computers

MDPI

Review

Securing Mobile Edge Computing Using Hybrid Deep Learning Method

Olusola Adeniyi [1], Ali Safaa Sadiq [2], Prashant Pillai [1], Mohammad Aljaidi [3] and Omprakash Kaiwartya [2,*]

[1] School of Engineering, Computing and Mathematical Sciences, University of Wolverhampton, Wolverhampton WV1 1LY, UK; o.adeniyi@wlv.ac.uk (O.A.); p.pillai@wlv.ac.uk (P.P.)
[2] Department of Computer Science, Nottingham Trent University, Clifton Campus, Nottingham NG11 8NS, UK; ali.sadiq@ntu.ac.uk
[3] Computer Science Department, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan; mjaidi@zu.edu.jo
* Correspondence: omprakash.kaiwartya@ntu.ac.uk

Abstract: In recent years, Mobile Edge Computing (MEC) has revolutionized the landscape of the telecommunication industry by offering low-latency, high-bandwidth, and real-time processing. With this advancement comes a broad range of security challenges, the most prominent of which is Distributed Denial of Service (DDoS) attacks, which threaten the availability and performance of MEC's services. In most cases, Intrusion Detection Systems (IDSs), a security tool that monitors networks and systems for suspicious activity and notify administrators in real time of potential cyber threats, have relied on shallow Machine Learning (ML) models that are limited in their abilities to identify and mitigate DDoS attacks. This article highlights the drawbacks of current IDS solutions, primarily their reliance on shallow ML techniques, and proposes a novel hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model for intrusion detection as a solution against DDoS attacks in the MEC environment. The proposed hybrid AE–MLP model leverages autoencoders' feature extraction capabilities to capture intricate patterns and anomalies within network traffic data. This extracted knowledge is then fed into a Multi-Layer Perceptron (MLP) network, enabling deep learning techniques to further analyze and classify potential threats. By integrating both AE and MLP, the hybrid model achieves higher accuracy and robustness in identifying DDoS attacks while minimizing false positives. As a result of extensive experiments using the recently released NF-UQ-NIDS-V2 dataset, which contains a wide range of DDoS attacks, our results demonstrate that the proposed hybrid AE–MLP model achieves a high accuracy of 99.98%. Based on the results, the hybrid approach performs better than several similar techniques.

Keywords: mobile edge computing; DDoS; machine learning; cyber security

Citation: Adeniyi, O.; Sadiq, A.S.; Pillai, P.; Aljaidi, M.; Kaiwartya, O. Securing Mobile Edge Computing Using Hybrid Deep Learning Method. *Computers* **2024**, *13*, 25. https://doi.org/10.3390/computers13010025

Academic Editor: Paolo Bellavista

Received: 19 December 2023
Revised: 10 January 2024
Accepted: 12 January 2024
Published: 16 January 2024

1. Introduction

The advent of Mobile Edge Computing (MEC) has signaled a paradigm shift in the design and management of wireless communication systems, especially in the current environment of pervasive network connectivity and expanding data-driven applications. This innovative idea involves assigning computational work and data processing to edge nodes located near end users, reducing latency and increasing system effectiveness. As MEC gains popularity for its potential to revolutionize network architecture, it simultaneously presents a variety of complex cyber-security concerns that demand in-depth scholarly research.

The MEC framework faces a wide range of cyber-security challenges. One prominent concern is the intensification of DDoS attacks. A DDoS attack, characterized by the orchestrated flooding of network resources, poses a significant threat to the reliability and availability of edge-hosted services [1]. The dynamic and distributed nature of MEC environments, encompassing various devices connected through heterogeneous networks, is well suited for DDoS attacks. Therefore, the threat of DDoS attacks on MEC networks

highlights the need for effective and flexible cyber-security measures. These measures should include proactive detection, quick response, and mitigation of malicious activities. Additionally, MEC networks should be monitored continuously to identify any suspicious activities and take necessary actions to prevent them.

In the field of cyber security, Intrusion Detection Systems (IDSs) are used to carefully monitor network processes, in order to identify and prevent future security breaches [2]. These solutions carefully examine data flow and engage in a discriminating analysis of patterns that differ from the norm, actions that raise suspicion, or attack signs that may be easily identified. IDS solutions play a crucial and useful role in bolstering a network's defensive perimeter and minimizing the negative effects of cyber attacks. However, the use of shallow machine learning is a notable drawback when it comes to IDSs.

While these techniques have been widely employed for anomaly detection and pattern recognition, they often exhibit limitations in handling the complex and evolving nature of modern cyber threats. Shallow machine learning models, such as SVM, decision trees, and K-nearest neighbours are characterized by their relative simplicity and shallow hierarchical structures. These algorithms may struggle to discern intricate patterns and subtle deviations that are indicative of sophisticated attacks [3].

Deep learning, a subfield of machine learning, has emerged as a potent and promising avenue for enhancing IDS [4]. Unlike traditional shallow machine learning techniques, deep learning leverages multi-layered neural networks to automatically extract intricate features and patterns from complex data, enabling it to more accurately capture the subtle and dynamic nature of contemporary cyber threats.

In the age of information proliferation, where data are generated at an unprecedented pace across diverse domains, the emergence of high-dimensional datasets is creating new challenges for data analysis [5]. Analyzing data in high-dimensional spaces requires new techniques and tools to uncover patterns and extract meaningful insights. As the dimensionality of data increases, the efficiency, interpretability, and generalization capacity of machine learning models, particularly deep learning architectures, can be profoundly impacted. As a result of this intricate interplay among the high-dimensional data environment, the need for optimal accuracy and efficient model performance calls for innovative approaches to feature reduction [6].

AE is an example of a feature reduction method with the overarching purpose of identifying the salient features in data that contain the core information necessary for model learning [7]. This transformative process seeks to strike a delicate balance between retaining meaningful attributes and discarding redundant or noise-laden variables. By condensing the data while preserving its intrinsic structure, these techniques not only facilitate computational efficiency but also hold the potential to significantly bolster the accuracy and generalization prowess of deep learning models.

The following are this article's primary contributions.

- AE is proposed to reduce the dimension of the features. This technique compresses large network traffic data into a lower dimension without sacrificing valuable essential network data.
- A hybrid AE–MLP is proposed for network traffic classification. This method analyzes the network traffic and classifies it based on a hybrid combination of AE and MLP. The AE is used to learn the hidden representation of the network data. Subsequently, the MLP is used to classify the network traffic.
- A comprehensive set of experiments was conducted with the NF-UQ-NIDS-V2 dataset to verify the performance of the hybrid AE–MLP model in binary classification scenarios.
- The proposed model's results were compared with other similar models to show its performance.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the dataset, its preprocessing, and the preliminary analysis carried out on it. Section 4 details the proposed hybrid AE–MLP model along with the dimension reduction process and classification method. Section 5 discusses the experimental setup,

the results, the performance of the proposed model, as well as comparisons with existing models of a similar nature. Conclusions and future works have been presented in Section 6

2. Related Work

Detection and classification of DDoS attacks are crucial to preventing malicious activities on a network. It is necessary to have a system in place that can detect and classify these attacks in real time. In addition, this system should be able to identify the different types of DDoS attacks and respond accordingly.

A series of approaches have been utilized by researchers to address the detection and classification of DDoS attacks. For instance, Wani et al.'s [8] study focused on the analysis and detection of DDoS attacks using ML. The study utilized some ML algorithms, including Naïve Bayes, random forest, and support vector machine. Overall, SVM performed better than others with an accuracy of 99.7%. The study identified SVM as an effective algorithm for detecting DDoS attacks with high accuracy. Bindra and Sood [9] also examined five ML models to determine which one would produce the best DDoS detection results. The study revealed that random forest achieved the highest accuracy of 96%.

Khare et al. [10] presented DT as a model for DDoS attack detection. The process involves extracting features, and the information obtained is calculated. Using the information obtained, a decision tree is constructed that identifies DDoS attacks and categorizes them. The author claimed the model achieved a 90.2% success rate. Kousar et al. [11] also show that the decision tree outperforms SVM and Naive Bayes.

