DNS Data Exfiltration Detection Using CNN-LSTM with Attention Mechanism(CLAM)

MSc Research Project
Cloud Computing

Jisha Joy
Student ID: 21240868

School of Computing
National College of Ireland

Supervisor: Dr. Shivani Jaswal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jisha Joy
Student ID:	21240868
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Shivani Jaswal
Submission Due Date:	14/12/2023
Project Title:	DNS Data Exfiltration Detection Using CNN-LSTM with Attention Mechanism(CLAM)
Word Count:	6029
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jisha Joy
Date:	13th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DNS Data Exfiltration Detection Using CNN-LSTM with Attention Mechanism(CLAM)

Jisha Joy
21240868

Abstract

Data is a critical feature of the data-driven technological world. During the Covid pandemic, most of the organizations shifted to the cloud network for data transfer and storage. As more organizations and individuals shift to the cloud platforms, exfiltration of the data in the cloud network has become a serious threat. DNS-based data exfiltration is a commonly used technique by attackers for accessing confidential data in cloud platforms using DNS query packets. Different methodologies especially machine learning models were proposed for the detection of exfiltration attacks in on-premises networks. In a cloud environment, security, availability, scalability, and most importantly reliability of the detection technique are the important performance metric. In this research, a cloud machine learning model which is a hybrid of CNN and LSTM with an additional mechanism of attention applied to them is proposed. By applying the attention technique to the outputs of the CNN and LSTM, the features critical in detecting exfiltration are highlighted thereby increasing the accuracy of the model and reducing the number of false positive predictions. This model provided higher accuracy, security, and reliability in DNS exfiltration detection in cloud platforms compared to the existing models.

Keywords- Exfiltration, DNS Query, Convolutional Neural Network, Attention, Long Short Term Memory, Reliability

1 Introduction

With the increasing volume of data, several serious security threats have been associated with data. Data exfiltration is one such kind of intrusion or unauthorized access that affects individuals' financial aspects and privacy. Exfiltration attacks on data are constantly increasing these days and the major reason for this is that some of the protocols used for the exchange of data plays as the gateway for such intrusion attacks. Moreover, attackers are now shifting their focus to organizations rather than individuals, and the reputed companies are now the targets of such exfiltration attacks. There are legal implications associated with such threats, especially in government organizations. The traditional firewall-based security methods have proved to be of no use in this scenario and thus other effective techniques have been introduced. It is always essential that exfiltration attacks on data should be detected and resolved way beforehand, which is considered a crucial and difficult task Ullah et al. (2018).

Domain Name System(DNS) based exfiltration is the most prevalent of these attacks.

A Domain Name System enables the detection of IP addresses using the domain name. This feature has been misused by attackers in the form of DNS tunneling. In this kind of attack, sensitive information from a user machine is fetched using a DNS packet. Network firewalls were the earlier solution to this problem but the detection of such data leakages using traditional techniques for screening the DNS packets can affect the performance parameters of the network like speed and efficiency. Traffic streaming of DNS was one of the exfiltration detection and prevention techniques widely used previously but it proved to be less efficient. Thus the discovery of machine learning has resulted in more efficient and accurate techniques that could identify and prevent data exfiltration on the network. Such techniques make use of the network features including traffic pattern, and length of the packets. Statistics state that machine learning techniques have been effective in detecting DNS exfiltration and intrusion attacks. Abualghanam et al. (2023).

Several machine learning models like Logistic Regression(LR), Decision Tree(DT) Classifier, and Random Forest(RF) Classifier have been used to detect DNS exfiltration using the exfiltrated data collected from the data packets in the network. In addition to this, deep learning models were also introduced that provided more accuracy in predictions and thereby effectively detecting DNS tunneling. Now with the rapid advent and shift towards cloud technologies, it has become essential to bring forth a model that could detect exfiltration attacks efficiently considering cloud metrics like reliability, scalability, and secure access. Additionally, data flowing through the cloud is obtained from a challenging variety of sources which can produce predictions that are false positive. This demonstrated the need for a reliable and securely encrypted deep learning model to detect and prevent DNS-based data exfiltration accurately and more efficiently.

1.1 Research Question

What is the impact on cloud security of using a hybrid Convolutional Neural Network(CNN) and Long Short-Term Memory(LSTM) with an attention technique for detecting DNS-based data exfiltration?

To answer the research question, a hybrid model of Convolutional Neural Network and Long Short Term Memory(CNN-LSTM) referred to as CLAM(CNN-LSTM with Attention Mechanism) is proposed in this study that can detect the exfiltration attempts in a cloud environment more effectively than the existing models. A Convolutional Neural Network is capable of identifying hidden DNS patterns from the DNS queries in a network data sequence like query length that effectively identifies it to be benign or exfiltration. Similarly, a Long Short Term Memory model is capable of reflecting the long-term dependencies in the data sequence. Thus, a combined model of CNN and LSTM improves the accuracy of predicting exfiltration and intrusion attacks and provides reliable, secure, and scalable results with large volumes of data in cloud platforms. This can further reduce the possibility of false positives that can provide inaccurate data to the users. Therefore the cascaded model can provide high-quality results in on-premise and cloud networks.

1.2 Motivation

With the increasing use of cloud technology, it is essential to ensure the safety of data residing in the cloud. Thus, the main motivations of this study are the following:

1. To develop an effective machine learning model that can be deployed onto the cloud environment to ensure cloud security
2. To use a mechanism in the model that can highlight the relevant features in the cloud network responsible for detecting exfiltration attacks
3. To improve the accuracy of predicting Domain Name System-based exfiltration attacks in cloud platforms

1.3 Ethics Consideration

Table 1 represents the ethical consideration of the research. This research uses only publically available secondary datasets.

Table 1: Ethics Consideration Table

This project involves human participants	Yes/[✓ No]
The project makes use of secondary dataset(s) created by the researcher	Yes/[✓ No]
The project makes use of public secondary dataset(s)	[✓ Yes]/No
The project makes use of non-public secondary dataset(s)	Yes/[✓ No]
Approval letter from non-public secondary dataset(s) owner received	Yes/[✓ No]

1.4 Report Structure

This report consists of the following sections. Section 2 elaborates on the different related works associated with DNS exfiltration detection including different techniques proposed over time and a critical analysis of the works. Section 3 describes the research methodology of the proposed solution and the corresponding workflow. Section 4 represents the design specification of the proposed methodology with the architecture whereas section 5 describes the implementation of the CNN-LSTM algorithm with the attention mechanism. Section 6 explains the evaluation and results of the proposed implementation and section 7 provides the conclusion and future work of the method.

2 Related Work

During the past years, several kinds of research were conducted for the detection and prevention of exfiltrations or intrusions in the network. The traditional approaches were overridden with the advent of machine learning models. Several supervised, unsupervised, and deep learning models were proposed to prevent exfiltration attacks on the network, and recently studies have been conducted for the detection and prevention of DNS exfiltration attacks on cloud networks.

2.1 DNS exfiltration detection using classification techniques

The study conducted by Alkasassbeh and Almseidin (2023) proposed machine learning classifiers to check the hidden tunnel of DNS traffic and detect the DNS tunneling attacks. They used classifiers like J48, Random Forest, and Multilayer Perceptron. These algorithms analyzed the relationship between the features and thereby detected exfiltration attacks in the network traffic. A cross-validation technique was used to test the data and the results were evaluated using machine learning performance evaluation metrics like accuracy, precision, and recall. However, these models tend to produce false positives in their predictions which is the greatest drawback of this technique. Moreover, they cannot be considered a reliable technique for ensuring network security.

