An Effective Fog-aware NIDS for DDoS Attack Detection

Ibrahim Moshen Selim, Rowayda A. Sadek

Abstract—**The Internet of Things (IoT) has introduced issues to traditional cloud infrastructure, leading to the emergence of an intermediary architectural design called fog computing. Fog computing, a subtype of cloud computing, is being utilized to address some of the challenges that cloud computing infrastructure faces, particularly reducing the response time or detection time when fog devices are often powered by limited energy sources. In particular, distributed denial of service (DDoS) attacks should be mitigated by fog-computing devices. This is possible if network traffic is continuously monitored by a Network Intrusion Detection System (NIDS) to detect DDoS attack patterns. This paper proposes a NIDS model called "DDoS-BiLSTM" for detecting DDoS attacks, specifically in fog computing, using deep learning (DL). Appropriate preprocessing and modeling phases were incorporated into the proposed model. The BiLSTM model employing the CICIDS2017 dataset was the foundation for the proposed fog-based NIDS model for detecting DDoS attacks. The results obtained were superior, with an accuracy of 99.91%. Numerous records of various DDoS attack types from multiple datasets were combined into a newly integrated NF-UQ-NIDS dataset. The proposed model was trained and validated with 99.62% accuracy using this dataset.**

Index Terms—**Internet of Things (IoT), Fog Computing, Intrusion Detection System, CICIDS2017 dataset, NF-UQ-NIDS dataset, Deep Learning (DL).**

I. Introduction

Current computing technology has been influenced by the exponential rise of connected smart devices. The term 'Internet of Things' (IoT) refers to the increase of non-computer elements (things) that are connected to the Internet. Recent advancements in wearable technology, smart cities and homes, connected cars, smart traffic lights, smart meters, and other areas have made the IoT popular [1]. The main purposes of IoT devices are to gather and send data for cloud processing and to obtain feedback or outcomes for decision-making. With billions of IoT devices joining the Internet infrastructure, IoT technology has become a vital part of our daily lives. Despite the rise in the popularity of IoT devices, they suffer from several difficulties, including limited battery life, storage, bandwidth, and computing power [2].

These issues negatively impact the user experience and quality of service (QoS). Cloud computing is regarded as an appropriate platform for providing services to customers to reduce the difficulties that IoT devices must overcome [3]. However, cloud computing does not offer a universally applicable answer to the issues affecting IoT performance. In 2012, Cisco presented Fog Computing as a solution to this issue [4].

By 2020, 50 billion devices were expected to connect to the Internet, and by 2025, 500 billion were projected to do so, according to Cisco. This suggests that data production will increase and that users will demand speedy responses and feedback. Therefore, fog computing aims to bring services closer to the end users of these devices [5]. Fog computing is a subset of cloud computing that offers effective middle-tier services to consumers. However, the same cannot be said for fog computing, despite the security benefits associated with cloud computing.

Fog computing has come to be recognized as a potential paradigm for effective and decentralized data processing in today's connected world, where data are produced at an unprecedented rate. Fog computing enables data processing and analysis closer to the data source by extending cloud computing capabilities to the network's edge [6]. This implies that the benefits and drawbacks of cloud computing cannot be directly transferred to fog computing.

Fog computing has security issues that are particularly relevant to this new paradigm. Security concerns have taken center stage due to the exponential growth of data and devices in fog computing environments. Distributed Denial of Service (DDoS) attacks pose a significant risk to these environments. Fog computing can be subjected to DDoS attacks that seek to overwhelm the network and disrupt services because of its distributed architecture and reliance on connected devices [7].

Deploying a network intrusion detection system (NIDS) is a key aspect of security in fog computing [8]. A NIDS is a security tool that monitors network traffic to spot malicious activity or unauthorized access attempts and take appropriate action. Traditionally, centralized systems such as data centers or cloud environments have used conventional NIDS solutions. However, because fog computing is distributed and heterogeneous, NIDS must be adapted to address this particular context.

Strategic placement of several fog nodes with detection sensors or actuators within the fog infrastructure is necessary for an NIDS to function effectively in fog computing [9]. These sensors continuously monitor and analyze network data, searching for patterns or anomalies that may indicate potential security risks. Network traffic can be monitored in real-time by an NIDS in fog computing, which can analyze patterns and behaviors to identify signs of DDoS attacks. It can detect common indicators of DDoS attacks, such as anomalous traffic spikes, unusual packet patterns, or large volumes of traffic originating from multiple sources.

The principal contribution of this paper is a proposal for an effective fog-based NIDS to accurately detect DDoS attacks

Manuscript received December 20, 2023; revised October 1, 2024.

Selim I. M. is an Assistant Lecturer at Information Technology Department, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt. (e-mail: ibrahiselim@fci.helwan.edu.eg).

Sadek R. A. is a Professor at Information Technology Department, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo, Egypt. (e-mail: rowayda_sadek@fci.helwan.edu.eg).

targeting fog nodes and breaching fog services. The following is a summary of the contributions made by the proposed NIDS model:

- A lightweight fog-based NIDS for identifying various DDoS attacks has been proposed.
- The proposed fog-based NIDS uses the NF-UQ-NIDS dataset to overcome the limitations of the CICIDS2017 dataset used in existing works, including small size, synthetic traffic, confined labeling, outdated attacks, and constrained network topology.
- Preprocessing and optimized feature selection techniques are applied.
- Various deep learning techniques are implemented to compare and select the best one, including LSTM, BiLSTM, GRU, SimpleRNN, and MLP.
- Using the CICIDS2017 and NF-UQ-NIDS datasets, a comparative analysis is specifically conducted with existing fog-based NIDSs. The experimental results show that the proposed method outperforms existing works in terms of metrics such as accuracy, precision, recall, and F1-score.

This paper is organized as follows: Background information about fog computing, intrusion detection systems, and distributed denial of service attacks is presented in Section II. Related work, both in general and specifically in the fog computing environment, is discussed in Section III. In Section IV, we present the proposed NIDS and several foundational concepts. The results, performance, and comparisons with other methods are detailed in Section V. We outline future work in Section VI. Section VII presents the conclusions drawn from the developed method.