Arshi et al. [12] presented a survey of machine learning techniques to detect DDoS attacks. The study discussed techniques, such as SVM, Naive Bayes, and DT. The author also provided more information on different types of DDoS attacks. The author concluded that the use of machine learning techniques is essential for understanding DDoS attacks and taking the necessary precautions to minimize them.

It is worth noting that all these methods are based on shallow machine learning, which has been widely studied and deployed for years, and has shown success in many ways. However, in recent times, the use of deep learning has been on the increase. This is partly due to the fact that shallow ML methods—though they may have high accuracy—perform poorly when used with large datasets [13]. Generally, in ML, dataset features play an important role in the outcome of the model. Getting the appropriate features that well represent the dataset is of uttermost importance. Dimensionality reduction is an important step in data pre-processing [14]. It helps to reduce noise, eliminate irrelevant features, and reduce the computational complexity of algorithms. It also helps to increase the accuracy of the model by removing redundant features. Researchers in recent times have utilized various methods to reduce high data dimensions and achieve the necessary accuracy.

For instance, Elsayed et al. [15] proposed Ddosnet to address DDoS attacks in SDN environments. The authors utilized a combination of RNN and autoencoder to build a model and evaluated the model's performance with the CIDDoD2019 dataset. According to the author, the results showed that the method offered a substantial improvement over alternative methods in detecting attacks.

Yuan et al. [16] proposed an approach called deep defense. The approach focuses on the optimal feature representation of the dataset. Recurrent deep neural networks were used to identify patterns from batches of network traffic and track network attack activity. The author claimed the model achieved a better performance when compared with other ML models. However, the dataset used is old and may not contain the latest form of attacks.

Mushtaq et al. [17] explored the feasibility of designing an effective intrusion detection system for the protection of networks from cyber attacks. The researchers propose a hybrid framework that combines deep autoencoder (AE) with long short-term memory (LSTM) and bidirectional long short-term memory (Bi-LSTM) for intrusion detection. The author validated the performance of the proposed model on the well-known NSL-KDD dataset. The results indicate that the proposed AE-LSTM framework outperforms other deep and shallow machine learning techniques, as well as recently reported methods.

Lee and Park [18] addressed the development of a high-performance network intrusion detection system (NIDS) using deep learning, specifically focusing on situations where there are significant imbalances between normal and abnormal network traffic data. The researchers proposed an AE-CGAN (Autoencoder-Conditional GAN) model that combines autoencoders and generative adversarial networks to improve intrusion detection in data-imbalanced situations [19]. The model's performance was evaluated on the CICIDS2017 dataset. According to the author, the proposed model effectively reduces false detections and improves the detection of rare classes, leading to better intrusion detection performance.

To improve the accuracy and efficiency of Network Intrusion Detection Systems (NIDSs) using deep learning techniques, Kunang et al. [20] combined a DNN with a Pretraining Technique with Deep Autoencoder (PTDAE) to create a deep learning Intrusion Detection System (IDS [21]. An automated optimal hyperparameter procedure was developed through grid search and random search techniques. The pretraining phase involves applying three feature extraction methods: Deep Autoencoder (DAE), Autoencoder, and Stack Autoencoder (SAE). According to the author, the results show that the DAE method provides the best performance, outperforming previous approaches in terms of performance metrics in multiclass classification.

Ultimately, to effectively address DDoS attacks in a high-dimensional data environment, there is a need for feature reduction to enhance the quality and efficiency of deep learning outcomes as demonstrated by the state-of-the-art research carried out.

3. Dataset Description, Preprocessing, and Preliminary Analysis

The NF-UQ-NIDS-V2 dataset contains network traffic data for intrusion detection generated from flows from multiple network setups and different attack settings. It combines four datasets (UNSW-NB15, BoT-IoT, ToN-IoT, and CSE-CIC-IDS2018.) released between 2015 and 2018 [22]. The dataset has a total of 11,994,893 records, out of which 9,208,048 (76.77%) are benign flows and 2,786,845 (23.23%) are attacks. The dataset contains 43 features, and the size is 13.73 GB.

The features of the dataset are listed in Table 1. The class label is a binary variable indicating whether the traffic is normal or benign traffic (labeled as 0) or malicious traffic (labeled as 1). In the data preprocessing stage, the study identified and selected relevant features that contributed the most to the modeling task. Redundant features, such as 'Attack' and 'Dataset', were removed to simplify the dataset and improve model performance. The dataset was split into training, validation, and testing sets to evaluate model performance. The study also ensures that the splitting maintains the original data distribution to avoid bias.

Table 2 provides a breakdown of the different types of attacks from the 100,000 rows of the NF-UQ- NIDS-V2 dataset utilized for this experiment. It also shows the distribution of the nineteen different attacks in the dataset and it reveals DoS and DDoS at the top of the list.

A preliminary analysis conducted on the NF-UQ-NIDS-V2 dataset using the Pycaret Python library is shown below. The Pycaret Python library automates ML workflows, allowing experiments to be conducted more quickly [23]. This library contains a comprehensive set of tools for data exploration, preprocessing, feature engineering, model training, and evaluation. It contains several pre-trained models that can be used to build a model quickly and it provides visualization capabilities to help users better understand their models.

Table 3 shows the model's settings. It shows a binary classification type and the shape of the training and test dataset. It also shows the model utilized, StratifiedKFold, which exposes the model to various subsets of the data. Stratifiedkfold is a variant of the k-fold cross-validation technique. It is particularly useful in a situation where samples in some classes are more than others, a situation known as class imbalance. Class imbalance can lead to bias in the training model. This technique splits data into K-folds or subsets. It

ensures that each fold contains roughly the same number of samples from each class as the original data. Consequently, each class in the original dataset is adequately represented in the training and validation sets. Hence, the likelihood of the model's evaluation being biased is minimized.

Table 1. NF-UQ-NIDS V2 dataset features.

Number	Feature	Number	Feature
1	IPV4_SRC_ADDR	25	RETRANSMITTED_IN_BYTES
2	L4_SRC_PORT	26	RETRANSMITTED_IN_PKTS
3	IPV4_DST_ADDR	27	RETRANSMITTED_OUT_BYTES
4	L4_DST_PORT	28	RETRANSMITTED_OUT_PKTS
5	PROTOCOL	29	SRC_TO_DST_AVG_THROUGHPUT
6	L7_PROTO	30	DST_TO_SRC_AVG_THROUGHPUT
7	IN_BYTES	31	NUM_PKTS_UP_TO_128_BYTES
8	IN_PKTS	32	NUM_PKTS_128_TO_256_BYTES
9	OUT_BYTES	33	NUM_PKTS_256_TO_512_BYTES
10	OUT_PKTS	34	NUM_PKTS_512_TO_1024_BYTES
11	TCP_FLAGS	35	NUM_PKTS_1024_TO_1514_BYTES
12	CLIENT_TCP_FLAGS	36	TCP_WIN_MAX_IN
13	SERVER_TCP_FLAGS	37	TCP_WIN_MAX_OUT
14	FLOW_DURATION_MILLISECONDS	38	ICMP_TYPE
15	DURATION_IN	39	ICMP_IPV4_TYPE
16	DURATION_OUT	40	DNS_QUERY_ID
17	MIN_TTL	41	DNS_QUERY_TYPEDNS_QUERY_TYPE
18	MAX_TTL	42	DNS_TTL_ANSWER
19	LONGEST_FLOW_PKT	43	FTP_COMMAND_RET_CODE
20	SHORTEST_FLOW_PKT	44	Label
21	MIN_IP_PKT_LEN	45	Attack
22	MAX_IP_PKT_LEN		
23	SRC_TO_DST_SECOND_BYTES		
24	DST_TO_SRC_SECOND_BYTES		

Table 2. Attack distribution.