Another research conducted by Nadler et al. (2019) proposes a classification model named Isolation Forest for detecting DNS tunneling attacks. This technique uses a combination of binary trees for detecting network anomalies. In addition to detecting DNS exfiltration attacks, this technique also detects low-throughput malware exfiltration from the logs of DNS network traffic thereby ensuring advanced security. This technique works by extracting the features from the traffic logs of a particular domain and performing classification using the Isolation Forest classifier. Thus, this has helped identify and block the DNS requests that are indicative of attacks and also provides fewer false positive results. They also found that this method can identify slow attacks with the help of the sliding window approach. The main drawback of this approach was that it worked well with a specific domain which makes it easier for the attackers to evade this technique. Hence, a technique that applies to different domains needs to be formulated. In the paper proposed by Buczak et al. (2016), a Random Forest Classifier is used in detecting DNS exfiltration. The earlier techniques of exfiltration detection on networks like firewalls proved to be less effective and thus a classification technique like the Random Forest algorithm was considered to be the Advanced detection system specifically known as APT(Advanced Persistent Threat). They considered this an effective exfiltration attack detection mechanism because of the capability of this technique to operate in complicated environments such as Defense-in-Depth(DiD) which is a security framework. A penetration testing using tunneling software for Domain Name System(DNS) was performed on AWS cloud servers to acquire the data that was required to train the Random Forest model and thereafter a feature extraction was done on the data thus obtained. The combined use of the Random Forest algorithm with bagging helps in reducing the variance of the learning outcome. The major disadvantage of this mechanism is that it results in a high number of false positive predictions which can severely affect the reliability of exfiltration attack predictions.

An ensemble model of machine learning for exfiltration detection was proposed in the paper Shafieian et al. (2017). The key advantage of this model is that it results in highly accurate predictions with minimum false positive predictions which is the main requirement of an efficient DNS exfiltration prediction model. The authors proposed an ensemble of three algorithms in this approach namely, Random Forest classifier, Multilayer Perceptron network, and k-Nearest Neighbor(KNN) where more attention was given to the Random Forest classifier which in turn was an ensemble method. Additionally, incremental learning can be performed on the k-NN algorithm of the ensemble model and this has proved to increase the performance of the model. Weka Java APIs that use a specific format named Attribute Relation File Format(ARFF)are used for the DNS packet evaluation since the ARFF header contains the data type of the features or attributes which is an essential requirement in the development of the learning model. One of the major drawbacks of this technique is that the learning model needs to be periodically updated and saved for incorporating the network traffic variations. All the above techniques were dependent on plaintext DNS transmission and it had severe privacy-related concerns and thus requires a more efficient technique for DNS exfiltration detection.

2.2 Detecting DNS exfiltration over HTTPS

In the paper by Steadman and Scott-Hayward (2022), DNS exfiltration detection over an encrypted Domain Name System(DNS) was proposed. They proposed a technique DoHxP that ensures DNS protection over HTTPS thereby avoiding DNS-based data exfiltration and also ensuring user data privacy. This technique was capable of distinguishing DoH traffic features from HTTPS traffic features. The DoHxP made use of Software-Defined Networking(SDN) that was programmable for detecting malicious network traffic. The five main features of the network traffic such as query length, name entropy, record type, frequency per domain, and volume per domain were used to identify the DNS over HTTPS(DoH) exfiltration. In the DoHxP system, an eBPF module labels the DoH traffic and determines if it is a blocklisted one, and thereafter a P4 flow processor determines if it is HTTPS traffic that is labeled as DoH. Finally, the ONOS application distinguishes between malicious and benign traffic. The main limitation of this approach is that in real networks the performance of the technique can be compromised and further evaluations need to be conducted.

A solution for effectively detecting DNS exfiltration over HTTPS was brought forward by Zhan et al. (2022) since the other techniques used in DNS plain text transmission detections are considered to be not secure. In the DNS over HTTPS technique, the DNS queries are transmitted over an HTTPS request that is fully encrypted and prevents unauthorized external access. DoH tunneling is detected using a technique of TLS fingerprint (Transport Layer Security fingerprint). The plain text information that is generated during the TLS handshake process is used for Transport Layer Security fingerprinting. During the process of TLS fingerprinting, the fingerprints of each client are examined to check if they are unique. This is achieved by performing a feature extraction on the DoH packet data and thereafter training the extracted features using different machine learning models like Decision Tree combined with Boosting, Linear Regression classification, and Random Forest classification technique for DNS exfiltration detection. Even though this method assures a safe and secure transmission of DNS packets, this can be compromised by some adversarial attacks by modifying certain features of the network traffic data. On the other hand, exfiltration over DoH detection with the help of a predetermined dataset was proposed by Singh and Roy (2020). Figure 1 indicates the general workflow of the DNS over the HTTPS process. The dataset that was used for this purpose was CIRA-CIC-DoHBrw-2020 which was a benchmarked MoH dataset. DoHMeter tools were used for extracting the features from the dataset consisting of normal data and malign data. The authors used different training models like the Random Forest algorithm, Naive Bayes classifier, K-Nearest Neighbour(KNN), Logistic Regression algorithm, and Gradient Boosting. Of this, the Gradient Boosting and Random Forest algorithms displayed a higher accuracy in predicting the DNS over HTTPS exfiltration attacks. As the predictions were dependent on a predefined dataset, this technique failed to provide accurate results with real-time data which is the major drawback of this technique. Since the above techniques for data exfiltration detection over DoH can be bypassed by attackers and are dependent on specific datasets, a more accurate and effective technique is required for DNS exfiltration detection.

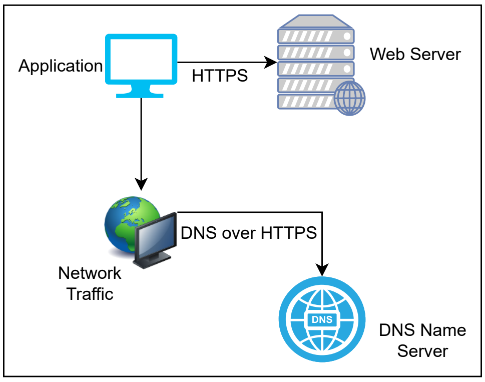


Figure 1: DoH(DNS over HTTPS) process flow

2.3 Detecting DNS exfiltration using Deep Learning techniques

The use of deep learning techniques like Convolutional Neural Networks in detecting DNS exfiltration was proposed by Liang et al. (2023). As per the authors, it is difficult for the attackers to invade the data on which feature extraction using a Convolutional Neural Network(CNN) is done. This specific technique for DNS tunneling detection was referred to as FECC(Feature Extraction CNN and Clustering). Extraction of the features in the data is performed using a Convolutional Neural Network(CNN) which is then evaluated with a clustering mechanism. The DNS network traffic data need not be preprocessed in FECC and the accountability of the features thus extracted is particularly high as it is done using a process of Forward Propagation. This was useful as the model can get trained with the features without multiple epochs of training thereby reducing the training time. This technique also proved to detect DNS tunneling that was of unknown types. However, this model was not capable of working efficiently with real-time network traffic and worked well with only an existing traffic sample. Furthermore, considering the requirement of cloud platforms where data comes in real-time, this technique seemed to be inefficient. Similarly, a deep neural mechanism combined with word embedding for performing feature extraction in the neural network fitting was proposed by Zhang et al. (2019) for DNS tunneling detection. They used different deep learning models like Recurrent Neural Network, Convolutional Neural Network, and Dense Neural Network for their experimental analysis. This technique enabled the detection of DNS tunneling with a single DNS request packet. Furthermore, the malicious top-level domain is removed and the extraction of the subdomain is done. Therefore, this technique accurately detects DNS tunneling. The deep learning model that has the highest prediction accuracy is assigned to be the DNS tunnel detection decision maker. Therefore this technique is capable of detecting McAfee's DNS tunneling. The major drawback of this technique is its low interpretability.

Chen et al. (2021) put forward a DNS tunnel detection model using Long Short Term Memory(LSTM) which is a Recurrent Neural Network. In this model, the DNS covert channels were detected with an LSTM network using the Fully Qualified Domain Names also referred to as FQDNs. A filtering mechanism that was packet-based was used on the DNS covert channel. Additionally, feature extraction of FQDN using the Long Short Term Memory provided more generality in the exfiltration detection procedure thereby resulting in reduced false positive results. Therefore, this technique provided a higher accuracy and reliability rate for the results when compared to other deep learning models.

