IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 1, 2022, accepted May 2, 2022, date of publication May 9, 2022, date of current version May 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3173319

Detection and Characterization of DDoS Attacks Using Time-Based Features

JAMES HALLADAY[1], DRAKE CULLEN[1], NATHAN BRINER[1], JACKSON WARREN[1], KARSON FYE[1], RAM BASNET[1], JEREMY BERGEN[1], AND TENZIN DOLECK[2]

[1]Department of Computer Science and Engineering, Colorado Mesa University (CMU), Grand Junction, CO 81501, USA
[2]Faculty of Education, Simon Fraser University, Burnaby, BC V5A 1S6, Canada

Corresponding author: Ram Basnet (rbasnet@coloradomesa.edu)

This work was supported by the State of Colorado through funds appropriated for Cybersecurity Law Dubbed "Cyber Coding Cryptology for State Records."

ABSTRACT In today's evolving cybersecurity landscape, distributed denial-of-service (DDoS) attacks have become one of the most prolific and costly threats. Their capability to incapacitate network services while causing millions of dollars in damages has made effective DDoS detection and prevention imperative for businesses and government entities alike. Prior research has found shallow and deep learning classifiers to be invaluable in detecting DDoS attacks; however, there is an absence of research concerning time-based features and classification among many DDoS attack types. In this article, we propose and study the efficacy of 25 time-based features to detect and classify 12 types of DDoS attacks using binary and multiclass classification. Furthermore, we ran experiments to compare the performance of eight traditional machine learning classifiers and one deep learning classifier using two different scenarios. Our findings show that the majority of models provided ∼99% accuracy on both the control and time-based experiments in detecting DDoS attacks while yielding ∼70% accuracy in classifying specific DDoS attack types. Training on the proposed time-based feature subset was found to be effective at reducing training time without compromising test accuracy; thus, the smaller time-based feature subset alone is beneficial for near-real time applications that incorporate continuous learning.

INDEX TERMS Time-based features, distributed denial of service attacks, machine learning, deep learning, multiclass, CICDDoS2019.

I. INTRODUCTION

Arbor Networks, a software company that supplies network security software to many of the world's largest internet service providers (ISPs), reports that more than 1,000 large distributed denial-of-service (DDoS) attacks are detected by their software every day [1]. These attacks vary from targeting personal computers to the ISPs that route network traffic. DDoS attacks are of growing concern because they are relatively easy to execute and difficult to defend against.

DDoS attacks are a type of denial-of-service attack where the attacker seeks to halt the operation of a network and to deny legitimate users access. As depicted in Figure 1, they are performed by using a network of infected computers—called a botnet—to bombard service providers with an overwhelming number of requests. The botnet is controlled and given instructions by a subset of the infected machines called "master machines" [1]. Mohammed *et al.* reported that the number of DDoS attacks has grown by 200% every year, causing losses of up to $100,000 an hour for the

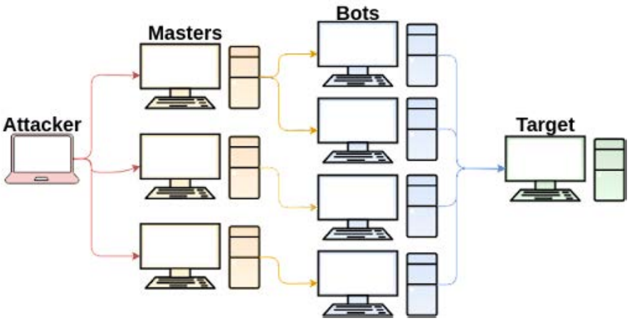


FIGURE 1. A DDoS attack hierarchy consisting of bots controlled by master machines under an attacker's control.

The associate editor coordinating the review of this manuscript and approving it for publication was Liangxiu Han.

targeted service providers [2]. In 2016, Dyn, a major address resolution service was brought offline by a record-breaking DDoS attack with a magnitude of 1.2 Tbps [1]. Dyn provides service to over 3,500 enterprises including Netflix, Twitter, and LinkedIn. This attack shows that DDoS attacks impose a significant threat to the internet as a whole. Moreover, with the proliferation of the Internet of Things (IoT), DDoS attacks have become more commonplace. This is due to IoT devices being widespread and many lacking proper security.