Class	Count
DoS	23,542
Benign	32,986
Scanning	4955
DDoS	28,742
Xss	3217
Bot	197
Reconnaissance	3444
Password	1519
Fuzzers	25
Injection	930
Theft	1
Brute Force	190
Infilteration	161
Exploits	34
Generic	19
Analysis	2
Backdoor	23
Mitm	8
Shellcode	2
Ransomware	3

Table 3. Model set up.

	Description	Value
0	Session id	539
1	Target	Label
2	Target type	Binary
3	Original data shape	(100,000, 44)
4	Transformed data shape	(100,000, 44)
5	Transformed train set shape	(70,000, 44)
6	Transformed test set shape	(30,000, 44)
7	Numeric features	43
8	Preprocess	True
9	Imputation type	simple
10	Numeric imputation	mean
11	Categorical imputation	mode
12	Fold Generator	StratifiedKFold
13	Fold Number	10
14	CPU Jobs	−1
15	Use GPU	False
16	Log Experiment	False
17	Experiment Name	clf-default-name
18	USI	5465

Table 4 shows the results of using fifteen different models on the dataset. The models included various algorithms, such as random forest classifier, decision tree, and K-nearest neighbors, among others. The models were evaluated in terms of accuracy, precision, recall, and F1 score. The results showed that some models performed better than others. In particular, the random forest classifier gave the best performance with an accuracy of 99.59%, recall of 99.63%, precision of 99.75%, and F1-score of 99.69%. This result shows the potential of the random forest classifier to accurately classify data.

Table 4. Model comparison result.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (s)
rf	Random Forest Classifier	0.9959	0.9995	0.9963	0.9975	0.9969	0.9906	0.9906	7.4260
lightgbm	Light Gradient Boosting Machine	0.9958	0.9998	0.9964	0.9974	0.9969	0.9905	0.9905	5.1650
et	Extra Trees Classifier	0.9956	0.9991	0.9962	0.9972	0.9967	0.9901	0.9901	5.4460
xgboost	Extreme Gradient Boosting	0.9954	0.9998	0.9962	0.9968	0.9965	0.9895	0.9895	1.6770
dt	Decision Tree Classifier	0.9937	0.9926	0.9958	0.9948	0.9953	0.9857	0.9857	0.6430
knn	K Neighbors Classifier	0.9917	0.9957	0.9931	0.9945	0.9938	0.9811	0.9811	7.9770
gbc	Gradient Boosting Classifier	0.9908	0.9990	0.9919	0.9944	0.9931	0.9792	0.9793	18.7560
ada	Ada Boost Classifier	0.9818	0.9975	0.9831	0.9896	0.9863	0.9589	0.9589	4.4750
ada	Ada Boost Classifier	0.9818	0.9975	0.9831	0.9896	0.9863	0.9589	0.9589	4.4750
lda	Linear Discriminant Analysis	0.8571	0.9499	0.8275	0.9531	0.8859	0.6974	0.7099	0.9540
ridge	Ridge Classifier	0.8569	0.0000	0.8281	0.9522	0.8858	0.6968	0.7091	0.1760
qda	Quadratic Discriminant Analysis	0.7656	0.7261	0.9876	0.7454	0.8496	0.3630	0.4503	0.5010
svm	SVM—Linear Kernel	0.6870	0.0000	0.9418	0.6973	0.8013	0.1358	0.1789	8.9140
nb	Naive Bayes	0.6711	0.8242	0.9998	0.6708	0.8029	0.0041	0.0154	0.2720
dummy	Dummy Classifier	0.6701	0.5000	1.0000	0.6701	0.8025	0.0000	0.0000	0.1170
lr	Logistic Regression	0.3299	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	1.7520

Figure 1 shows the confusion matrix result for the dataset. The result shows that 20,034 of the attacks were correctly classified, and 70 of the benign class were misclassified. This shows an appreciable performance of the model.

Figure 2 shows the feature importance of the dataset, highlighting the most significant features. This can help identify features that are most relevant to the problem, which can then be used to build a more accurate model. It shows the source address having the highest importance of 0.14, indicating that this feature has the most significant impact on

the outcome or predicting the target variable. This is followed by the destination address, with 0.12, a higher importance score. This shows that this feature holds more influence on the outcome or predicting the target variable when compared to other features. The diagram also shows flow duration in the third place of importance, with 0.09. This position shows relatively higher importance, and this suggests the feature plays a notable role in predicting the target variable.

The source and destination IP are important features relevant to the detection of DDoS attacks [24]. The source and destination IP can be used to track the attackers and determine the origin of the attack. Additionally, they can be used to block the source and prevent further attacks. It is possible that these IP addresses have been used and blocked elsewhere in the past, which provides a quick warning sign. IP addresses can also be used to identify malicious patterns, such as a large number of requests coming from a single source.

In the same way, SHAP was used to analyze the dataset. SHAP is a Python package that uses game theory to explain machine learning models [25]. It estimates how individual features contribute to the outcome of a machine learning model. This tool is useful for analyzing a model's performance and determining which features are most important for predicting the outcome accurately. As shown in Figure 3, the destination address, source address, and duration are at the top of the list of the features with the most significant impact on the outcome of the model. Based on the absolute SHAP value, it is clear that Stripped_IPV4_DST made a substantial contribution to prediction, followed by Stripped_IPV4_SRC. The model is either more likely to increase or decrease its prediction based on the colors red and blue, respectively.

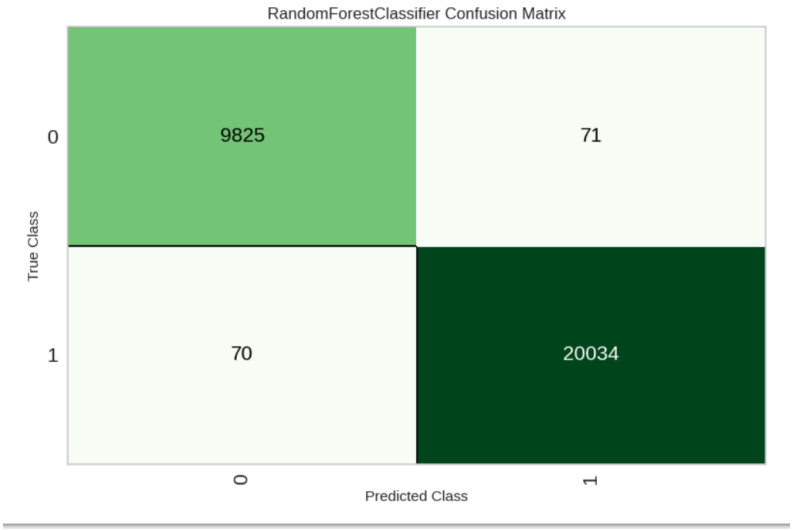

Figure 1. Confusion matrix result.

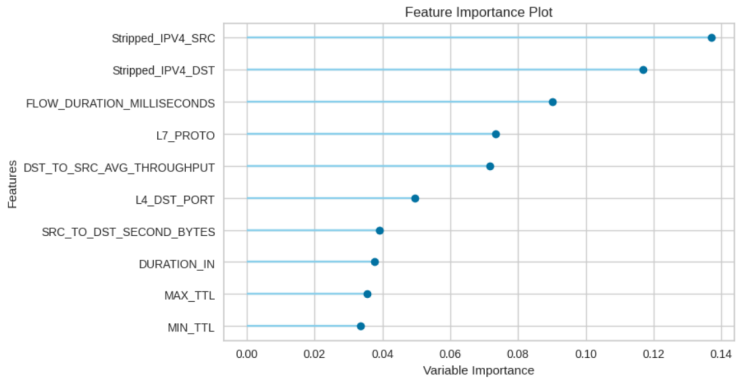

Figure 2. Feature importance result.