However, an effective and incremental learning technique is required in cloud platforms as improved accuracy, reliability, and security are the major performance metrics concerning cloud platforms. In the paper Altuncu et al. (2021) a Deep Feed Forward based neural network(DFF) mechanism for exfiltration detection was proposed. The basic principle of this mechanism is that the hidden layers of the network are combined to produce the required results. The authors trained the DNS network traffic data initially with a Deep Feed Forward(DFF) network having 2 hidden layers, then 3, and finally, 5 hidden layers to evaluate the reliability of the mechanism with real-time traffic. It was observed that the accuracy of the results was directly proportional to the number of hidden layers in the model. The response time of the DFF algorithm in detecting DNS exfiltration when compared with other machine learning algorithms was found to be higher. However, it was found that the model complexity seemed to increase with the number of hidden layers and also this model failed to provide a secure, reliable, and inter-platform operability in cloud platforms.

2.4 DNS exfiltration detection in Cloud Environments

A cloud-based DNS exfiltration detection technique was proposed in Borges et al. (2022). In this technique, an unsupervised learning algorithm was used in the cloud platforms for accurately detecting DNS exfiltration. The learning was performed on the data collected from the AWS cloud platform wherein the DNS registration, DNS packet routing, and DNS packet health monitoring are performed by Route 53 which is an AWS cloud service. The features of the DNS traffic data are divided into dimensions of which Top-Level Domain dimensions are used to construct the learning model. Due to this reason, this mechanism ensures advanced security against low bandwidth attacks. One of the main limitations of this technique is that the accuracy of the technique is highly dependent on the recent DNS traffic data as this technique uses an unsupervised learning model. This implies the need for an incremental learning process that boosts the efficiency of detecting DNS tunneling. Salat et al. (2023) proposed a technique for DNS exfiltration detection on cloud environments such as Amazon Web Service and Google using tools like DNScat and Iodine. In this technique, the DNS monitoring was done using an open-source tool named Elastic Stack. They performed DNS tunneling in two scenarios- one with firewall protection and the other without. A DNS exfiltration detection was performed in both scenarios by performing checks on the DNS logs. This was achieved by using methods like analysis of DNS signature, DNS traffic, time, DNS filtering, and DNS payload. However, this technique indicated that a real-time monitoring system is required to detect DNS exfiltration, and as this technique does not make use of any machine learning technique, the security and reliability of exfiltration detection in real-time cannot be guaranteed.

2.5 Other Works

Lundteigen Mohus and Henry Flakk (2022) proposed a technique that uses a conditional GAN(Generative Adversarial Networks) which can be combined with unsupervised machine learning models like K-Means to form an ensemble for DNS tunneling detection. Static features and noise are fed as conditionals to the generator. The features thus generated are given as input to the discriminator and during the training phase of the model, a backpropagation is performed. Even though this technique is useful in detecting malicious DNS requests, it is unreliable with real-time DNS network traffic data.

2.6 Comparison of Major Related Works

A comparative analysis of the different related works is shown in table 2

Table 2: Comparative Analysis of Major Related Works

Research	**Technique Used**	**Advantage**	**Disadvantage**
This Research	CLAM-Hybrid model of Convolutional Neural Network and Long Short Term Memory(CNN-LSTM) with attention mechanism	Improved accuracy of detection in cloud network with reduced false positives	
Alkasassbeh and Almseidin (2023)	Supervised learning models like J48, Random Forest, between, and Multilayer Perceptron with cross-validation	Evaluation of feature relationships for the data in the network traffic	Increased false positives and unreliability
Zhan et al. (2022)	DNS over HTTPS tunneling detection using TLS fingerprinting and training using Decision Tree, Linear Regression, and Random Forest	Enhanced safety and security for DNS packet transmission	Easily bypassed by adversarial attacks
Liang et al. (2023)	Feature extraction using Convolutional Neural Network and Clustering for evaluation(FECC)	Reduced training time with the use of forward propagation	Inefficient with real-time network traffic as in cloud
Borges et al. (2022)	Training of AWS data using an unsupervised technique by partitioning DNS traffic data dimensions	Enhanced security even against attacks of low bandwidth	Accuracy dependent on more recent network traffic data
Lundteigen Mohus and Henry Flakk (2022)	cGAN combined with unsupervised learning and backpropagation	Effective in malicious traffic detection	Unreliable with real-time data

3 Methodology

DNS exfiltration detection in the cloud has become a critical challenge in the current era, particularly a reliable technique with high accuracy and fewer false positives is extremely essential. This helps improve data security and has different applications. Moreover, the cloud has now turned out to be the data storage and transmission platform. Thus, in this research, we propose a hybrid deep learning model CLAM (Convolutional Neural Network and Long Short Term Memory(CNN-LSTM) with an Attention Mechanism). This ensures a highly reliable, scalable, and secure learning model in detecting DNS-based exfiltration in cloud platforms. CLAM acts as a filtering technique in the cloud network that can filter out exfiltration attacks and thereby ensure the flow of safe DNS requests through the cloud network, making the cloud platform more reliable and secure for individuals and organizations. Figure 2 provides the overview of the methodology used in this research to develop CLAM.

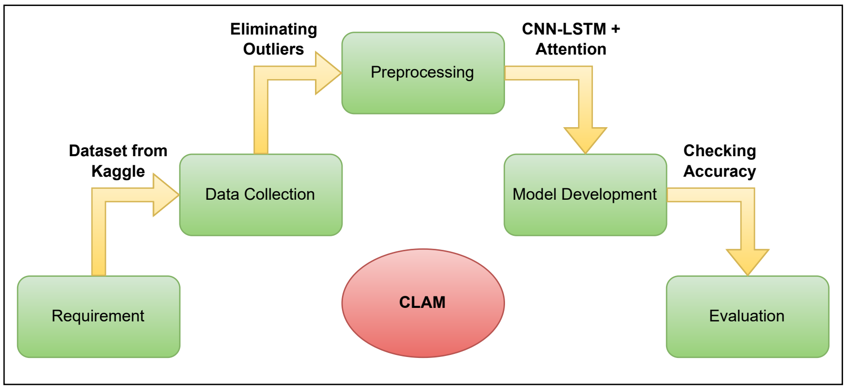


Figure 2: DNS exfiltration detection methodology using CLAM

The major steps involved in this research are as follows:

1. **Requirement:** To effectively and accurately detect the exfiltration attacks in the cloud network and safeguard the confidential data in cloud platforms.

2. **Data Collection:** The data required for training the model is obtained from a publicly available website, the University of New Brunswick(UNB) [1]. The NSL-KDD data set is a modified version of the KDD data set Tavallaee et al. (2009) consisting of training and testing data namely, KDDTrain+ and KDDTest+. The training data consists of around 125973 records and each record consists of 42 columns. There are 38 target classes for training representing normal data and other types of suspicious data. All other target classes except normal are considered exfiltration attacks.

3. **Preprocessing:** To make the data ready for training, a preprocessing on the dataset is performed which eliminates any outliers in the data, compensates for any missing value, and ensures the dimensions of the data are compatible with the training model thereby balancing the dataset. Furthermore, binning and sampling techniques are applied to the dataset. Also, KDDTrain+ itself is divided into train and test data during preprocessing.

4. **Model Development:** The CLAM(CNN-LSTM with Attention Mechanism) model is developed by combining a one-dimensional Convolutional Neural Network(CNN) and a Long Short Term Memory(LSTM) and finally applying an attention on top of them. The dataset is trained on this development model for detecting exfiltration attacks.

[1] https://www.unb.ca/cic/datasets/nsl.html

5. **Evaluation:** The trained model is evaluated to ensure the efficiency of the model using different metrics like accuracy, false positive, and confusion matrix. Additionally, a comparison with existing models is also performed.