II. Background

This section covers intrusion detection systems and fog computing as well as how to use them to identify DDoS attacks. We began by providing a brief overview of fog computing and its significance, followed by an overview of intrusion detection systems that work in fog. We discuss DDoS and how it affects the fog environment in the last part.

A. Fog Computing

Thousands of Internet of Things (IoT) devices coexist at the network's edge due to ongoing developments aimed at performing daily tasks across a range of industries, including smart industrial systems, smart homes, and smart vehicles. Fog computing initially emerged in the IoT space to facilitate the execution of time-sensitive applications and services. A highly virtualized fog-computing platform provides real-time networking, storage, and computing services between endpoints and traditional cloud data centers [1].

Fog computing can be positioned adjacent to IoT devices to create a novel network architecture, as shown in Figure 1. Fog nodes, or fog servers, are typically placed near the edge of the network, close to the IoT devices, within a fog computing architecture. By serving as a bridge between IoT devices and the cloud, these fog nodes offer local processing and storage capabilities. They can perform a variety of tasks, including preprocessing, analytics, data filtering, and real-time decision-making. Fog computing enhances navigation assistance, scalability, interoperability, and location awareness [6].

A fog computing network consists of switches, routers, proxy servers, base stations (BS), and other components with varying computing, storage, and networking capabilities. Effective performance in terms of latency, power usage, and network traffic can be achieved using fog computing. Possible interactions between the cloud, fog, and edge layers include:

- **Fog to Cloud**: The fog node is directly connected to the cloud data centers.
- **Fog to Fog**: The fog nodes are located near one another.
- **Edge to Fog**: The fog node is directly connected to edge devices, such as cellphones, sensors, and small processor boards.

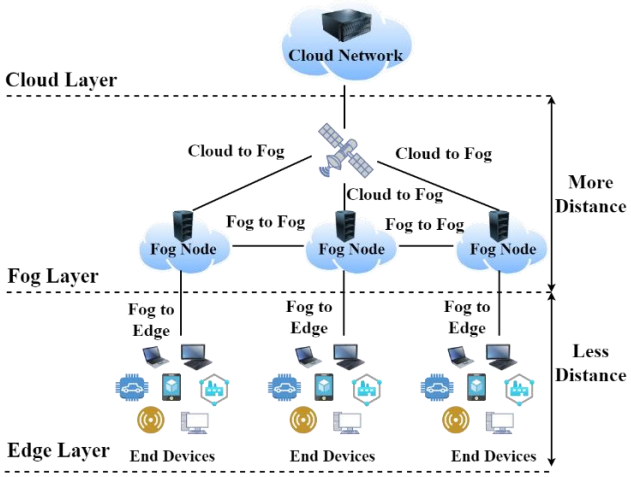

Fig. 1. Computing's Hierarchical Architecture ([1] and copyright obtained).

B. Intrusion Detection System

An Intrusion Detection System (IDS) is a vulnerability avoidance system that monitors data obtained from various sources to protect the network by identifying potential attacks. The purpose of an IDS is to analyze the data, look for patterns or attack signatures, check system correlations, and create alarms if any matches are found [10].

In addition, IDSs maintain records of recognized patterns or signatures. Network monitoring collects data from network packets. Attackers use various methods to conduct attacks on a network. These attacks may target a server that manages all network transactions or a host machine that performs activities within the network. IDSs utilize deep learning and machine learning to recognize network threats. Data can be acquired from several IoT device sources for intrusion analysis. Depending on the information source, intrusion detection systems are divided into two categories: host IDS (HIDS) and network IDS (NIDS) [11].

C. Distributed Denial of Service

Distributed Denial of Service (DDoS) refers to a specific type of cyberattack in which multiple compromised systems, often known as a "botnet," are used to overwhelm a network or website with traffic, exceeding its capacity and blocking access for authorized users [12]. By flooding a network or website with traffic from various sources, DDoS attacks seek

to prevent normal network operation. As a result, users may experience system or service slowness or complete collapse.

Fog computing environments can also be targeted by DDoS attacks [13]. Fog computing extends processing power to the edge of the network. While this allows applications to operate more rapidly and efficiently, it also opens up new attack vectors that DDoS attackers can exploit. Figure 2 illustrates how DDoS attacks can cause systemic disruption by overloading network devices and fog nodes' processing and communication capabilities.A person or organization in charge of a network of compromised computers, referred to as "bots," is called a "bot master." These bots can be remotely managed by the bot master without the computer owners' knowledge, and they are usually infected with malware. To protect fog computing environments from DDoS attacks, safety mechanisms must be implemented at every tier of the architecture.

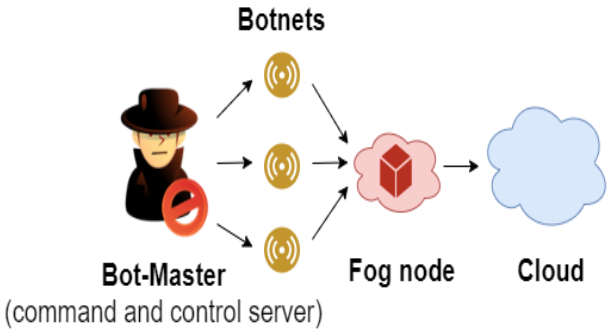

Fig. 2. IoT Botnet Attacks in Fog Computing

III. Related Work

Several recent studies have addressed NIDSs for DDoS attacks. By observing network traffic patterns and spotting unusual behavior that can point to an ongoing attack, an NIDS for DDoS detection attempts to detect and mitigate DDoS attacks. These systems use various methods, including statistical analysis, machine learning, and signature-based detection, to detect and reduce DDoS attacks. Algorithms for machine learning and deep learning can be trained to assess network traffic patterns and detect DDoS attack-related irregularities.