To create a global measure of feature importance, the mean absolute value was utilized as shown in Figure 4. Using the mean absolute shapely value of each feature, Stripped_IPV4_DST was the most important feature that changes the DDoS attack probability prediction by an average of 0.27 on the x-axis, as seen in Figure 4. Also, in Figure 5, the y-axis shows the features in their order of importance, while the x-axis reveals the SHAP values. The x-axis has both positive and negative sides. The graph has red and blue color representations of the values. Red indicates high values, while blue indicates low values. The figure illustrates that MN_TTL is high and has negative SHAP values. Thus, higher MN_TTL counts tend to have a negative impact on output. In all cases mentioned above, the destination address was shown to have the greatest magnitude effect.

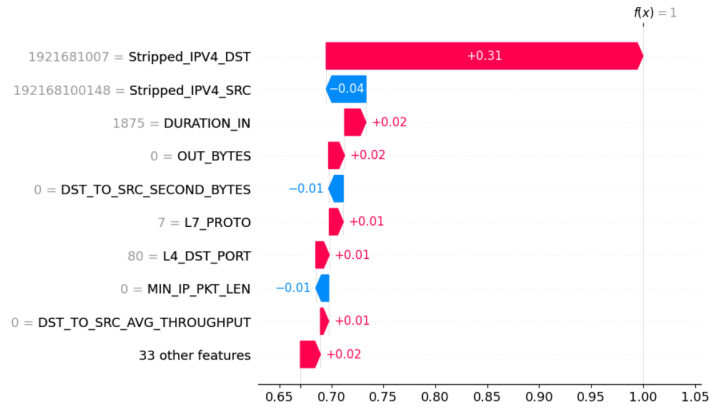

Figure 3. Feature importance results using SHAP.

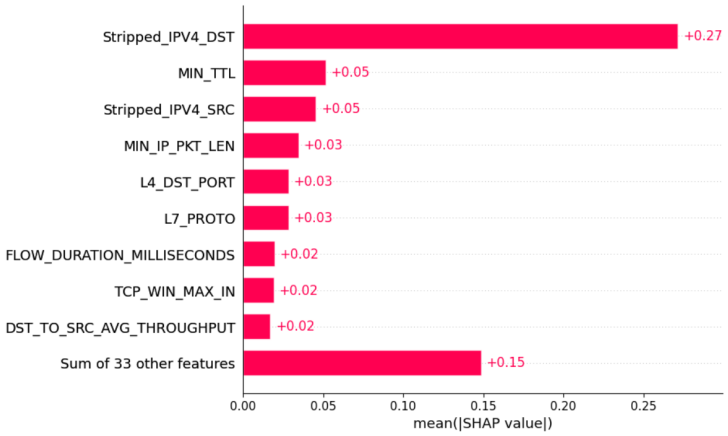

Figure 4. Mean absolute value to determine feature importance.

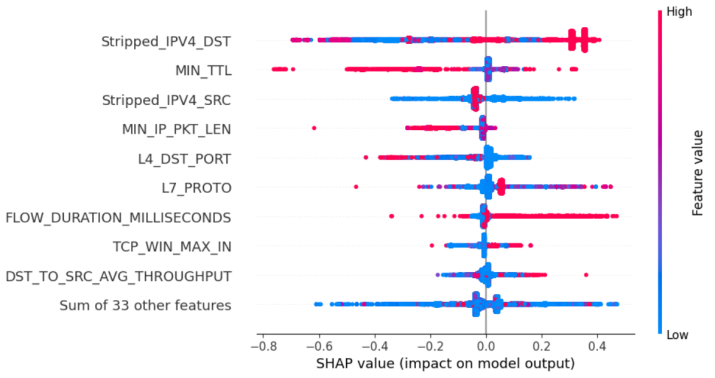

Figure 5. Feature impact on model's output.

Summary

In this section, two Python libraries (pycaret and shap) have been used to carry out a preliminary analysis of the NF-UQ-NIDS-V2 dataset. This preliminary work indicates that shallow machine learning models can effectively classify the attack traffic from benign traffic as seen by the accuracy result achieved by random forest in the analysis conducted. However, more complex models, such as deep neural networks, may be needed to further increase the accuracy of the classification. Additionally, further research is needed to explore how to use deep learning techniques to detect more sophisticated attack traffic. In addition, a future importance analysis is carried out on the dataset. Future importance analysis is crucial to discovering how the features impact the model's result. In this way, features that are not beneficial to the model's performance can be identified and removed, resulting in a more accurate and better-performing model. Based on the results, fundamental network traffic information, such as source IPs and destination IPs, had a significant impact on the model's output. This confirms why most researchers rely on them for malware detection. Meanwhile, the analysis indicates some features have no impact on the model's results. Consequently, some features may not be useful. Thus, it is important to carefully consider which features to include in a model to ensure its effectiveness.

The task of reliable labeling in the domain of cyber security, particularly in DDoS sample identification, emerges as a critical and formidable task that is made challenging due to various factors. A potential misclassification can be attributed to the evolution of attack techniques, intentional mislabeling by threat actors, and the complexity and multifaceted nature of these attacks. Furthermore, the absence of universal standards and the possibility of human error further complicate the process of accurately labeling and classifying DDoS samples. The cooperation of security experts in establishing standardized labeling criteria, the continuous updating of systems to address new threats, and the implementation of rigorous validation processes are vital in improving the reliability of these labels. Despite these measures, the achievement of absolute accuracy continues to be a persistent challenge in the constantly evolving field of cyber-security threats.

4. Proposed Hybrid Method for DDoS Attack Detection in MEC Networks

This section presents the proposed hybrid AE–MLP approach to DDoS attack detection in MEC. The hybrid AE–MLP combines an autoencoder with a multilayer perceptron for improved performance. It uses the compressed representation of the autoencoder as an input to the MLP for further processing. The hybrid architecture is advantageous as it utilizes the benefits of both components for better performance.

4.1. Autoencoder

Autoencoders are artificial neural networks used primarily in unsupervised learning tasks in machine learning [26]. Generally, it works by reconstructing input data at the output through the compression of the information into a latent or bottleneck layer. In essence, it encodes the input data into a lower-dimensional representation and then reconstructs the original input from this compressed representation. This type of neural network architecture is referred to as an auto-associative neural network. In an auto-associative neural network, input patterns are encoded and decoded within the same network architecture, making it ideal for tasks such as data reconstruction and pattern completion.

Dimensionality reduction is one of the principal uses of autoencoders. Autoencoders can capture the most significant characteristics and patterns in the data while eliminating less significant information [27]. Dimensionality reduction makes it possible to display the data in a more condensed manner, which lowers the amount of storage space needed and the computing complexity.

There are two distinct components of AE, an encoder and a decoder. The encoder takes the input data x, applies a linear transformation using weights W, adds a bias term b, and then applies an activation function $g()$ to produce the compressed representation Z.

This compressed representation Z can later be used in the decoder part of the autoencoder to reconstruct the original input data.

$$Z = g(Wx + b) \tag{1}$$

where the bottleneck layer representation = Z , the weight of the matrix = W, bias = b, and the activation function = g.

The decoder is

$$X' = g'(W'z + b') \tag{2}$$

where the decoder output = X', weight of the matrix = W', bias = b', and the activation function = g'.

Z indicates the input, while X' indicates the output. The decoder utilizes the output of the bottleneck layer to reconstruct the original input data.

4.2. Multilayer Perceptron

Multilayer perceptrons (MLPs) are a class of feedforward artificial neural networks with multiple layers, including input, hidden, and output layers [28]. The MLP is called a feedforward network because it transfers data from input to output in a single direction without requiring loops or feedback connections between neurons. This enables efficient learning and prediction compared to recurrent neural networks that rely on feedback connections [29]. As a result of its ability to model intricate relationships in data via interconnected nodes and layers, it is widely used for various tasks, including classification, regression, and pattern recognition.

The multilayer perceptron forms the foundation for more sophisticated architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). An MLP consists of multiple layers of interconnected neurons (also called nodes or units). In a typical network, there are three types of layers: the input layer, the hidden layer, and the output layer.