4 Design Specification

The CLAM model is an advanced machine-learning model with a learning model module and a network module. The network module represents the cloud network through which the DNS traffic flows and where the filtered normal traffic after training passes through. The main components of the learning module part are Convolutional Neural Network(CNN), Long Short Term Memory(LSTM), and a feature highlighting part called attention mechanism.

- **CNN:** A CNN is a deep-learning model with different layers like convolutional layers, max pooling layers, and fully connected layers. The convolutional layer calculates the weighted sum of the inputs, whereas the pooling layer is responsible for performing dimensionality reduction on the input features and they generalize the Convolutional Neural Network model. The fully connected layer is responsible for outputting the class scores of the target variables using the activation function applied to the network. The CNN networks are thus an effective machine learning model O'Shea and Nash (2015).

- **LSTM:** An LSTM is a type of Recurrent Neural Network(RNN) of deep learning that is used for problems having long-term dependencies. A Long Short Term Memory(LSTM) has different memory blocks interconnected recurrently which enables the information flow through the logic gates. Hence, they are used mainly for applications involving pattern analysis, analysis of sentiments, and Natural Language Processing(NLP) techniques Hochreiter and Schmidhuber (1997).

- **Attention Mechanism:** Attention mechanism is a specific technique that is applied to the learning model to focus particular parts or features of an input sequence. Thus, this highlighting mechanism ensures that more weightage is given to the relevant features of the input data, unlike the traditional techniques where all the features or attributes in the input data are given equal importance and weightage.

CLAM Architecture
In this research, a hybrid model combining CNN and LSTM with the above-mentioned attention mechanism is used for detecting DNS data exfiltration in the cloud network. The input data is preprocessed and the categorical data is converted to numerical data using a '*LabelEncoder*'. This preprocessed data is passed on to a CNN network and LSTM network. In the CNN part, a 1-dimensional CNN layer with 64 filters and kernel size 3 is used, and thereafter a 1-dimensional max pooling layer ensures the dimensionality reduction of the data. This helps in extracting the high-level features of the input data thereby reducing the number of features, especially the ones that are irrelevant for the training. Finally, a global average pooling function is used that further reduces the dimensionality of the data by taking the average of the elements. This also increases the robustness of the data to noise or distortions. The activation function applied to the CNN is '*relu*'.

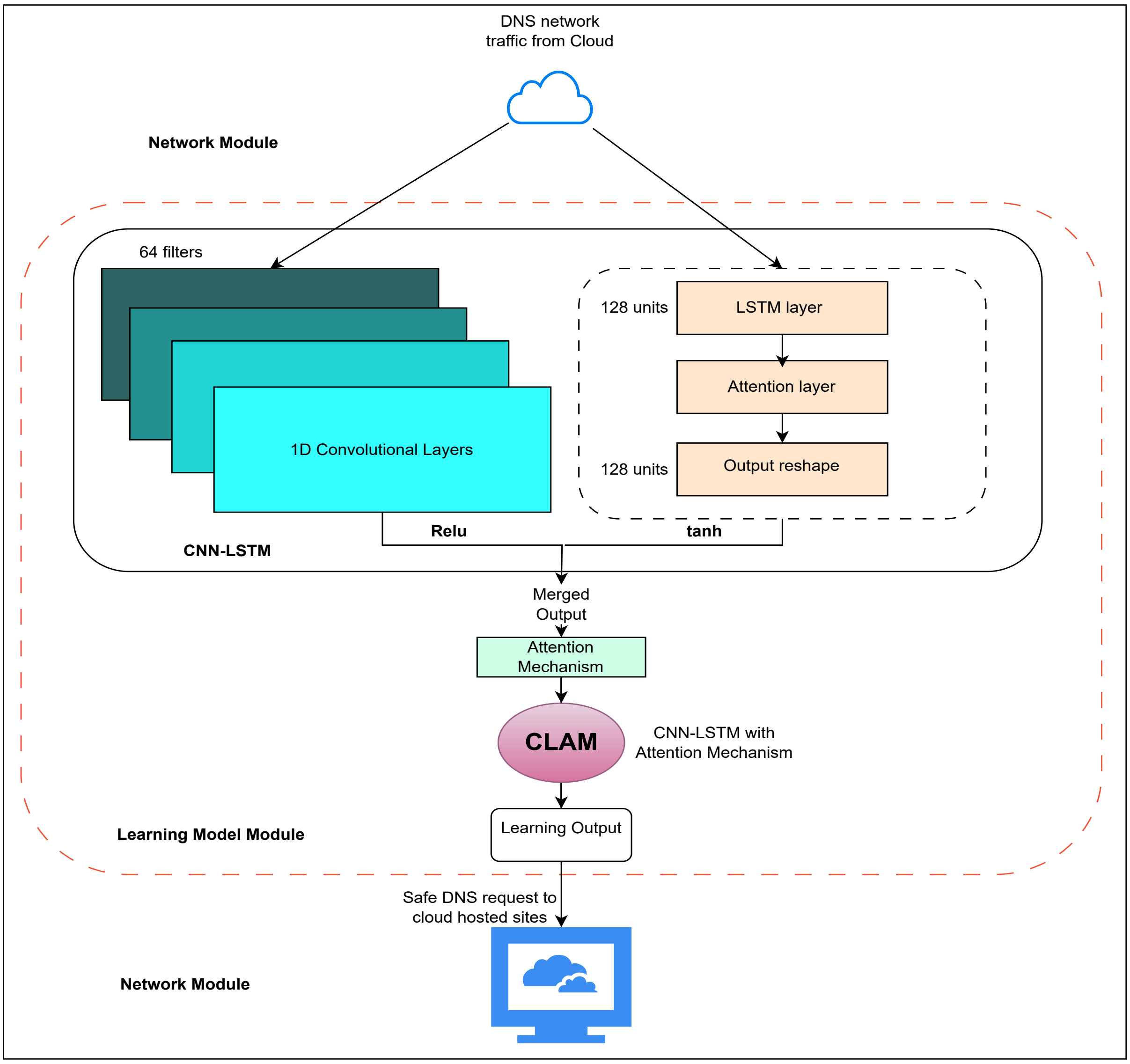


Figure 3: Architectural Diagram of CLAM

In the LSTM part, the input data is fed to an LSTM layer with 128 units and an activation function '*tanh*'(hyperbolic tangent) is used. Thereafter an attention mechanism is applied over it which compares all the features from the LSTM layer to compute the attention weights and thereby determine the most relevant features in them. Here also a dimensionality reduction is performed using a global average pooling layer. The outputs from the CNN part and LSTM part are combined and again an attention mechanism is applied over the concatenated output of the CNN and LSTM. This double-checks and ensures that only the most relevant features for detecting DNS exfiltration are given the maximum focus, thereby ensuring the reliability and accuracy of the model. A reshape() and flatten() functions are applied to this for output reshaping to a one-dimensional array. This output is then passed through a series of dense layers for further processing of the attention-applied output data and to form fully connected layers. Figure 3 represents the architectural diagram of the DNS exfiltration detection learning model.