These methods can be implemented in fog nodes to provide real-time detection and mitigation of attacks on the network edge. The most recent advancements in NIDS for DDoS detection have concentrated on improving the accuracy and effectiveness of detection algorithms, strengthening the resistance of systems to complex attacks, and tackling the difficulties caused by emergent environments such as fog computing. This section discusses the most recent NIDS works as well as prior studies in the fog computing environment to detect DDoS attacks on IoT.

To accurately identify various application-layer DDoS attacks, Asad et al. [14] presented a novel deep neural network-based detection mechanism that uses feed-forward backpropagation. On the CICIDS2017 dataset, which contains several types of DDoS attacks, the proposed neural network architecture can identify and use the most important high-level packet flow components with an accuracy of 98%. To test this method, only application-layer DDoS attacks are used.

Sabeel et al. [15] suggested two ML models, DNN and LSTM, for binary classification of unidentified DoS and DDoS attacks. The benchmark CICIDS2017 dataset was used to train the models. DNN and LSTM performed these tasks with accuracy rates of 98.72% and 96.15%, respectively. However, real-time detection was not performed, and the authors used only binary classification.

Haider et al. [16] suggested a deep CNN framework for the detection of DoS assaults in SDN. With a total accuracy rate of 99.45%, the ensemble CNN technique surpassed other competing methods that were already in use. This method extends the timeframes. Consequently, the mitigation mechanism can be compromised, making attacks more harmful.

Wang et al. [17] suggested using information entropy and the DL approach to identify DDoS attacks in an SDN context. With a rate of 98.98%, CNN exceeded its competitors in terms of precision, accuracy, f1-score, and recall. The time detection process in the model was longer.

An IDS was developed by Monika et al. [18] using a combination of a Convolutional Neural Network (CNN) integrating Long Short-Term Memory (LSTM) deep learning techniques for identifying the attack and the NSGA-II multi-objective optimization method for data dimension reduction. The experiment had a 99.03% accuracy rate and used the most recent CISIDS2017 statistics for DDoS attacks.

Mural et al. [19] developed a deep classification strategy to recognize HTTP sluggish DoS attacks. The CICIDS2017 dataset was used to evaluate classifiers. The obtained findings show that the model classifies attacks with an overall accuracy of 99.61%. The limitation of this methodology was that only slow HTTP DoS attacks were evaluated.

For IoT intrusion detection at the fog computing layer, Souza et al. [20] described a hybrid binary classification solution utilizing deep neural networks (DNN) and the k-nearest neighbor (KNN) technique. They tested their strategy using a publicly accessible dataset (CICIDS2017) and found that it had a high accuracy of 99.85% in identifying attacks.

IV. Proposed Work

Fog computing is susceptible to DDoS attacks. Fog computing systems are typically constructed using a large number of networked devices and sensors, which makes them vulnerable to such attacks.

As shown in Figure 3, fog layer nodes receive packets from IoT devices. These nodes employ deep learning (DL) to create a Network Intrusion Detection System (NIDS) model that is subsequently applied to data analysis and intrusion detection. The NIDS at the fog node gathers traffic, generates security alerts, logs warnings, and transmits them to neighboring fog nodes and cloud servers when an intrusion is detected [21].

An NIDS monitors network activity, analyzes data from IoT devices, checks system configurations for security flaws, identifies suspicious patterns or signs, stores these patterns in a database, and sends them to the cloud layer, where a warning is issued if any matches are found. An improved NIDS becomes feasible through the use of machine learning

(ML) and deep learning technologies for detecting network threats.

The proposed fog-based NIDS model has been evaluated on the NF-UQ-NIDS dataset and is divided into several phases, each with a different purpose.

- **Phase 1**: Before applying deep learning techniques to the data, preprocessing techniques should be employed to enhance and prepare the data. Preprocessing involves preparing data for analysis and modeling by cleaning and enhancing it. The techniques required to preprocess a dataset for the proposed model are as follows:
  - Data cleaning removes noisy and irrelevant data and ensures that feature types are accurate.
  - Feature selection involves determining the most important features of the data using appropriate techniques.
  - Feature encoding converts feature values into numerical representations.
  - Feature scaling adjusts feature values to a specified range.

- **Phase 2**: In the modeling phase, recurrent neural networks (RNNs) of the BiLSTM type can process sequential data both forward and backward. BiLSTM is currently one of the most powerful deep learning models. The use of BiLSTM has made attack detection more precise and effective.

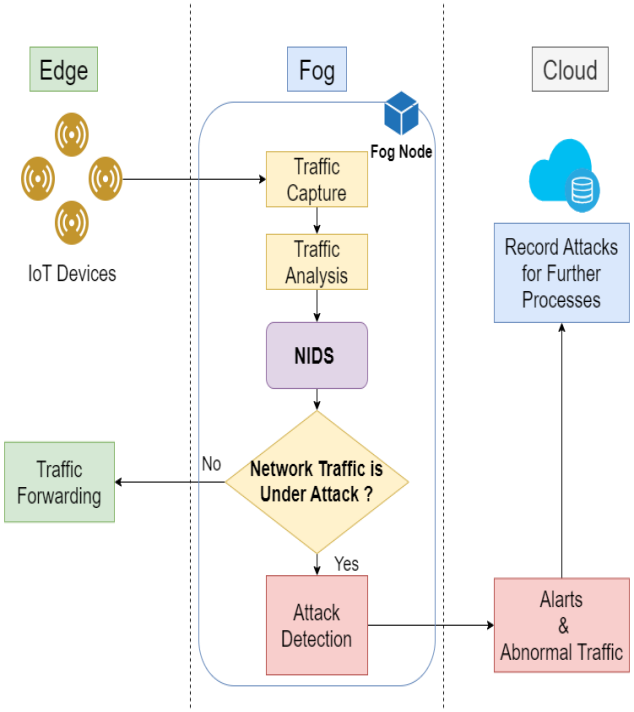

Fig. 3. Graph of the Proposed NIDS Architecture's Flow

As shown in Figure 4, the fog-based NIDS model is composed of several phases, the outcomes of which are carried over into the subsequent phase. The proposed DDoS-BiLSTM NIDS is divided into several phases, each of which performs a specific task.

