9707	0.9679	0.9573	0.9675
SCNN	0.9466	0.9418	0.9488	0.9403
DNN	0.9290	0.9121	0.9182	0.9198
SVM	0.8188	0.8080	0.8051	0.8161

In the Table. II, a comparative analysis of different techniques for an Intrusion Detection System (IDS) in a Robot Operating System (ROS) environment, denoted as ROSIDS23, is presented. The techniques evaluated include the proposed MHA Stacked Convolutional Network (MHA-SConv), a Stacked Convolutional Neural Network (SCNN), a Deep Neural Network (DNN), and a Support Vector Machine (SVM). The evaluation metrics include Accuracy, Precision, Recall, and F1-score, which are critical for assessing the performance of IDS models. The proposed MHA-SConv model outperforms the other models across all metrics, with an Accuracy of 97.07%, Precision of 96.79%, Recall of 95.73%, and an F1-score of 96.75%. This indicates a well-balanced model with high reliability in correctly identifying intrusions (high precision), a strong ability to detect a high number of actual intrusions (high recall), and a harmonic mean of precision and recall (high F1-score). Compared to MHA-SConv, SCNN exhibits moderately high metrics but lags, especially in Accuracy, Precision, and F1-score, suggesting less effectiveness in distinguishing network traffic classes. DNN shows a further decrease, notably in Precision, implying a higher false positive rate compared to MHA-SConv and SCNN. SVM has the lowest scores across all metrics, indicating less effective capture of data complexity compared to neural network-based models, particularly in ROSIDS23. The findings suggest that the proposed MHA-SConv demonstrates exceptional performance in identifying intrusions within ROS environments, positioning it as a promising cybersecurity solution for robotic systems.

A. Discussion

The results demonstrate the remarkable performance of the proposed MHA Stacked Convolutional Network architecture in accurately detecting intrusions within the Robot Operating System (ROS) environment. The high training and validation accuracy, peaking at 99.1% and 97.43% respectively, indicate the model's exceptional learning capabilities and generalization to unseen data. Furthermore, the comparative analysis across different techniques solidifies the MHA-SConv model's superiority, outperforming other models like Stacked Convolutional Neural Network (SCNN), Deep Neural Network (DNN), and Support Vector Machine (SVM) in critical