The algorithm for the hybrid model for the CNN-LSTM with Attention Mechanism(CLAM) learning model proposed in this research is given below:

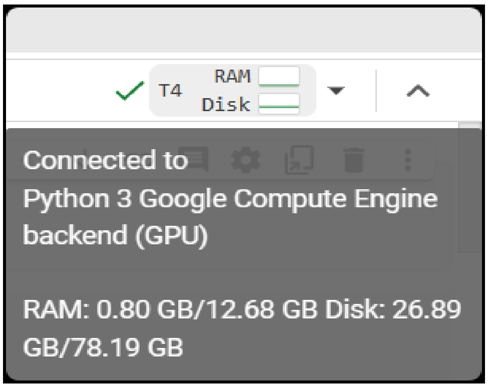
Algorithm 1 Algorithm for the CNN-LSTM with Attention Mechanism learning model

```
1. Input: NSL-KDD dataset(KDDTrain+ and KDDTest+)
2. Preprocess the input data into a 1-dimensional array by removing the outliers
   and perform dimensionality reduction on the data
3. Initialize the model and feed the input data to the 1D Convolutional Neural
   Network(CNN) with 64 filters and to a Long Short Term Memory(LSTM) with
   128 units and apply attention on the output of LSTM
4. Combine the output from CNN and LSTM and apply the attention mechanism
   on the concatenated output
5. Train the model with this data for 10 epochs and validate the results on the
   validation data while reducing the model loss
6. Perform optimization on the trained model by performing parameter tuning
7. Evaluate the performance of the model by calculating performance metrics like
   accuracy and precision
8. Deploy the trained model in a Google Colab environment, which is a cloud platform,
   for detecting DNS exfiltration in cloud networks
```

5 Implementation

The dataset used for this research is the NSL-KDD dataset consisting of the train and test datasets - KDDTrain+ and KDDTest+. The training and testing data are combined for training the CLAM model. Each record in the dataset has 43 columns where column 42 represents the target variable. The 38 target classes of the dataset are given in table 3. The learning model requires a cloud platform and Google Colab with a Python 3 runtime type and T4 Graphical Processing Unit(GPU) is used for the same GoogleColab (n.d.). Google Colab provides a cloud machine learning platform with zero configuration and high-speed data processing using GPUs. The datasets are then uploaded to Google Drive from where it is accessed while running the code. The specification of the Google Colab platform used for training the model is shown in figure 4. The model is developed using Python 3 and different Python library packages are required to develop the learning model like pandas, matplotlib, sklearn, tensorflow, and numpy.

The combined dataset is preprocessed by performing a label encoding on each of the columns for converting the categorical data into numerical data to ensure that the data format and dimensions are compatible with the machine learning model. A transformation is applied to the data as shown in figure 5 to make the dimensions acceptable for the 1D CNN and LSTM which accepts data as a sequence. Also reshaping of the input is performed and a histogram representing the target variables is plotted to represent the occurrence of each target variable. This input is then passed through the CNN and LSTM models and the attention mechanism is applied to the merged output. A combined CNN-LSTM model was developed initially and the performance of the model was evaluated. To make the model more efficient in exfiltration detection, attention was added



Connected to
Python 3 Google Compute Engine backend (GPU)

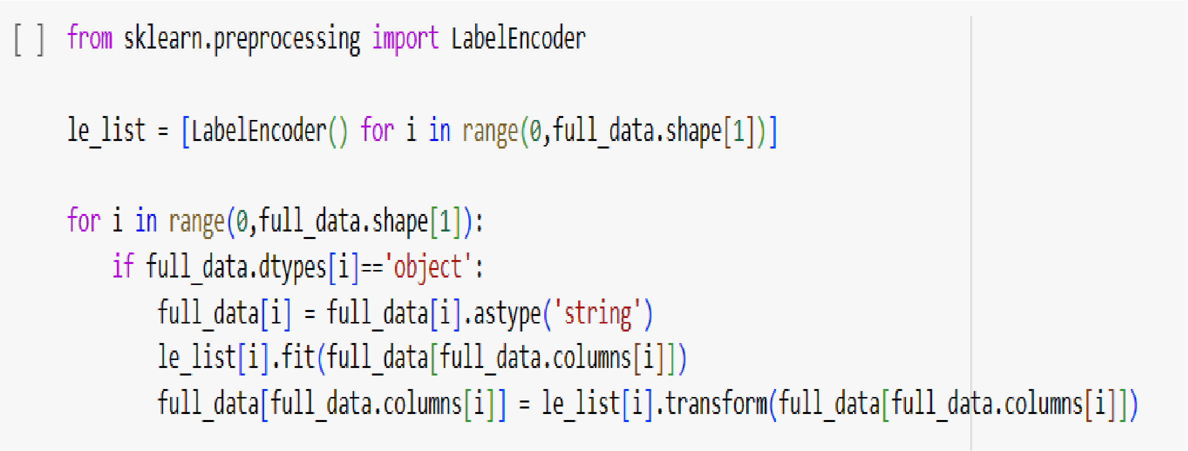
RAM: 0.80 GB/12.68 GB Disk: 26.89 GB/78.19 GB

Figure 4: Specifications of Google Colab Platform

on top of them as a feature highlighting mechanism. Three dense layers are added to this to establish a dense connection with the previous layer. A dropout layer is added to the dense layers which helps in preventing overfitting. The loss function applied to the model is '*categorical crossentrophy*' and the model is trained in 10 epochs. During each epoch, the accuracy of the model increases and the model loss decreases. The accuracy and the loss of the model are then plotted using Matplotlib which aids in visualization of the model performance, and a confusion matrix is also generated. Thereafter the performance of the model is further evaluated on the KDDTest+ dataset and a classification report of the same is also generated.

Table 3: Target Classes of the dataset and their assigned index values

Target Class	**Index Number**
apache2	0
back	1
buffer_overflow	2
ftp_write	3
guess_passwd	4
httptunnel	5
imap	6
ipsweep	7
land	8
loadmodule	9
mailbomb	10
mscan	11
multihop	12
named	13
neptune	14
nmap	15
normal	16
perl	17
phf	18
pod	19
portsweep	20
processtable	21
ps	22
rootkit	23
saint	24
satan	25
sendmail	26
smurf	27
snmpgetattack	28
snmpguess	29
sqlattack	30
teardrop	31
udpstorm	32
warezmaster	33
worm	34
xlock	35
xsnoop	36
xterm	37



```
[ ] from sklearn.preprocessing import LabelEncoder

    le_list = [LabelEncoder() for i in range(0,full_data.shape[1])]

    for i in range(0,full_data.shape[1]):
        if full_data.dtypes[i]=='object':
            full_data[i] = full_data[i].astype('string')
            le_list[i].fit(full_data[full_data.columns[i]])
            full_data[full_data.columns[i]] = le_list[i].transform(full_data[full_data.columns[i]])
```

Figure 5: Data Preprocessing for model training

6 Evaluation

The CLAM model is evaluated using some performance metrics like accuracy, false positives, and precision. In a cloud network, assessing these factors is essential to ensure the security, reliability, and efficiency of the data in the cloud. The existing classification algorithms and deep learning models like CNN and LSTM are compared with the developed CLAM model to evaluate the performance of CLAM.

6.1 Evaluation of Existing Classification Algorithms

The existing classification algorithms like the Random Forest Classifier, Decision Tree, Gradient Boost Classifier, K-Nearest Neighbour, Naive Bayes, Logistic Regression, Support Vector Machine, and Extreme Gradient Boosting are evaluated using the KDDTrain+ dataset. This is done using a technique called Automated Machine Learning(AutoML) which consists of different machine-learning libraries that automate the data training process Sihombing et al. (2022). This dataset is evaluated using Pycaret which is an open source library in Python. It enables the comparison of the performance of different machine-learning models on the KDDTrain+ dataset.

6.2 Evaluation of Deep Learning Models

Deep learning models were used to test the performance with the KDDTrain+ dataset. For this purpose, deep learning models like Convolutional Neural Network (CNN) and Long Short Term Memory(LSTM) were used as the training models. The CNN model with two one-dimensional convolutional layers each of 32 filters was trained in 10 epochs. The activation function used by the CNN layer is '*relu*' while the two dense layers use '*softmax*' and '*relu*' activation functions. The LSTM model with 2 LSTM layers each of 6 memory units was trained in 10 epochs and the activation function used in the single dense layer is '*softmax*'.

6.3 Evaluation of the CLAM Model

The CLAM model has one convolutional layer of 64 filters and '*relu*' activation function in the 1D CNN part and one LSTM layer with 128 units and '*tanh*' activation function in the LSTM part. The merged output is further processed by applying the attention mechanism with three layers of dense layers for training the dataset. The model is trained in 10 epochs where the model validation is performed on 30% of the dataset in KDDTrain+. An open-source library TensorFlow is used in developing the CLAM model.