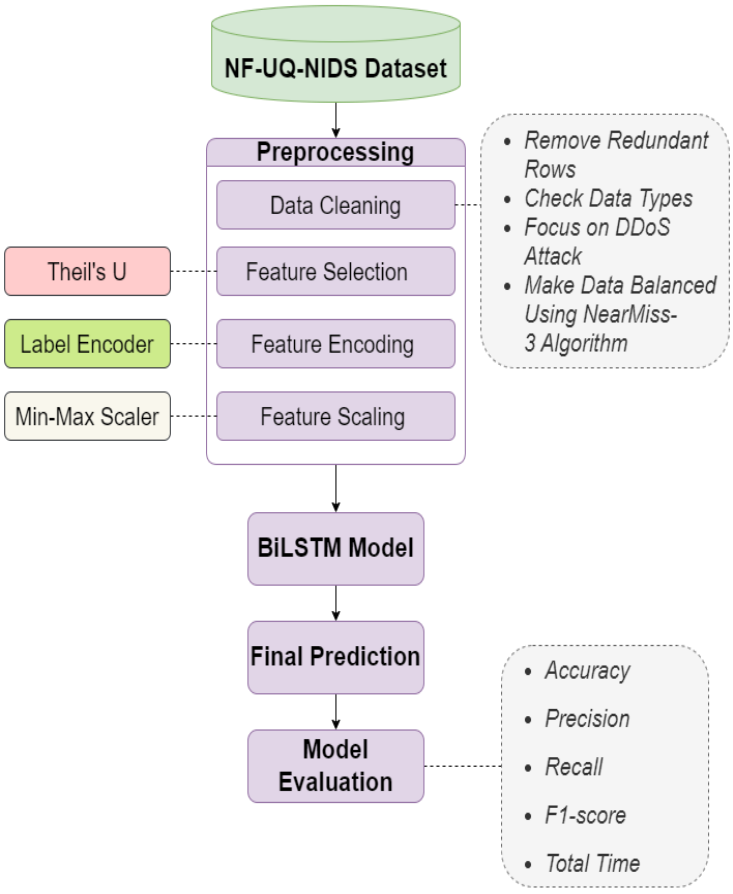

Fig. 4. The Fog-Based DDoS-BiLSTM Model

A. Preprocessing Phase

Data preprocessing prepares data and can be done more rapidly and effectively in data science to ensure reliable outcomes. Fog nodes exist between IoT devices and the cloud layer. IoT devices receive a variety of feature formats, including numeric and categorical data, in incoming communications. To increase the effectiveness of the proposed fog-based NIDS model, the analysis and preprocessing of the traffic must be conducted as described below.

1) Data Cleaning Process

Duplicate data rows should be removed, as their existence distorts the data and affects the outcomes. The data cleaning process in our work ensures that feature types are accurate, eliminates noisy and irrelevant data, balances the data, and focuses specifically on DDoS attacks. Undersampling is a method for balancing unequal datasets, ensuring that the minority class is fully represented while minimizing the size of the majority class. Using NearMiss-3, samples from the majority class are selected based on their proximity to the minority class and their average distance from their k nearest neighbors [22]. The advantages of NearMiss-3 include optimal noise reduction, improved generalization performance, minimized bias, and the preservation of important information.

2) Feature Selection Process

Feature selection is crucial for deep learning and statistical modeling to improve model performance, reduce overfitting, and increase interpretability. The most effective solution for a particular problem depends on the specific data and modeling goals; no single feature selection technique addresses all challenges. The Theil's U [23] technique uses a filter method and statistics to evaluate the relationship between two categorical variables. Theil's U is a normalized variant of the mutual information measure that accounts for differences in the distributions of the feature and the target variable. This approach assesses redundancy among features as well as the information exchange between each feature and the target variable.

3) Feature Encoding Process

Feature values areconverted into numerical representations as part of the feature encoding process. Label encoding can use less memory and process data faster than other encoding techniques, such as one-hot encoding [24]. This is because label encoding, unlike one-hot encoding, only requires one column to represent a categorical variable. Given the abundance of categorical data in the datasets used, label encoding is employed.

4) Feature Scaling Process

The feature scaling process is one of the most important operations. Scaling is crucial when developing a deep learning model, as it affects the model's performance. Among the scaling techniques, normalization and standardization are the most frequently used. Min-Max scaling [25] is employed to apply normalization, scaling, and translating each feature separately to ensure it falls within the desired range.

B. Modeling Phase

Machine learning (ML) and deep learning (DL) techniques are applied to model the dataset and produce a predictive model that can be used to make predictions or to apply to new data points. The goal of modeling is to find relationships and patterns in the data that can be used to categorize new data or predict future occurrences. Several deep learning techniques can be applied to the preprocessed data. The effectiveness of each technique must be evaluated before selecting the best one. Techniques such as LSTM, BiLSTM, GRU, SimpleRNN, and MLP have been used.

1) Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional LSTM (BiLSTM) is a recurrent neural network primarily used in natural language processing [26]. It is a valuable tool for observing relationships between phrases and words in both directions of the sequence since, unlike standard LSTM, the input data flows in two directions, allowing it to utilize data gathered from both sides.

The BiLSTM model was developed from the LSTM architecture and consists of two LSTMs: one processes input in the forward direction and the other in the backward direction. The four layers that comprise the BiLSTM include the input layer, the forward transmission layer, the reverse transmission layer, and the output layer [26]. BiLSTM is capable of handling sequential input, capturing both past and future context, and addressing vanishing gradient problems. It can achieve high performance due to its long-term memory capabilities.

V. Experimental Results And Discussion

Several attack detection techniques will be evaluated using our proposed fog-based NIDS. Experiments were conducted on an Anaconda machine using Python and Jupyter Notebooks. The TensorFlow and Keras packages employ various learning methods. The performance of the proposed NIDS is assessed using the NF-UQ-NIDS dataset. These tests were carried out on a laptop running 64-bit Windows 10, equipped with an Intel® Core i5-8265U CPU at 1.80 GHz and 12 GB of RAM.