DDoS attacks can be mitigated by using traffic classification to detect and characterize malicious network traffic. Applications using the internet produce chains of packets that are mixed together as the signal travels from the original device to its destination. These signals have properties that can be measured within the context of some predefined time interval to produce flows. This time interval is called the flow interval. Callado *et al.* [3] define a flow as "a set of packets that share origin and destination addresses, origin and destination ports, transport protocol and are observed within a set timeframe." Traffic flows do not contain the actual data being transmitted in the packets but merely the measurement of the packets' meta-data, frequency, and direction (to or from the source). After enough flows are collected, machine learning techniques can be applied to classify malicious DDoS traffic from legitimate traffic.

Once the network traffic is characterized, network operators can employ strategies to mitigate the attack. Furthermore, as the techniques to detect DDoS attacks become more sophisticated, they can identify specific applications under attack. This granular application-specific classification can enable operators to tailor their mitigation strategies to more effectively eliminate the malicious traffic while providing a better experience for the legitimate network and application users.

Various research concerning machine and deep learning have suggested the benefits of smaller feature sets namely in reducing dimensionality, training time, and noise [4]. In the domain of DDoS detection, reduced feature sets have seen experimentation; yet, there are numerous unexplored feature groups to consider for model optimization. To that end, our work addresses the knowledge gap by focusing on detecting and classifying DDoS attacks over network traffic by investigating time-related features in traffic flows. First, we detect generic DDoS attacks, then we classify the attacks as one of twelve common DDoS attack types. Legitimate traffic and DDoS traffic have different signatures in traffic flows that can be exploited to train machine learning classifiers to recognize these patterns. Furthermore, various statistics gathered in traffic flows must depend on the frequency of packets in the flow, primarily containing metadata such as time, average flow, direction, etc. in the flow. Since DDoS traffic will have different signatures compared to legitimate traffic concerning time-related meta-data in the traffic flows, it follows that we can use exclusively time-based features to effectively classify and characterize DDoS traffic.

Lashkari *et al.* [5] define time-based features as the set of features concerning time-related statistics in a traffic flow. They have used time-based features to classify standard web traffic originating from The Onion Router (Tor) network and identify the application type that the Tor traffic originated from. The ISCXTor2016 dataset used by Lashkari *et al.* [5] and the CICDDoS2019 dataset used in our experiments [6] share the same overall set of features [7]. This is due to the fact that both datasets were generated using CICFlowMeter, formerly known as ISCXFlowMeter.

The novelty of our work is to show that the subset of time-based features alone can successfully be applied to the domain of DDoS attack detection and characterization, as opposed to existing literature which select features through feature selection algorithms or use the entire feature set. Furthermore, we show that the training and testing times of most classifiers are significantly reduced when the dimensionality of the dataset is reduced from the 70 baseline features to the 25 time-based features. As the DDoS attack landscape is constantly shifting, the ability to train a model continuously can better protect organizations from evolving threats. Moreover, differentiating a large number of attack types can help better form appropriate mitigation strategies.

Our Contributions:

- We train nine classifiers on a time-based feature subset to detect and classify DDoS attacks by analyzing temporally related features in traffic flows. To our knowledge, the time-based feature set has not been investigated in the domain of DDoS attacks.
- The smaller time-based feature set reduces overall training time, decreases dimensionality and noise, reduces the necessary computational resources, and promotes the viability of continuous learning.
- We've found few related works concerning multiclass classification in the domain of DDoS detection. One such study by Chen *et al.* focused on differentiating five types of DDoS attacks whereas our classifiers differentiate 12 different attack types.
- Our binary classifiers produced comparable or higher accuracy results than the ones published in existing literature, and our multiclass classifiers' performances are in line with Chen *et al.*'s classifiers despite using seven more attack types.
- Lashkari *et al.* found the time-based feature set to be effective in classifying Tor and VPN traffic. We have filled in the knowledge gap in the domain of DDoS attacks by finding the time-based feature subset alone to be comparably effective and providing much better training time.

The rest of the paper proceeds as follows: Section 2 reviews related works, while Section 3 outlines the dataset processing and experiment structure. In section 4, the nine classifiers are explained. Section 5 reviews our experiments and performance metrics in-depth, and Section 6 presents the experimental results. Our results are further discussed in

Section 7. Section 8 explores avenues for future work, and the paper concludes in Section 9.

II. RELATED WORKS

Elsayed *et al.* [8] proposed DDoSNet, a DDoS detection system for software-defined networking environments trained on the CICDDoS2019 dataset. DDoSNet utilized a recurrent neural network (RNN) which is specialized in sequential data. This RNN was paired with an autoencoder, an artificial neural network that encodes and decodes input data aiding noise reduction and anomaly detection. DDoSNet had a 99% accuracy with an AUC of 98.8%, substantially outperforming the classical machine learning models it was compared against.