6.4 Results

The results of the above three evaluations are calculated using a few cloud machine-learning performance metrics Chen et al. (2021):

1. **Accuracy:** Accuracy is the most important evaluation metric of a machine learning model. It determines the percentage of predictions that are correct of the total predictions made.

$$Accuracy = \frac{Numberofcorrectpredictions}{TotalPredictions} \tag{1}$$

2. **Precision:** Precision is a parameter that determines the percentage of correct predictions of the positive class. This helps in determining the reliability of the model. A highly precise model is a highly reliable model for the cloud platforms.

3. **Rate of False Positives:** This represents the percentage of negative target classes that have been incorrectly classified as positive classes. The lower the false positive rate for the model, the higher the reliability and security.

$$RateofFalsePositives = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{2}$$

Apart from that a confusion matrix representing the summary of the classification with True Positives, True Negatives, False Negatives, and False Positives is also used to evaluate the model.

Results of Existing Classification Models

In the existing classification models, the Decision Tree(DT) classifier provided the highest training accuracy of 94.31% followed by the Random Forest(RF) classifier with an accuracy of 94.23%. Accuracies for the Gradient Boosting classifier, k-Nearest Neighbour, and Extreme Gradient Boosting are 93.45%, 92.70%, and 82.35% respectively. Therefore, the Decision Tree classification model is the best model among these but the reliability of the model cannot be assured as equal weightage is given to all features. The validation curve for the Decision Tree classifier is shown in figure 6.

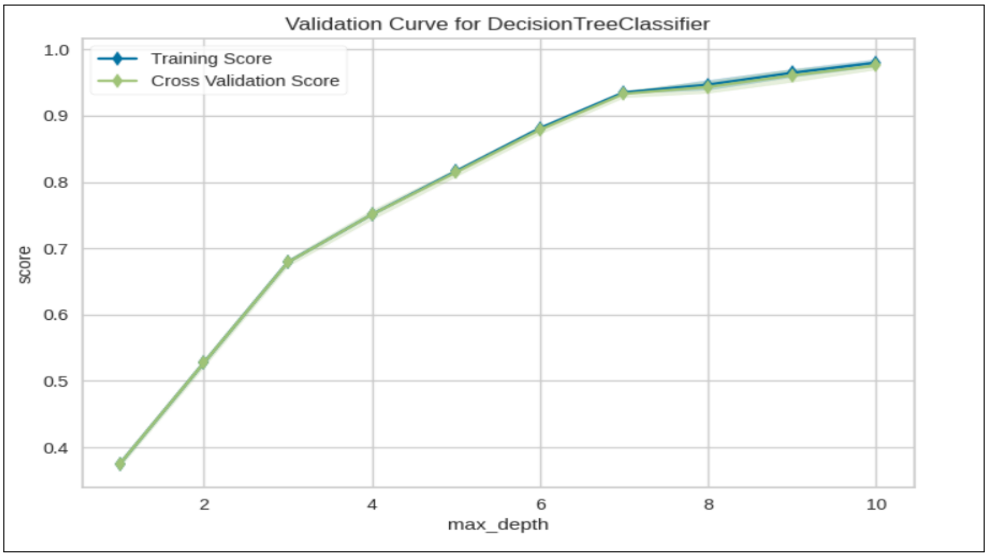


Figure 6: Decision Tree classifier validation curve

Results of Deep Learning Models

The performance of the CNN and LSTM deep learning models was evaluated in DNS exfiltration detection and it was found that the Convolutional Neural Network(CNN) model provided an accuracy of 88.74%. Whereas the Long Short Term Memory(LSTM) model provided an accuracy of 92.43%. The graphs representing the training accuracy and training loss of the CNN and LSTM models are given in figure 7 and figure 8 respectively. Here, all the features were given equal weightage thereby reducing the reliability of the model in making accurate and precise predictions.

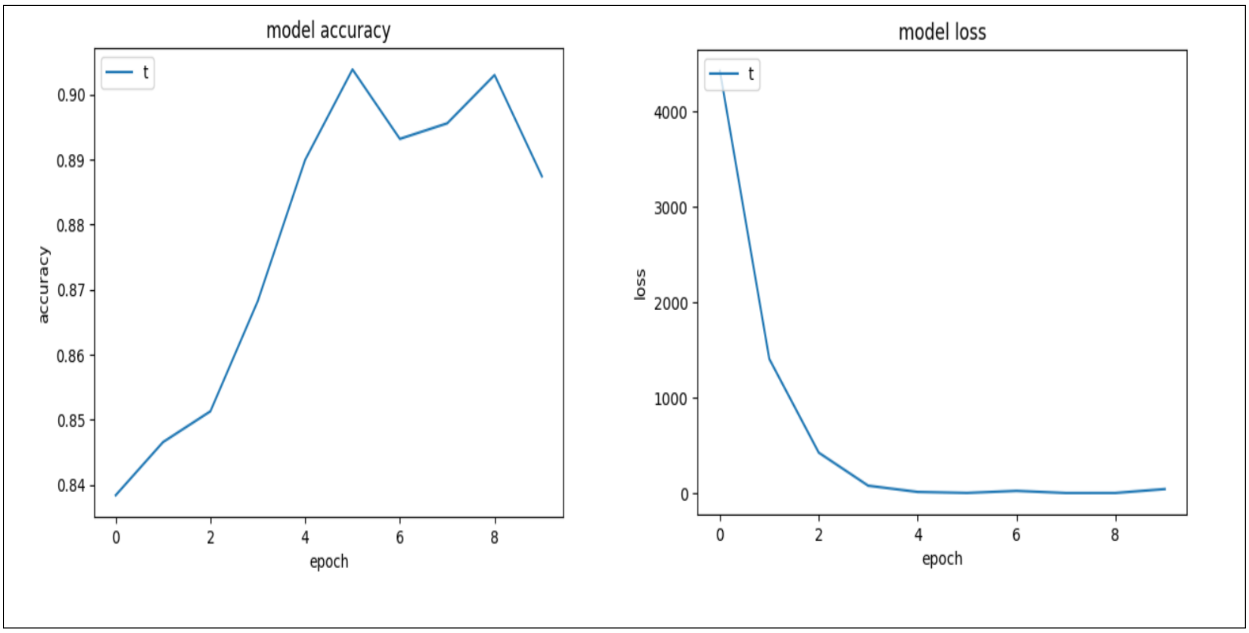


Figure 7: Graphs representing training accuracy and loss of the CNN model with epochs