A. NF-UQ-NIDS Dataset

The CICIDS2017 [27] dataset was used in most earlier studies to test NIDSs for identifying DDoS attacks. Although it is a helpful dataset for network security practitioners and researchers, it has several shortcomings that must be addressed, including its small size, synthetic traffic, constrained labeling, outdated attacks, and limited network topology. The proposed NIDS utilizes the NF-UQ-NIDS dataset to overcome these issues.

This dataset is derived from Cisco's NetFlow network protocol, which is used to track network traffic flow [28]. Datasets created and analyzed with the NetFlow protocol are known as NetFlow datasets. NetFlow datasets provide valuable insights into network traffic patterns, containing information on the protocols and ports used, the sources and destinations of the traffic, and the volume of traffic moving between various parts of the network.

The NF-UQ-NIDS [29] dataset combines the UNSW-NB15 [30], ToN-IoT [31], BoT-IoT [32], and CSE-CIC-IDS2018 [33] datasets in a NetFlow-based format. This newly released dataset demonstrates the benefits of shared dataset feature sets, enabling the combination of several smaller datasets. Ultimately, a broader and more comprehensive NIDS dataset will be produced, encompassing flows from various network topologies and

attack scenarios. The updated attack categories incorporate all parent categories.

The NF-UQ-NIDS collection includes 9,208,048 records of various attack categories, including DDoS, reconnaissance, injection, DoS, brute force, password attacks, XSS, infiltration, exploits, scanning, fuzzers, backdoors, generic attacks, analysis, theft, shellcode, MITM, worms, and ransomware [29].

B. *Experiment Scenarios*

Fog nodes can be configured with NIDS to detect and mitigate malicious traffic. To spot irregularities that might indicate an ongoing DDoS attack, NIDS can monitor traffic patterns. This model focuses specifically on recognizing DDoS attacks. The NF-UQ-NIDS dataset contains 763,285 records for DDoS attacks and 9,208,048 records for benign flows. All datasets that comprise the NF-UQ-NIDS dataset include records of DDoS attacks, except for the UNSW-NB15 dataset.

In contrast to earlier datasets, the NF-UQ-NIDS dataset contains a substantial number of novel attacks and their feature values. Therefore, it was crucial to address data issues, such as removing unnecessary and redundant data, ensuring accurate data types, and handling missing data. After eliminating redundant rows, focusing on DDoS attacks, and discarding records from the UNSW-NB15 dataset, the total number of network traffic records is 5,747,026 for benign and 305,588 for DDoS.

The Theil's U statistic is a valuable method for feature selection in learning techniques and data analysis, as it helps identify the most significant features contributing to variation or inequality in the target variable. Its non-parametric nature, robustness, interpretability, flexibility, and decomposability make it an excellent choice for feature selection. Using Theil's U, the most important features are selected based on their scores, with the highest-scoring features representing the top eight.

Since deep learning models only work with numerical data, nominal or categorical features must first be transformed into numerical values. The label encoder method is used for encoding because it creates a clean data frame and is quick and simple to apply. The features are scaled using the Min-Max scaling method, also referred to as normalization, to a range between 0 and 1. Each feature's minimum and maximum values in the dataset are identified, and depending on where each value falls between these extremes, it is scaled to a number between 0 and 1.

A training set and a test set were created from the dataset, with the training set containing 70% of the data and the test set holding the remaining 30%. The test set is used to evaluate how well the model performs when applied to unlabeled data, while the training set is used to train the model on the preprocessed features. The dataset underwent preprocessing, and the resulting features were then fed into deep learning techniques for learning. Several deep learning techniques can be applied to the preprocessed data. It is essential to assess the performance of each technique using appropriate evaluation metrics before selecting the best one. Techniques such as LSTM, BiLSTM, GRU, SimpleRNN, and MLP were used.

The proposed model employs a deep learning architecture consisting of two BiLSTM layers, each containing 32 units, followed by a single-cell output layer with a sigmoid activation function for binary classification. To mitigate overfitting between training and testing, dropout layers with a rate of 0.2 were incorporated after each BiLSTM layer. This regularization technique ensures that the model generalizes effectively to unseen data, enhancing overall performance.

In this architecture, BiLSTM layers are responsible for capturing complex, bidirectional temporal dependencies in the data, improving classification accuracy. The output layer uses the sigmoid function, ideal for binary classification, converting the final layer's output into a probability score between 0 and 1. The model is trained using binary cross-entropy loss to measure the error between predicted and actual values.

Optimization is performed with the Adam optimizer, a widely used and effective method that adjusts learning rates adaptively during training. The model utilizes ReLU activation in the BiLSTM layers to introduce non-linearity, which enhances the model's ability to learn intricate patterns in network traffic data. Training is conducted for 50 epochs with a batch size of 32, balancing computational efficiency and convergence.

Figure 5 demonstrates the reconstruction loss for both the training and testing phases, showcasing the model's ability to effectively learn from the dataset without overfitting. Meanwhile, Figure 6 presents the model's accuracy progression over time, highlighting the steady improvement of both training and testing accuracy as the number of epochs increases. After 50 epochs, the model achieves optimal accuracy, demonstrating the effectiveness of the architecture and training configuration in handling the NF-UQ-NIDS dataset.