MLPs learn optimal weights and biases through a process called backpropagation during training. This process involves iteratively adjusting the network's parameters to minimize the difference between the predicted output and the actual target values. The choice of an appropriate optimization algorithm and loss function is critical for successful training.

MLPs are capable of learning complex and nonlinear relationships in data, enabling them to perform tasks, such as image recognition, natural language processing, and speech recognition. However, they are also prone to overfitting, especially on small datasets. In order to achieve good generalization performance, regularization techniques and careful tuning of hyperparameters are often required.

$$y = l(Wx + b) \tag{3}$$

where l is the activation function, W is weight, x is the input, b is bias, y is the output.

4.3. Proposed Hybrid AE–MLP Model Architecture

Our proposed model consists of two main components. The first component utilizes AE for feature dimensionality reduction. Figure 6 shows the architecture of AE and its working principle. The second component utilizes MLP for DDoS detection. AE learns from the input data and produces an output that represents the reduced-dimensional encoding of the input data. The MLP takes the encoded representations produced by the autoencoder as input. The MLP then learns to classify whether the input represents normal or potentially malicious network behavior. The hidden layers of the MLP enable it to capture more complex relationships in the data and make more accurate predictions.

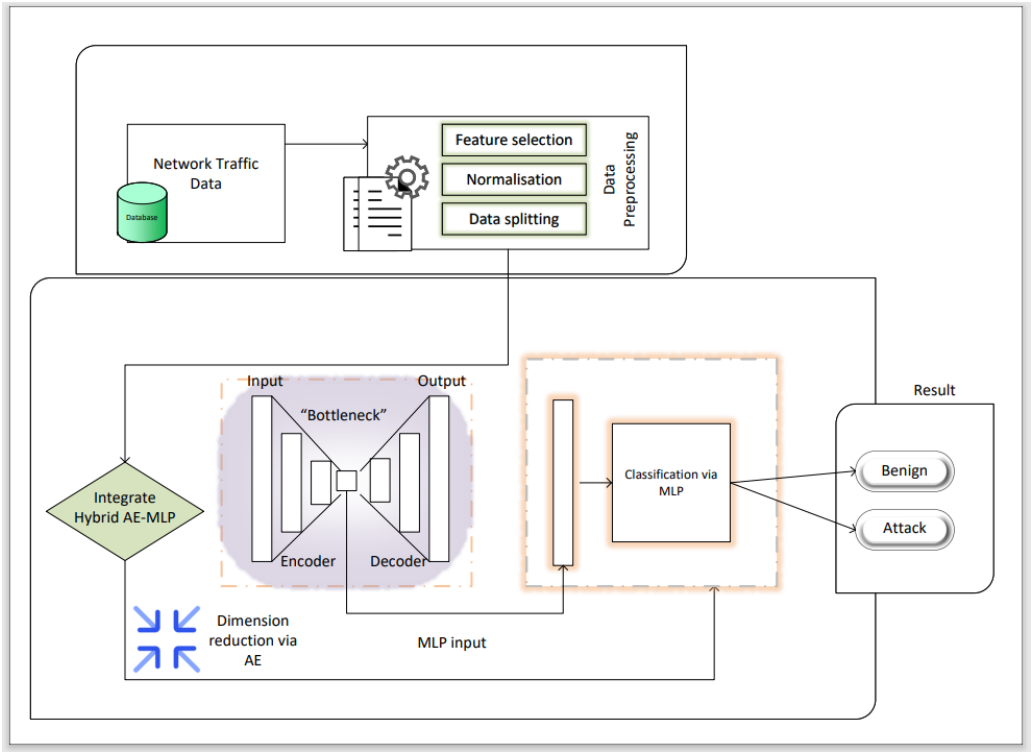

Figure 6. A representation of the hybrid AE–MLP architecture.

5. Results and Discussion

In this section, we provide the details of the experiment, the setup of the environment, the results, the analysis, and the conclusions.

The main system's processor is Intel(R) Core(TM) i7-4510U CPU @ 2.00 GHz 2.60 GHz, RAM-8.00 GB, and OS-Win 10, with the Google Colab Setting-Python 3 Google Compute Engine backend, RAM: 1.13 GB, and Disk: 26.26 GB.

To evaluate the strength of our proposed model, we utilized performance metrics, such as accuracy, precision, recall, and F1-score. These metrics measure the performance of the model in terms of its ability to correctly classify data points.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively.

5.1. Hybrid AE–MLP Was Employed for the Detection of DDoS Attack

The simulations for the NF-UQ-NIDS-V2 dataset have been conducted with 80/20, 70/30, and 60/40 data splits. In the first batch of experiments, the effectiveness of the model was evaluated in a binary class scenario. Experiments were conducted by using an 80/20 split. Performance scores of the proposed hybrid AE–ML are presented in bar graphs in Figure 7. The proposed method attained the highest accuracy of 99.98%. In the second stage, all experiments were repeated for a 70/30 split. Performance scores of

the proposed AE–MLP are presented in bar graphs in Figure 7. The proposed method attained the highest accuracy of 99.97%. In the third stage, all experiments were repeated for a 60/40 split. Performance scores of the proposed hybrid AE–MLP are presented in bar graphs in Figure 7. The proposed method attained the highest accuracy of 99.96%. The 80/20 split achieved the highest accuracy score for the model.

Table 5 shows the different train and test splits that were utilized, as well as the metrics.

Table 5. Train/Test split comparison-Hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model results.

Train/Test Split	Accuracy	Precision	Recall	F1-Score
80/20	99.98	98.92	94.24	96.52
70/30	99.97	98.10	95.20	97.20
60/40	99.96	98	95.10	97

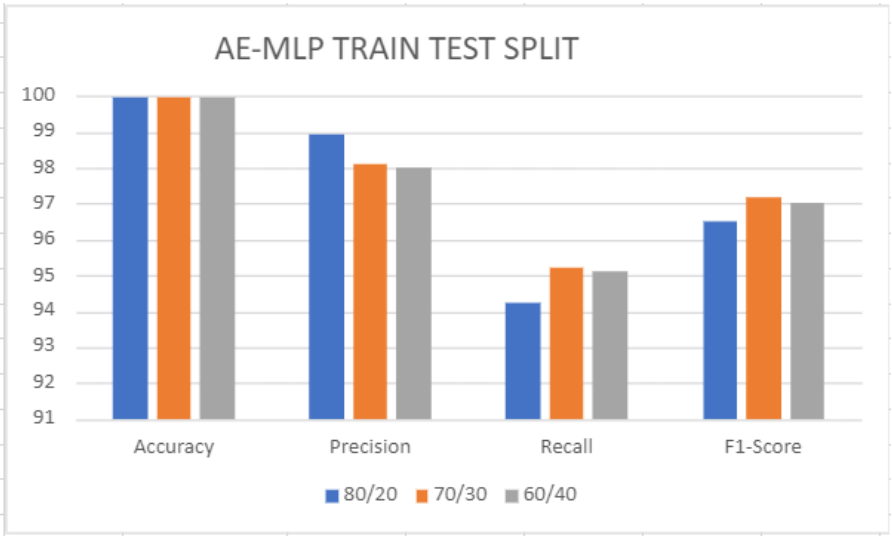

Figure 7. Result of the hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model train and test split.

The time taken for training and prediction was also captured with 80/20, 70/30, 60/40 splits for the NF-UQ-NIDS-V2 dataset, as shown in Figure 8. Experiments conducted using an 80/20 split took 4.96 s for training and 0.74 s for prediction. The shortest time recorded for training was with a 60/40 split, while the shortest prediction time recorded was with an 80/20 split. In comparison, the 80/20 split recorded the shortest prediction time, making it the most preferable split for our scenario.

Table 6 shows the time taken for training and prediction for the different train and test splits that were utilized.

Table 6. Time for hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model.

Train/Test Split	Training Time (s)	Prediction Time (s)
80/20	4.96	0.74
70/30	4.8	1.32
60/40	3.71	1.44

In this analysis, the hybrid AE–MLP method was found to be the most effective method for detecting DDoS attacks. The proposed DDoS detection model outperformed all other algorithms with an accuracy of 99.98%. Table 7 shows the performance comparison between the proposed hybrid algorithm and two others.