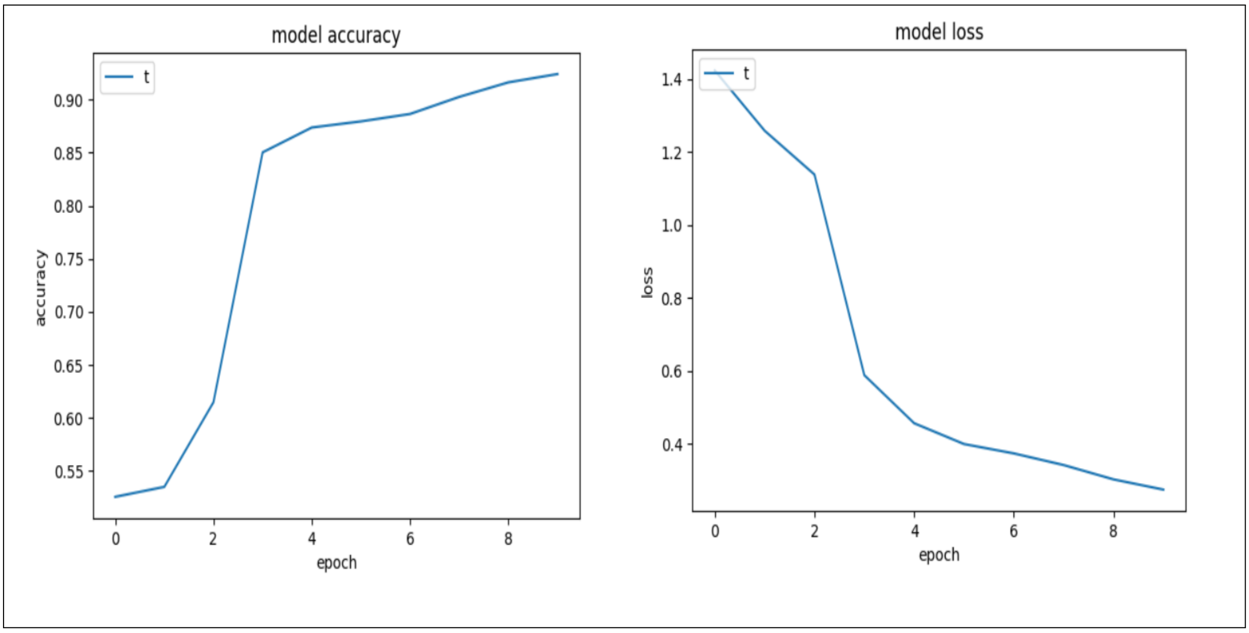


Figure 8: Graphs representing training accuracy and loss of the LSTM model with epochs

Results of CLAM Model

The developed CLAM model consisting of a 1D CNN and LSTM model with an attention mechanism provides an accuracy of 95.46% in DNS exfiltration detection in a cloud platform. A confusion matrix representing the predictions of target classes is represented which demonstrates a low number of false positive predictions. This increases the reliability of the CLAM model in cloud networks. The training accuracy and training loss of the CLAM model over the 10 epochs are demonstrated as a graph in figure 9.

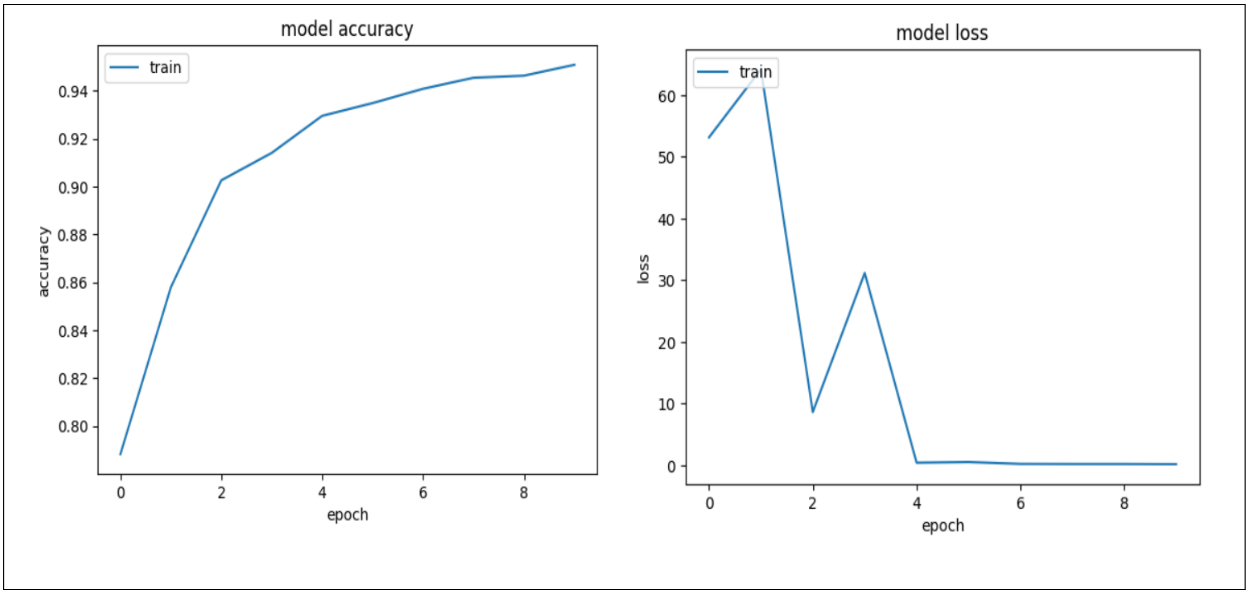
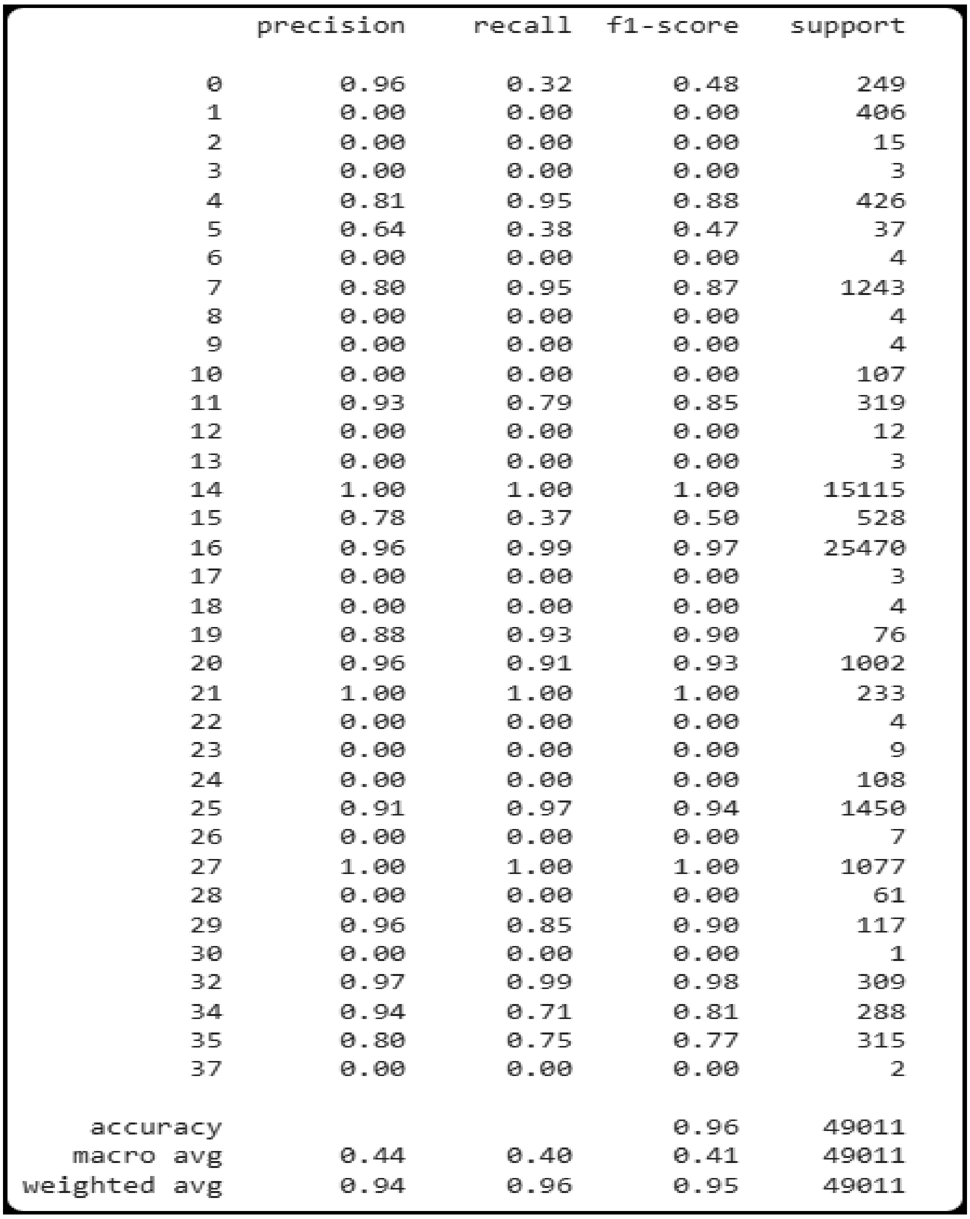