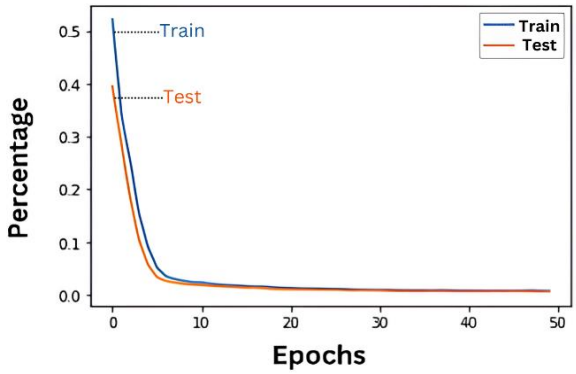

Fig. 5. Reconstruction Loss for Binary Classification using BiLSTM

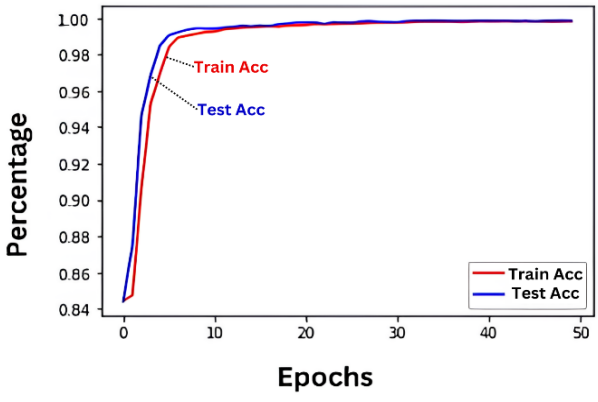

Fig. 6. Accuracy for Binary Classification using BiLSTM

A. Evaluation Metrics

On the CICIDS2017 dataset, we evaluated our proposed NIDS model against other models used for the same dataset, and we subsequently applied it to the new NF-UQ-NIDS dataset that is suitable for the fog computing environment. In comparison, our work yields exceptional and extraordinarily high results. Several metrics were used to assess performance.

1) *Accuracy:*

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2) *Precision:*

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

3) *Recall:*

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

4) *F1-score:*

$$F1\text{-}score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Where,

TP (True Positive): The proportion of instances that were successfully identified as attacks.

FP (False Positive): The proportion of instances that were misclassified as attacks.

TN (True Negative): The number of cases that were mistakenly labeled as normal.

FN (False Negative): The number of instances that were wrongly labeled as attack.

The performance of the DL techniques that have already been discussed is seen in Table I. The results displayed indicate that when compared to all other pertinent DL techniques, BiLSTM performs the best. Here, we see that the BiLSTM performs noticeably better than the other techniques. By comparing the performance of the LSTM model and the BiLSTM model, for instance, we find that:

- The BiLSTM model provides an accuracy that is 0.13% better than the LSTM model.
- In comparison to the LSTM model, we achieve a recall with the BiLSTM model that is 0.29% higher.
- We achieve a precision that is 0.45% higher using the BiLSTM model than the LSTM model.
- When we use the BiLSTM model instead of the LSTM model, we can achieve an F1 score that is 0.1% higher.

TABLE I
PERFORMANCE OF PREVIOUSLY DISCUSSED DEEP LEARNING METHODS ON PREPROCESSED NF-UQ-NIDS DATASET

	Accuracy	Recall	Precision	F1-score
MLP	99.55%	97.69%	96.69%	99.21%
SimpleRNN	99.56%	97.73%	96.46%	99.51%
GRU	99.52%	97.64%	96.49%	99.32%
LSTM	99.49%	97.57%	96.41%	99.25%
BiLSTM	**99.62%**	**97.86%**	**96.86%**	**99.35%**

The dataset, the quantity and quality of training data, the hyperparameter settings, and other variables can all affect how well deep learning approaches operate. Therefore, to guarantee the robustness and generalizability of the performance comparisons, it is advised to take these factors into account and carry out thorough assessments, such as cross-validation or statistical significance testing.

Figure 7 displays the time to predict spent on each deep-learning technique which is the sum of time to train and time to predict. We discover that the BiLSTM model takes longer than other models even though it performs better. Overall, the MLP model takes the least time, although it doesn't work very well. Despite BiLSTM having the longest overall time, we discover convergence with the LSTM and GRU models at this point.

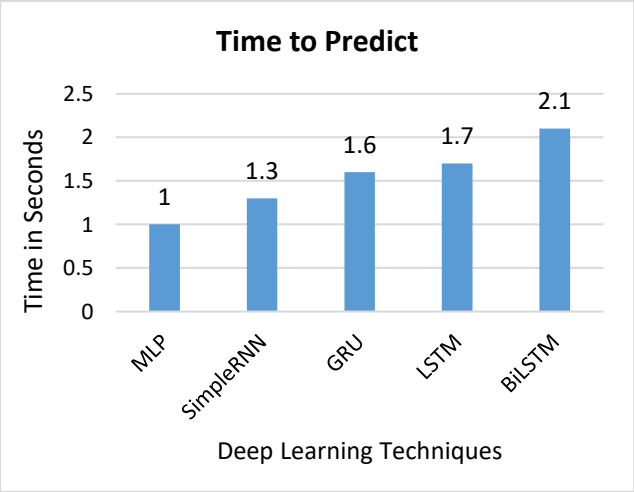

Fig. 7. Time to Predict for Each DL Model

The proposed model was chosen because it outperformed all other models; however, maintaining or attempting to improve these excellent results requires consideration of the time factor. Therefore, a two-phased model was implemented. Preprocessed data is fed into each deep learning technique to determine which one optimizes processing speed and performance. When testing the NF-UQ-NIDS dataset with the proposed model using BiLSTM, high result rates of 99.62% for accuracy, 96.86% for precision, 97.86% for recall, and 99.35% for the F1-score were achieved.

The CICIDS2017 dataset, as previously mentioned, is a well-known benchmark for evaluating network intrusion detection systems (NIDSs) in various studies. We applied the proposed model to the CICIDS2017 dataset to compare our results with those of these studies. Therefore, under these scenarios, evaluating IDSs becomes a superior choice.

The recall, accuracy, precision, and F1-score estimates for the proposed model using BiLSTM on the CICIDS2017 dataset are compared with previous works in Table II. High result rates of 99.91% for accuracy, 99.88% for precision, 99.73% for recall, and 99.60% for the F1-score were found when evaluating the CICIDS2017 dataset with the proposed model using BiLSTM. The prediction time is 3 seconds, which is more than a 0.30% increase when using the NF-UQ-NIDS dataset.