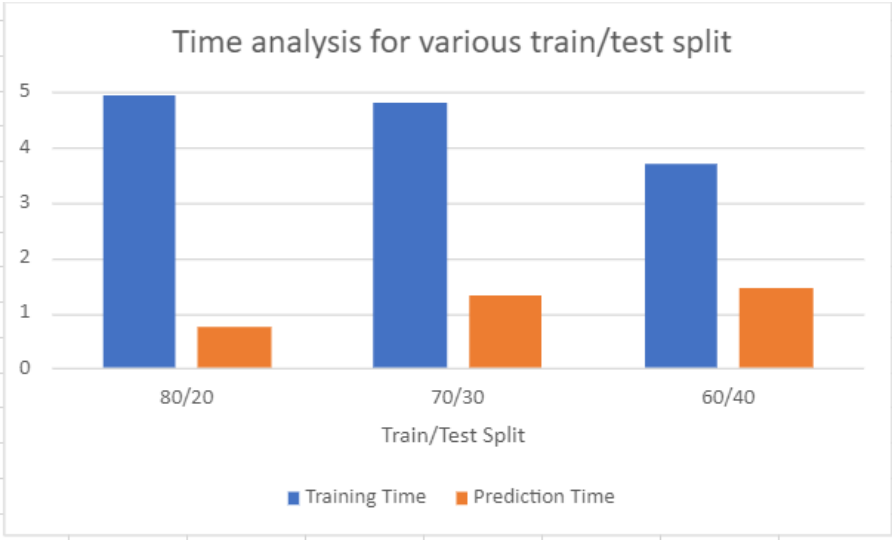

Figure 8. Time taken for hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model.

Table 7. Literature comparison-Hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model results.

Algorithm	Accuracy	Precision	Recall	F1-Score
LSTM	94.98	98.30	94.14	96
GRU	97.10	97.20	98.52	98
AE-MLP	99.98	98.92	94.24	96.52

Figure 9 compares the accuracy, precision, recall, and f1-score for LSTM, GRU, and hybrid AE–MLP.

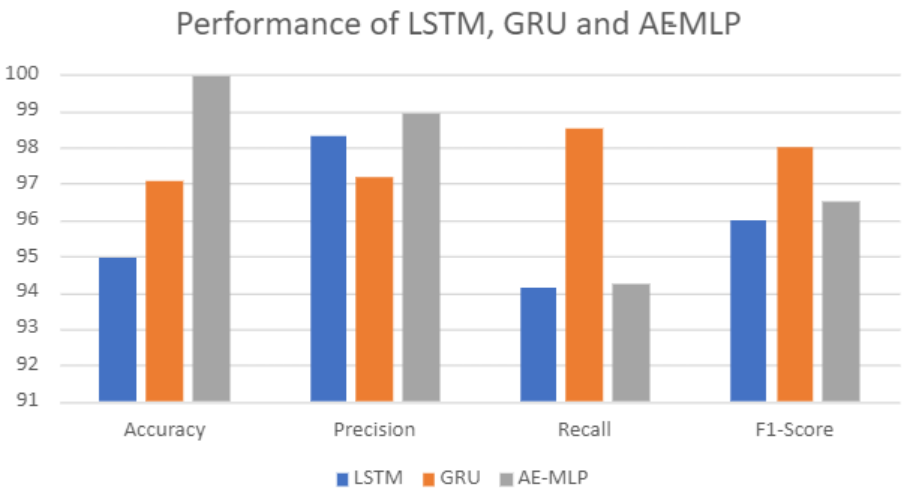

Figure 9. Comparison between Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and the hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model.

Table 8 shows the performance comparison between the proposed hybrid AE–MLP algorithm and MLP.

Figure 10 shows the performance comparison between the proposed hybrid AE–MLP and MLP.

Table 9 and Figure 11 show the accuracy comparison among different feature selections of the proposed hybrid AE–MLP algorithm and MLP.

Table 8. Proposed work comparison-Hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model results.

Algorithm	Accuracy	Precision	Recall	F1-Score
MLP	98.63	99.40	98.55	99
AE-MLP	99.98	98.92	94.24	96.52

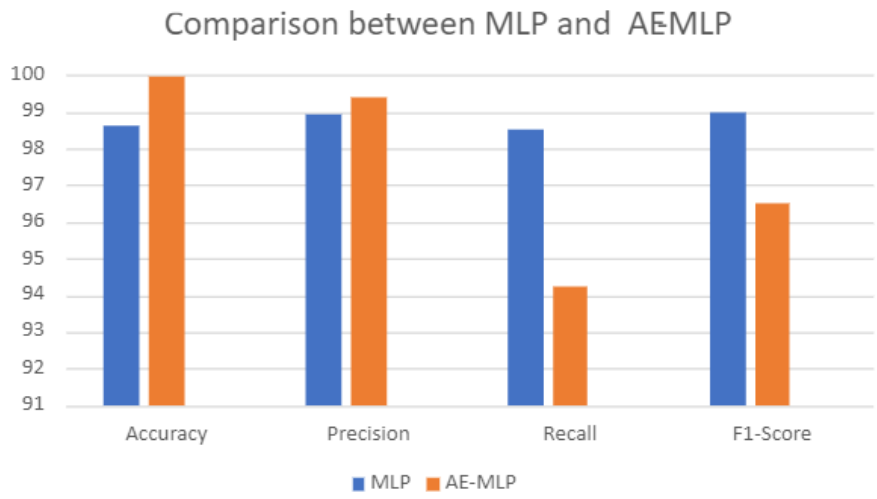

Figure 10. Hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model comparison with Multi-Layer Perceptron (MLP).

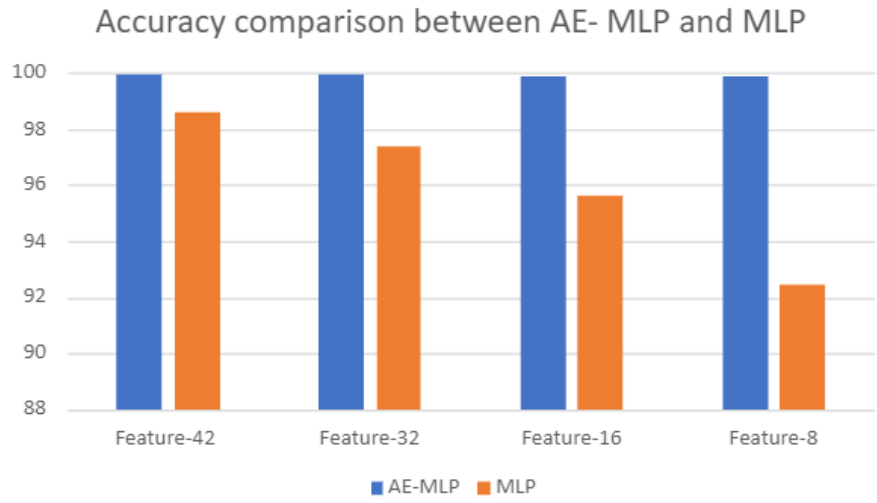

Figure 11. Feature hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model comparison with Multi-Layer Perceptron (MLP).

Table 9. Number of features comparison-Hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model results.

Nos of Features	Accuracy	MLP	AE-MLP
Feature-42	Accuracy	98.63	99.98
Feature-32	Accuracy	97.41	99.93
Feature-16	Accuracy	95.62	99.91
Feature-8	Accuracy	92.45	99.91

Table 10. The result of individual attack class with hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model.