Cil *et al.* [9] researched the effectiveness of feedforward neural networks (FNNs) for DDoS detection on the CICDDoS2019 dataset. Their model contained an input layer, three hidden layers, and a two-unit output layer. The FNN used the sigmoid activation function for the input and hidden layers and the softmax function for the output layer. Their model was trained and tested on two datasets, one which differentiated normal and attack traffic and another which differentiated the traffic as reflection or exploitation based. The FNN model accurately detected DDoS attacks with 99.9% accuracy and successfully classified the attack type with approximately 95% accuracy.

Salahuddin *et al.* [10] trained an anomaly detection system using time-based features and an autoencoder. They defined time-based features as "statistical information of a subset of packets collected over a specific period or time-window." Note that the time-based features used in our work are defined as temporally related statistics of traffic flows. For clarity, our definition of time-based features varies from Salahuddin's because we look at the statistics of a traffic flow with a static time interval. The researchers trained an autoencoder NN on 20 time-based statistical features generated using TShark and Kitsune [11] to examine the impact of time windows on anomaly detection. The model performed exceptionally well at detecting various types of DDoS attacks achieving precision, recall, and F1 score of over 99% for most attacks with aggregated time windows of 10 seconds and 10 milliseconds. Moreover, they found that when compared to an autoencoder trained on flow-based features, the time window-based model greatly outperformed it in detecting NetBIOS, PortMap, TCP, SYN, and UDPLag attacks. In the study, the researchers claimed that flow-based and packet-based features fail to provide models with adequate anomaly detection. However, this claim is inconsistent with many of the findings in table 1 as well our work due to the limited number of classifiers represented in their study.

Chen *et al.* [12] developed a DDoS detection model, DAD-MCNN, using a multi-channel convolutional neural network (MCNN), incremental training, and unique preprocessing methods. In their experiment, the researchers split data into feature groups as input for their MCNN. Their feature groups included traffic features, packet features, and host features. To train the MCNN, an incremental training method was deployed. This entailed training the model with a single convolutional layer for each channel, adding a new convolutional layer and training again until a suitable model was created. DAD-MCNN had higher accuracies than other deep and traditional machine learning models it was compared against, namely CNN, LSTM, SVM, KNN, and RF.

Elejla *et al.* [14] compared five machine learning classifiers (DT, SVM, NB, KNN, and a Multilayer Perceptron NN) for DDoS detection to determine which are the most efficient and effective at detecting popular IPv6 (Internet Protocol version six) DDoS attacks. The dataset chosen to train the models was a labeled flow-based IPv6 dataset with a variety of different ICMPv6 DDoS attacks [15]. DT and KNN had the highest evaluation metrics with accuracy, precision, recall, F1 score, and true positive rates above 85%.

Fouladi *et al.* [16] trained a convolutional neural network to detect DDoS attacks on Software-Defined Networks (SDNs). Training a deep neural network on raw features is computationally expensive, so they perform feature extraction using a continuous wavelet transform (CWT). Fouladi *et al.* found that the frequency of unique source/destination IPs in relation to the total number of packets varies over time, and CWT can be applied to these features to get two separate two-dimensional features that are passed to the CNN. After detecting an attack, their model restricts forwarded traffic to the target destination IP. The model outperforms previous schemes with an accuracy of 99%.

The majority of the research described above focuses on classifier comparisons for DDoS detection and classification. Only one of the discussed studies, Chen *et al.*, utilized multiclass classification to classify five types of DDoS attacks. In our study, we compare a relatively larger number of classifiers (9) and use multiclass classification with substantially more DDoS attack types (12) than prior works. Our binary classifiers perform on par with previous research, while our multiclass classifiers achieved substantially better results given the large number of attack types. Moreover, we add to the literature on the effectiveness of small feature sets, specifically in the under-researched time-based feature set. Reduced feature sets not only lower the potential for overfitting, but can lessen the computational burden as well as the training and testing time required for continuous training and testing. The frequent adjustments of models could improve the effectiveness and adaptability of DDoS detection schemes.

III. DATASET

A. DATASET OVERVIEW

The CICDDoS2019 dataset, provided by the Canadian Institute of Cybersecurity, contains benign and DDoS attack flows from 13 different modern DDoS attack types. Approximately 112,000 of the 70 million samples are benign [6]. Since there were less than a thousand samples of the WebDDoS attack type, which was substantially smaller compared to

TABLE 1. A comparison of DDoS traffic detection works.