However, the computational efficiency of the proposed model is higher with the CICIDS2017 dataset, and its prediction time is greater compared to the NF-UQ-NIDS dataset. The NF-UQ-NIDS dataset indicates a reasonably efficient processing time for real-time intrusion detection. The findings suggest that the proposed fog-based model performs exceptionally well on the CICIDS2017 dataset compared to previous studies, making it a promising candidate for enhancing network security through effective intrusion detection.

TABLE II
THE PROPOSED MODEL ON THE CICIDS2017 DATASET COMPARED TO OTHER EXISTING STUDIES

Authors	Method	Dataset	Accuracy
Asad et al. [14]	Novel Deep Neural Network	CICIDS2017	98%
Sabeel et al. [15]	DNN	CICIDS2017	98.72%
	LSTM		96.15%
Haider et al. [16]	Deep CNN	CICIDS2017	99.45%
Wang et al. [17]	CNN	CICIDS2017	98.98%
Monika et al. [18]	CNN + LSTM	CICIDS2017	99.03%
Mural et al. [19]	Deep Classification Approach	CICIDS2017	99.61%.
Souza et al. [20]	DNN + KNN	CICIDS2017	99.85%
Proposed DDoS-BiLSTM Model	**BiLSTM**	**CICIDS2017**	**99.91%**

The proposed DDoS-BiLSTM model was also used on the ToN_IoT dataset, focusing on DDoS attacks, to compare its results with those obtained from the NF-UQ-NIDS dataset, as shown in Table III. High result rates of 98.77% for accuracy, 98.37% for precision, 98.73% for recall, and 99.18% for the F1-score were achieved when the ToN_IoT dataset was evaluated using BiLSTM. The results support the hypothesis that using the NF-UQ-NIDS dataset for detecting DDoS attacks in fog computing is advantageous, as it demonstrates that applying the proposed NIDS model to the NF-UQ-NIDS dataset outperforms its application to the CICIDS2017 dataset. In light of this, fog computing-based IoT networks can benefit from the NF-UQ-NIDS dataset, which is designed to leverage the unique features of the fog environment, such as diverse data sources, flexible network topology, and resource-constrained devices.

TABLE III
PERFORMANCE OF THE PROPOSED FOG-BASED DDOS-BILSTM NIDS ON THE NF-UQ-NIDS, CICIDS2017 AND TON_IOT DATASETS

Dataset Used	Accuracy	Precision	Recall	F1-score
NF-UQ-NIDS (DDoS)	99.62%	96.86%	97.86%	99.35%
CICIDS2017 (DDoS)	99.91%	99.88%	99.73%	99.60%
ToN_IoT (DDoS)	98.77%	98.37%	98.73%	99.18%

The NF-UQ-NIDS dataset offers a more complex and realistic testbed for the evaluation of NIDS in fog computing. By detecting DDoS attacks in real-world scenarios, the proposed model enhances fog computing security through its improved performance on the NF-UQ-NIDS dataset. This dataset is recommended for assessing DDoS attacks in fog computing, as it provides a clearer view of the challenges and complexities of this environment.

This highlights the potential benefits of using the NF-UQ-NIDS dataset for fog computing-based IoT networks. By utilizing the NF-UQ-NIDS dataset, these networks can be better equipped to detect, predict, and mitigate DDoS attacks, thereby improving their security posture and resilience.

VI. FUTURE WORK

The use of NIDS models based on deep learning is a popular and effective method for detecting and mitigating security risks in fog computing. Future work on this project could benefit from employing more sophisticated deep learning methods. Transfer learning techniques can enhance the performance of deep learning models in fog computing by pre-training them on large datasets and then fine-tuning them on smaller, domain-specific datasets. Novel techniques such as reinforcement learning, attention mechanisms, and one-shot learning can further improve the precision, efficiency, and adaptability of NIDS in fog computing.

VII. CONCLUSION

Fog computing is a powerful solution for reducing latency in time-sensitive IoT applications. NIDS is one of the most important tools for detecting new attack families. This study proposes NIDS models based on fog computing, which offer more effective and efficient solutions for intrusion detection in these environments, demonstrating the promise of deep learning-based techniques in addressing the challenges they face.

DDoS attacks pose a significant threat to the security of fog computing; thus, novel methods of detection and mitigation are required. The proposed fog-based NIDS-specific attack detection model includes preprocessing and modeling phases. During the preparation stage, the dataset is cleaned and balanced, after which significant features are selected using Theil's U method, encoded, and scaled for modeling. The BiLSTM model is utilized in the modeling step to achieve high accuracy compared to other DDoS attack detection techniques.

Numerous testing scenarios were conducted on diverse datasets. Many recent studies have used the CICIDS2017 dataset, which is outdated and does not include recent DDoS attacks. In this context, a more appropriate dataset is the NF-UQ-NIDS dataset, which combines multiple previous datasets and includes a large number of records of various DDoS attack types. The model was tested using the NF-UQ-NIDS dataset, yielding excellent results with 99.62% accuracy, 96.86% precision, 97.86% recall, and 99.35% F1-score. Additionally, this model was tested using the outdated CICIDS2017 dataset and performed exceptionally well compared to earlier research, achieving high rates of 99.91% accuracy, 99.88% precision, 99.73% recall, and 99.60% F1-score.

REFERENCES

[1] I. M. Selim, and R. A. Sadek, "DAE-BILSTM: A Fog-Based Intrusion Detection Model Using Deep Learning for IoT," *Journal of Theoretical and Applied Information Technology*, vol. 101, no. 5, 2023.

[2] G. S. S. Chalapathi, V. Chamola, A. Vaish, and R. Buyya, "Industrial internet of things (iiot) applications of edge and fog computing: A review and future directions," *Fog/edge computing for security, privacy, and applications*, pp. 293-325, 2021.