Figure 9: Graphs representing training accuracy and loss of the CLAM model with epochs

The classification report of the model representing the accuracy is demonstrated in figure 10.



```
              precision    recall  f1-score   support

           0       0.96      0.32      0.48       249
           1       0.00      0.00      0.00       406
           2       0.00      0.00      0.00        15
           3       0.00      0.00      0.00         3
           4       0.81      0.95      0.88       426
           5       0.64      0.38      0.47        37
           6       0.00      0.00      0.00         4
           7       0.80      0.95      0.87      1243
           8       0.00      0.00      0.00         4
           9       0.00      0.00      0.00         4
          10       0.00      0.00      0.00       107
          11       0.93      0.79      0.85       319
          12       0.00      0.00      0.00        12
          13       0.00      0.00      0.00         3
          14       1.00      1.00      1.00     15115
          15       0.78      0.37      0.50       528
          16       0.96      0.99      0.97     25470
          17       0.00      0.00      0.00         3
          18       0.00      0.00      0.00         4
          19       0.88      0.93      0.90        76
          20       0.96      0.91      0.93      1002
          21       1.00      1.00      1.00       233
          22       0.00      0.00      0.00         4
          23       0.00      0.00      0.00         9
          24       0.00      0.00      0.00       108
          25       0.91      0.97      0.94      1450
          26       0.00      0.00      0.00         7
          27       1.00      1.00      1.00      1077
          28       0.00      0.00      0.00        61
          29       0.96      0.85      0.90       117
          30       0.00      0.00      0.00         1
          32       0.97      0.99      0.98       309
          34       0.94      0.71      0.81       288
          35       0.80      0.75      0.77       315
          37       0.00      0.00      0.00         2

    accuracy                           0.96     49011
   macro avg       0.44      0.40      0.41     49011
weighted avg       0.94      0.96      0.95     49011
```

Figure 10: Classification report representing the accuracy of the CLAM model

6.5 Discussion

The attention-applied CNN-LSTM hybrid model in the cloud platform effectively detected the 38 target classes thereby identifying the normal DNS queries and malicious DNS queries with an average accuracy of 96% which ensures that DNS-based exfiltration attacks in the cloud can be detected reliably using the CLAM model. When compared with the existing classification models and deep learning models, the CLAM model provided more accuracy and fewer false positives thereby ensuring an efficient model that can be deployed in any cloud platform. The accuracies of the different models evaluated are given in table 4.

Table 4: Comparison of different models based on accuracies

Model	**Accuracy(%)**
CLAM	95.46
Decision Tree	94.31
Random Forest	94.23
Gradient Boosting	93.45
k-Nearest Neighbour	92.70
Long Short Term Memory	92.43
Convolutional Neural Network	88.74
Extreme Gradient Boosting	82.35

7 Conclusion and Future Work

Detection and prevention of DNS-based data exfiltration in the cloud network is challenging especially where data is critical and sensitive. This requires an efficient detection technique with a mechanism to highlight the features responsible for detecting exfiltration. With the rapid evolution of machine learning, research has been conducted to develop a machine learning model that is reliable and secure and more importantly can detect DNS exfiltration in cloud environments accurately.

In this research, a hybrid model combining a one-dimensional Convolutional Neural Network(CNN) and Long Short Term Memory(LSTM) with an attention technique(CLAM) was used in effectively detecting DNS exfiltration in a cloud network. The developed CLAM model was a reliable and secure model with an accuracy of about 95.46%. Moreover, the false positive rate of the model for the target classes was less compared to the other models.

In the future, CLAM can be modified to include the prevention technique along with the detection mechanism for DNS exfiltration by integrating an auto-monitoring and filtering technique in the cloud infrastructure. Also, it can be trained using multiple real-time cloud datasets to improve the speed and efficiency of detections.

References

Abualghanam, O., Alazzam, H., Elshqeirat, B., Qatawneh, M. and Almaiah, M. A. (2023). Real-time detection system for data exfiltration over dns tunneling using machine learning, *Electronics* **12**(6): 1467.

Alkasassbeh, M. and Almseidin, M. (2023). Machine learning techniques for accurately detecting the dns tunneling, *Science and Information Conference*, Springer, pp. 352–364.

Altuncu, M. A., Gülağiz, F. K., Özcan, H., Bayir, Ö. F., Gezgın, A., Nıyazov, A., Çavuşlu, M. A. and Şahın, S. (2021). Deep learning based dns tunneling detection and blocking system, *Adv. Electr. Comput. Eng* **21**(3): 39–48.

Borges, L. d. S. B., de Oliveira Albuquerque, R. and de Sousa Júnior, R. T. (2022). A security model for dns tunnel detection on cloud platform, *2022 Workshop on Communication Networks and Power Systems (WCNPS)*, IEEE, pp. 1–6.

Buczak, A. L., Hanke, P. A., Cancro, G. J., Toma, M. K., Watkins, L. A. and Chavis, J. S. (2016). Detection of tunnels in pcap data by random forests, *Proceedings of the 11th Annual Cyber and Information Security Research Conference*, pp. 1–4.

Chen, S., Lang, B., Liu, H., Li, D. and Gao, C. (2021). Dns covert channel detection method using the lstm model, *Computers & Security* **104**: 102095.

GoogleColab (n.d.).
URL: *https://colab.research.google.com/*

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory, *Neural computation* **9**(8): 1735–1780.

Liang, J., Wang, S., Zhao, S. and Chen, S. (2023). Fecc: Dns tunnel detection model based on cnn and clustering, *Computers & Security* **128**: 103132.

Lundteigen Mohus, M. and Henry Flakk, E. (2022). Conditional gans against dns based unsupervised detection of malicious domains, *2022 International Conference on Cyberworlds (CW)*, pp. 261–262.

Nadler, A., Aminov, A. and Shabtai, A. (2019). Detection of malicious and low throughput data exfiltration over the dns protocol, *Computers & Security* **80**: 36–53.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks, *arXiv preprint arXiv:1511.08458* .

Salat, L., Davis, M. and Khan, N. (2023). Dns tunnelling, exfiltration and detection over cloud environments, *Sensors* **23**(5): 2760.

Shafieian, S., Smith, D. and Zulkernine, M. (2017). Detecting dns tunneling using ensemble learning, *Network and System Security: 11th International Conference, NSS 2017, Helsinki, Finland, August 21–23, 2017, Proceedings 11*, Springer, pp. 112–127.

Sihombing, D. J. C., Dexius, J. U., Manurung, J., Aritonang, M. and Adinata, H. S. (2022). Design and analysis of automated machine learning (automl) in powerbi application using pycaret, *2022 International Conference of Science and Information Technology in Smart Administration (ICSINTESA)*, pp. 89–94.

Singh, S. K. and Roy, P. K. (2020). Detecting malicious dns over https traffic using machine learning, *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, IEEE, pp. 1–6.

Steadman, J. and Scott-Hayward, S. (2022). Detecting data exfiltration over encrypted dns, *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)*, IEEE, pp. 429–437.

Tavallaee, M., Bagheri, E., Lu, W. and Ghorbani, A. A. (2009). A detailed analysis of the kdd cup 99 data set, *2009 IEEE symposium on computational intelligence for security and defense applications*, Ieee, pp. 1–6.

Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M. A. and Rashid, A. (2018). Data exfiltration: A review of external attack vectors and countermeasures, *Journal of Network and Computer Applications* **101**: 18–54.

Zhan, M., Li, Y., Yu, G., Li, B. and Wang, W. (2022). Detecting dns over https based data exfiltration, *Computer Networks* **209**: 108919.

Zhang, J., Yang, L., Yu, S. and Ma, J. (2019). A dns tunneling detection method based on deep learning models to prevent data exfiltration, *Network and System Security: 13th International Conference, NSS 2019, Sapporo, Japan, December 15–18, 2019, Proceedings 13*, Springer, pp. 520–535.