Class	Accuracy	Precision	Recall	F1-Score	FNR	TNR	FPR	TPR	AUC-ROC
DoS	94.54	94.29	94.50	94.24	0.04	0.94	0.04	0.94	0.90
DDoS	94.56	94.37	94.44	94.36	0.04	0.94	0.04	0.94	0.90
Scanning	93.92	93.90	93.92	93.88	0.05	0.93	0.05	0.93	0.89
Xss	93.90	93.97	93.89	93.80	0.05	0.93	0.05	0.93	0.89
Reconnaissance	93.90	93.10	93.90	93.79	0.05	0.93	0.05	0.93	0.89

Table 10 shows the results of some of the individual attacks as enumerated in Table 2 with the proposed hybrid AE–MLP model. The purpose of this is to provide a more comprehensive evaluation of the proposed model. As can be seen, other metrics, such as False Negative Rate (FNR), True Negative Rate (TNR), False Positive Rate (FPR), True Positive Rate (TPR), and Area Under the Receiver Operating Characteristic Curve (AUC-ROC) have been added to provide a more detailed analysis of the model's performance.

Table 11 shows the performance of the proposed hybrid AE–ML model against shallow ML models using the NF-UQ-NIDS-V2 dataset. The shallow RF model had an accuracy of 99.59%, demonstrating its effectiveness in this instance. On the other hand, SVM performed significantly lower, with an accuracy of 68.70%. The discrepancy may be explained by SVM's sensitivity to the selected kernel and its difficulty handling large-scale, non-linear datasets, both of which are relevant in this case [30].

Table 11. Comparison of proposed hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model with Multi-Layer (ML) models.

Algorithm	Accuracy	Precision	Recall	F1-Score	Training Time (s)
XGB	99.54	99.68	99.62	99.65	1.67
Random Forest	99.59	99.75	99.63	99.69	7.42
SVM	68.70	69.73	94.18	80.13	8.91
KNN	99.17	99.45	99.31	99.38	7.97
AE-MLP	99.98	98.92	94.24	96.52	4.96

When considering the training time, XGB stands out as the fastest learner, with an impressive time of 1.67 s. The accelerated training speed can be attributed to XGB's built-in optimization techniques, such as parallel computing and its decision tree-based approach, which facilitate efficient computation and model development. Among the models, SVM demonstrates the longest training time of 8.91 s, primarily due to its iterative nature and the computational intensity associated with computing the support vectors [30].

Table 12 shows a comparison between the proposed hybrid AE–MLP model and three related works, demonstrating its superiority with a 99.98% attack detection accuracy. Furthermore, the proposed model maintains a competitive edge across precision, recall, and F1-score metrics, standing at 98.92%, 94.24%, and 96.52%, respectively. This demonstrates its comprehensive approach to discerning threats.

Table 12. Comparison of proposed hybrid Autoencoder–Multi-Layer Perceptron (AE–MLP) model with existing work.

Study	Technique	Accuracy	Precision	Recall	F1-Score
Alghazzawi et al. [31]	CNN + BiLSTM	94.52	94.74	92.04	93.44
Tanveer [32]	CNN + LSTM	96	95.94	97.06	96.50
Mhamdi et al. [33]	SAE + 1SVM	99.35	99.97	98.28	99.11
Our proposal	AE + MLP	99.98	98.92	94.24	96.52

5.2. Key Findings

Various experiments were conducted, and comparisons were made to show the effectiveness of the proposed hybrid AE–MLP model:

- In the first category of the experiment, an 80/20 split of the training and test dataset yielded the highest accuracy score of 99.98% for the model.
- The training time was found to be 4.96 s, and the prediction time was 0.74 s.
- The proposed hybrid AE–MLP model was compared with other deep learning models in the second category of the experiment. The hybrid AE–MLP achieved a higher accuracy of 99.98% compared with 94.98% and 97.10% for LSTM and GRU, respectively.
- The proposed hybrid AE–MLP model was also compared with MLP. The hybrid model achieved a higher accuracy of 99.98% compared with MLP, which was 98.63%.
- The hybrid AE–MLP model was compared with the shallow ML model, and the training time was recorded. The proposed hybrid model achieved a remarkable accuracy of 99.98% in a record time of 4.96 s.

5.3. Advantages of AE–MLP Hybrid Approach

An AE–MLP (Autoencoder–Multi-Layer Perceptron) hybrid approach offers significant advantages for intrusion detection systems (IDSs) [34]. Firstly, autoencoders excel at anomaly detection, which is pivotal for identifying unknown and novel threats. They can automatically learn patterns and anomalies in network traffic or system behavior, reducing the reliance on manual feature engineering. Furthermore, the feature extraction capabilities of autoencoders enable them to capture both common and rare patterns, enhancing the IDS's adaptability and robustness to evolving attack techniques. By employing an MLP for classification after the autoencoder, the hybrid model can validate the anomalies detected and make informed decisions about the maliciousness of network events or system activities, reducing false positives and improving the overall accuracy of intrusion detection [35].

Secondly, the hybrid approach benefits from transfer learning and adaptability. A transfer learning method uses pre-trained models to solve a problem related to the original problem, allowing for efficient and effective learning on smaller datasets and in new domains [36]. Autoencoders can be pre-trained on large datasets, allowing the IDS to generalize better to new and emerging threats. This pre-training serves as a form of transfer learning, enabling the model to detect threats that were not present in the training data. Additionally, the hybrid model is more adaptable to changing attack scenarios and network environments. It can be updated and fine-tuned with new data to capture evolving attack patterns, making it more responsive and effective in addressing emerging security challenges. The combination of multiple models in the AE–MLP ensemble not only enhances detection capabilities but also provides a robust and resilient defense against a wide range of security threats.

5.4. Drawback of AE–MLP Hybrid Approach

The downside of using a hybrid model is the time taken for the process to complete. As can be seen in the experiment, the hybrid model recorded the highest training time of 4.96 s. This was significantly higher than the other models. This suggests that the hybrid model is more computationally intensive. This drawback can be mitigated by employing a cloud-edge collaboration technique. In this technique, the training is conducted in the cloud environment where there are vast resources to do this in a record time. The trained model is then deployed to the edge for prediction. Considering the proposed model's fastest prediction time, it is obvious that malicious activities will be detected quickly.

6. Conclusions

Mobile Edge Computing (MEC) has undeniably ushered in a new era of computing, offering unparalleled speed and responsiveness at the edge of the network. However, with the immense potential of MEC come security challenges, and none is more formidable

than the threat of Distributed Denial of Service (DDoS) attacks. One critical issue that has plagued MEC security is the reliance on shallow Machine Learning (ML) models for Intrusion Detection Systems (IDSs). While these systems have served as a valuable line of defense, their limitations in accurately detecting and mitigating DDoS attacks have become increasingly evident. The shallow ML models lack the depth and sophistication required to discern the evolving tactics employed by attackers in the dynamic MEC environment.

To resolve these shortcomings, this study presents a hybrid DL method for the detection of DDoS attacks in the MEC environment. The proposed hybrid DL model's architecture comprises two components: the autoencoder and the multi-layer perceptron network. The AE extracts the most important and relevant features necessary to find malicious DDoS network payloads from a large-scale network traffic sample. The compressed and reduced features produced by the AE model are then fed into MLP to effectively classify different DDoS attack types. The proposed model's effectiveness was validated by performing extensive experiments with the most relevant publicly available dataset (NF-UQ-NIDS-V2). Based on experimental results, the proposed AE–MLP outperforms single models in the vast majority of situations. Similarly, comparisons were drawn between the proposed model and various existing related studies. The accuracy achieved by the proposed model was higher than that of its counterparts. A hybrid model is more suitable for IDS because it can learn patterns in a large amount of traffic data and classify it quickly as benign or malicious. However, due to the time element involved in hybrid models, cloud–edge collaboration is suggested. In this suggested method, the hybrid model training could take place in the cloud, and the trained model is deployed to the MEC environment for effective DDoS attack detection.