[3] L. J. M. Nieuwenhuis, M. L. Ehrenhard, and L. Prause, "The shift to Cloud Computing: The impact of disruptive technology on the enterprise software business ecosystem," *Technological forecasting and social change*, vol. 129, pp. 308-313, 2018.

[4] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE access*, vol. 6, pp. 47980-48009, 2018.

[5] I. Selim, and R. Sadek, "A Review of Intrusion Detection and Prevention Systems in Fog Computing Environment," *FCI-H Informatics Bulleti*n, vol. 3, no .2, pp. 17-22, 2021.

[6] M. R. Anawar, S. Wang, M. A. Zia, A. K. Jadoon, U. Akram, and S. Raza, "Fog computing: An overview of big IoT data analytics," *Wireless Communications and Mobile Computing 2018*, 2018.

[7] A. M. Alwakeel, "An overview of fog computing and edge computing security and privacy issues," *Sensors*, vol. 21, no .24, p. 8226, 2021.

[8] W. A. Mahmoud, M. Fathi, H. El-Badawy, and R. Sadek, "Performance Analysis of IDS_MDL Algorithm to Predict Intrusion Detection for IoT Applications," *2023 40th National Radio Science Conference (NRSC).IEEE*, vol. 1, 2023.

[9] S. Raponi, M. Caprolu, and R. D. Pietro, "Intrusion detection at the network edge: Solutions, limitations, and future directions," *Edge Computing–EDGE 2019: Third International Conference, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings 3*. Springer International Publishing, 2019.

[10] A. Abd El-Rady, H. Osama, R. Sadik, and H. El Badwy, "Network Intrusion Detection CNN Model for Realistic Network Attacks Based on Network Traffic Classification," *2023 40th National Radio Science Conference (NRSC). IEEE*, vol. 1, 2023.

[11] R. H. Mohamed, F. A. Mosa, and R. A. Sadek, "Efficient intrusion detection system for IoT environment," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 4, 2022.

[12] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: a survey," *The Journal of Supercomputing*, vol. *76*, pp. 5320-5363, 2020.

[13] L. Zhou, H. Guo, and G. Deng, "A fog computing based approach to DDoS mitigation in IIoT systems," *Computers & Security*, vol. 85, pp. 51-62, 2019.

[14] M. Asad, M. Asim, T. Javed, M. O. Beg, H. Mujtaba, and S. Abbas, "Deepdetect: detection of distributed denial of service attacks using deep learning," *The Computer Journal*, vol. 63, no .7, pp. 983-994, 2020.

[15] U. Sabeel, S. S. Heydari, H. Mohanka, Y. Bendhaou, K. Elgazzar, and K. El-Khatib, "Evaluation of deep learning in detecting unknown network attacks," *2019 International Conference on Smart Applications, Communications and Networking (SmartNets). IEEE*, 2019.

[16] S. Haider, A. Akhunzada, I. Mustafa, T. B. Patel, A. Fernandez, K. K. R. Choo, and J. Iqbal, "A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks," *Ieee Access*, vol. 8, pp. 53972-53983, 2020.

[17] L. Wang, and Y. Liu, "A DDoS attack detection method based on information entropy and deep learning in SDN," *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE*, vol. 1, 2020.

[18] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against ddos attacks in iot networks," *2020 10th annual computing and communication workshop and conference (CCWC). IEEE*, 2020.

[19] N. Muraleedharan, and B. Janet , "A deep learning based HTTP slow DoS classification approach using flow data," *ICT Express*, vol. 7, no. 2, pp. 210-214, 2021.

[20] C. A. D. Souza, C. B. Westphall, R. B. Machado, J. B. M. Sobral, and G. D. S. Vieira, "Hybrid approach to intrusion detection in fog-based IoT environments," *Computer Networks*, vol. 180, p. 107417, 2020.

[21] Y. Labiod, A. A. Korba, and N. Ghoualmi, "Fog computing-based intrusion detection architecture to protect iot networks," *Wireless Personal Communications*, vol. 125, no.1, pp. 231-259, 2022.

[22] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas, "Handling imbalanced datasets: A review," *GESTS international transactions on computer science and engineering*, vol. 30, no.1, pp. 25-36, 2006.

[23] M. F. Dzulkalnine, and R. Sallehuddin, "Missing data imputation with fuzzy feature selection for diabetes dataset," *SN Applied Sciences*, vol. 1, pp. 1-12, 2019.

[24] E. Jackson, and R. Agrawal, "Performance evaluation of different feature encoding schemes on cybersecurity logs," *IEEE*, 2019.

[25] P. Singla, M. Duhan, and S. Saroha, "Different normalization techniques as data preprocessing for one step ahead forecasting of solar global horizontal irradiance," *Artificial Intelligence for Renewable Energy Systems. Woodhead Publishing*, pp. 209-230, 2022.

[26] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," *2019 IEEE International conference on big data (Big Data). IEEE*, 2019.

[27] A. Yulianto, P. Sukarno, and N. A. Suwastika, "Improving adaboost-based intrusion detection system (IDS) performance on CIC IDS 2017 dataset," *Journal of Physics: Conference Series*, vol. 1192, 2019.

[28] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "Netflow datasets for machine learning-based network intrusion detection systems," *Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10*. Springer International Publishing, 2021.

[29] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile networks and applications*, pp. 1-14, 2022.

[30] M. S. Al-Daweri, K. A. Z. Ariffin, S. Abdullah, and M. F. E. M. Senan, "An analysis of the KDD99 and UNSW-NB15 datasets for the intrusion detection system," *Symmetry*, vol. 12, no. 10, p. 1666, 2020.

[31] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," *Ieee Access*, vol. 8, pp. 165130-165150, 2020.

[32] J. M. Peterson, J. L. Leevy, and T. M. Khoshgoftaar, "A review and analysis of the bot-iot dataset," *2021 IEEE International Conference on Service-Oriented System Engineering (SOSE). IEEE*, 2021.

[33] J. L. Leevy, and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data," *Journal of Big Data*, vol. 7, no. 1, pp. 1-19, 2020.