Utilizing the hybrid AE–MLP model in MEC environments holds significant implications. The integration of AE's data compression and feature extraction capability with MLP's classification capabilities, results in effective processing and extraction of significant features from data at the edge. This integration enables quicker decision making and optimizes latency and bandwidth utilization. Thus, the hybrid AE–MLP model provides an ideal solution for MEC applications that require real-time decision making. Furthermore, the AE–MLP model can also be used for predictive analytics, allowing for more accurate prediction of future outcomes. Future work could examine methodologies for facilitating effective collaboration between edge and cloud resources. The development of frameworks will facilitate the hybrid model's seamless offloading of complex processing to the cloud, therefore, optimizing overall system performance.

Author Contributions: Conceptualization, O.A. and A.S.S.; methodology, O.A. and A.S.S.; software, O.A.; validation, O.A., A.S.S. and P.P.; formal analysis, O.A.; investigation, O.A.; resources, A.S.S., P.P. and M.A.; data curation, O.A.; writing—original draft preparation, O.A.; writing—review and editing, O.A., A.S.S., P.P., M.A. and O.K.; visualization, O.A.; supervision, A.S.S. and P.P.; project administration, P.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the ongoing research with it.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xiao, L.; Wan, X.; Dai, C.; Du, X.; Chen, X.; Guizani, M. Security in mobile edge caching with reinforcement learning. *IEEE Wirel. Commun.* **2018**, *25*, 116–122. [CrossRef]
2. Kaja, N.; Shaout, A.; Ma, D. An intelligent intrusion detection system. *Appl. Intell.* **2019**, *49*, 3235–3247. [CrossRef]
3. Zainudin, A.; Ahakonye, L.A.C.; Akter, R.; Kim, D.S.; Lee, J.M. An efficient hybrid-dnn for ddos detection and classification in software-defined iiot networks. *IEEE Internet Things J.* **2022**, *10*, 8491–8504. [CrossRef]
4. Ahmad, Z.; Shahid, Khan, A.; Wai, Shiang, C.; Abdullah, J.; Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4150. [CrossRef]

5. Oo, M.C.M.; Thein, T. An efficient predictive analytics system for high dimensional big data. *J. King Saud-Univ.-Comput. Inf. Sci.* **2022**, *34*, 1521–1532. [CrossRef]
6. Jia, W.; Sun, M.; Lian, J.; Hou, S. Feature dimensionality reduction: A review. *Complex Intell. Syst.* **2022**, *8*, 2663–2693. [CrossRef]
7. Naina Chaturvedi. Dimensionality Reduction Using an Autoencoder in Python. 2021. Available online: https://medium.datadriveninvestor.com/dimensionality-reduction-using-an-autoencoder-in-python-bf540bb3f085 (accessed on 5 August 2023)
8. Wani, A.R.; Rana, Q.P.; Saxena, U.; Pandey, N. Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 870–875.
9. Bindra, N.; Sood, M. Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. *Autom. Control Comput. Sci.* **2019**, *53*, 419–428. [CrossRef]
10. Khare, M.; Oak, R. Real-Time distributed denial-of-service (DDoS) attack detection using decision trees for server performance maintenance. In *Performance Management of Integrated Systems and Its Applications in Software Engineering*; Springer: Singapore, 2020; pp. 1–9.
11. Kousar, H.; Mulla, M.M.; Shettar, P.; Narayan, D.G. Detection of DDoS attacks in software defined network using decision tree. In Proceedings of the 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 18–19 June 2021; pp. 783–788.
12. Arshi, M.; Nasreen, M.D.; Madhavi, K. A survey of DDoS attacks using machine learning techniques. *E3S Web Conf.* **2020**, *184*, 01052. [CrossRef]
13. Suthishni, D.N.P.; Kumar, K.S. A review on machine learning based security approaches in intrusion detection system. In Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 23–25 March 2022; pp. 341–348.
14. Kasun, L.L.C.; Yang, Y.; Huang, G.B.; Zhang, Z. Dimension reduction with extreme learning machine. *IEEE Trans. Image Process.* **2016**, *25*, 3906–3918. [CrossRef]
15. Elsayed, M.S.; Le-Khac, N.A.; Dev, S.; Jurcut, A.D. Ddosnet: A deep-learning model for detecting network attacks. In Proceedings of the 2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 31 August–3 September 2020; pp. 391–396.
16. Yuan, X.; Li, C.; Li, X. DeepDefense: Identifying DDoS attack via deep learning. In Proceedings of the 2017 IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, China, 29–31 May 2017; pp. 1–8.
17. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A. A two-stage intrusion detection system with auto-encoder and LSTMs. *Appl. Soft Comput.* **2022**, *121*, 108768. [CrossRef]
18. Lee, J.; Park, K. AE-CGAN model based high performance network intrusion detection system. *Appl. Sci.* **2019**, *9*, 4221. [CrossRef]
19. Hara, K.; Shiomoto, K. Intrusion detection system using semi-supervised learning with adversarial auto-encoder. In Proceedings of the NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–8.
20. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *J. Inf. Secur. Appl.* **2021**, *58*, 102804. [CrossRef]
21. Li, X.; Chen, W.; Zhang, Q.; Wu, L. Building auto-encoder intrusion detection system based on random forest feature selection. *Comput. Secur.* **2020**, *95*, 101851. [CrossRef]
22. Sarhan, M.; Layeghy, S.; Portmann, M. Towards a standard feature set for network intrusion detection system datasets. *Mob. Netw. Appl.* **2022**, *27*, 357–370. [CrossRef]
23. Pycaret. 2020. Available online: https://pycaret.readthedocs.io/en/latest/ (accessed on 5 August 2023).
24. Lansky, J.; Ali, S.; Mohammadi, M.; Majeed, M.K.; Karim, S.H.T.; Rashidi, S.; Hosseinzadeh, M.; Rahmani, A.M. Deep learning-based intrusion detection systems: A systematic review. *IEEE Access* **2021**, *9*, 101574–101599. [CrossRef]
25. SHAP. An Introduction to Explainable AI with Shapley Values. 2018. Available online: https://shap.readthedocs.io/en/latest/example_notebooks/overviews/An%20introduction%20to%20explainable%20AI%20with%20Shapley%20values.html (accessed on 5 August 2023).
26. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [CrossRef]
27. Ieracitano, C.; Adeel, A.; Morabito, F.C.; Hussain, A. A novel statistical analysis and autoencoder driven intelligent intrusion detection approach. *Neurocomputing* **2020**, *387*, 51–62. [CrossRef]
28. Singh, J.; Banerjee, R. A study on single and multi-layer perceptron neural network. In Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 27–29 March 2019; pp. 35–40.
29. Imran, M.; Haider, N.; Shoaib, M.; Razzak, I. An intelligent and efficient network intrusion detection system using deep learning. *Comput. Electr. Eng.* **2022**, *99*, 107764.
30. Guo, Y.; Zhang, Z.; Tang, F. Feature selection with kernelized multi-class support vector machine. *Pattern Recognit.* **2021**, *117*, 107988. [CrossRef]
31. Alghazzawi, D.; Bamasag, O.; Ullah, H.; Asghar, M.Z. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Appl. Sci.* **2021**, *11*, 11634. [CrossRef]
32. Tanveer, M. Internet of Things attack detection using hybrid Deep Learning Model. *Comput. Commun.* **2021**, *176*, 146–154.

33. Mhamdi, L.; McLernon, D.; El-Moussa, F.; Zaidi, S.A.R.; Ghogho, M.; Tang, T. A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs. In Proceedings of the 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), Hammamet, Tunisia, 27–30 October 2020; pp. 1–6.
34. Keser, R.K.; Töreyin, B.U. Autoencoder based dimensionality reduction of feature vectors for object recognition. In Proceedings of the 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Sorrento, Italy, 26–29 November 2019; pp. 577–584.
35. Gharib, M.; Mohammadi, B.; Dastgerdi, S.H.; Sabokrou, M. Autoids: Auto-encoder based method for intrusion detection system. *arXiv* **2019**, arXiv:1911.03306.
36. Zhu, Z.; Lin, K.; Jain, A.K.; Zhou, J. Transfer learning in deep reinforcement learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 13344–13362. [CrossRef] [PubMed]