Paper	Dataset	Classifier(s)	Features	Multiclass	Results
Cil et al. [9]	CICDDoS2019	FNN (3 hidden, 128 units)	86	No	~99.9% acc dataset 1 ~95% acc. dataset 2
Elejla et al. [14]	Generated Flow-based Datasets	DT, SVM, NB, KNN, MLP	11	No	85% acc.
Elsayed et al. [8]	CICDDoS2019	RNN-Autoencoder	77	No	~99% acc.
Fouladi et al. [16]	MAWI dataset and Ubuntu Docker	CNN	N/A Extracted with CWT	No	~99.9% acc.
Hussain [17]	CICDDoS2019	BN, Bagging, KNN, SMO	24	No	Acc. N/A F1: >92%
Maranhao et al. [18]	CICDDoS2019 & NSL-KDD	ADA, LDA, LR, RF	64	No	98.78% acc. for RF
Salahuddin et al. [10]	CICDDoS2019	Autoencoder	20	No	99% acc.
Shieh et al. [19]	CICIDS2017 & CICDDoS2019	GMM, Bidirectional LSTM	80	No	94% acc.
Sindian et al. [20]	CICDDoS2019	EDSA, DNN using Autoencoder	80	No	~96.3% acc.
Chen et al. [12]	CICIDS2017 KDDCUP99	MCNN	75	Yes (5)	99.18% acc. binary 67.96% avg acc. multiclass
This Paper	CICDDoS2019	LD, KNN, RF, XGB, LGBM (Top 5)	70, 25	Yes (12)	Control: >99.9% acc. binary >74% acc. multiclass Time-based: >98.5% acc. binary >69% acc. multiclass

the other attack types, we ignored this attack type in our experiments. Our work utilizes the CICDDoS2019 dataset because it contains a high number of attack types and samples which are vital for multiclass classification. Moreover, the dataset was created by authors from a reputable institution and many related works performed experimentation on the CICDDoS2019; thus, our results are more comparable to the related works.

The dataset is published in both raw pcap files and tabular data that is preprocessed by CIC-FlowMeter. Sun *et al.* [22] note that data can be split into the categories of structured and unstructured data where tabular data is a subset of structured data. Deep learning methods such as CNNs and RNNs generally outperform other algorithms when classifying unstructured data. However, when dealing with structured data, shallow machine learning algorithms such as k-Nearest Neighbors, Support Vector Machines, and Random Forests tend to perform better. In particular, XGBoost and Light Gradient Boosting Machines are boosting algorithms that have performed extremely well in data science competitions, consistently outperforming other methods of classification for structured data [22].

B. DATA PREPROCESSING

Before training the classifiers, the data was cleaned, pruned, encoded, and normalized. This involved removing 18 of the 88 features. Nine features containing only zero values and four of the identifier features that would overfit to the classifiers, namely source and destination ports and source and destination IP addresses, were removed from the training data. The other features that were removed were either duplicates or unlabeled features. Next, samples with missing or infinite values were eliminated. To avoid incorporating unintended bias into the dataset, the normalization was conducted after the data was split into training and testing subsets.

Since the dataset has over 70 million samples, downsampling was performed in order to maintain reasonable training times and overcome the memory limitation on a typical desktop computer. Furthermore, downsampling was done to balance the highly imbalanced dataset consisting of less than 1% benign samples. A classifier could misclassify every sample as DDoS and still achieve over 99% accuracy on the imbalanced dataset. To balance the dataset, the binary classification (Scenario A) data was composed of 200K samples with a 50/50 split between benign and malicious samples. The malicious samples were composed of an equal proportion of all 12 DDoS attack types considered selected at random. The multiclass classification (Scenario B) data incorporated approximately 160K samples per attack type. The pruning, downsampling, and balancing of the dataset made it easier to conduct our experiments as well as improved the classifiers' performances.

C. TIME-BASED FEATURES

Our time-based dataset consists of 25 features. Table 2 lists the feature names and their descriptions used in our time-based dataset. These features were aggregated by Lashkari *et al.* [5] for their research, and have proven to be effective in detecting VPN and Tor traffic. Due to the fact

TABLE 2. Description of time-based features. Features with an * include the mean, min, max, and standard deviation of that feature.

Feature	Description
fiat*	Forward Inter Arrival Time, the time between two packets sent in the forward direction.
biat*	Backward Inter Arrival Time, the time between two packets sent backwards.
flowiat*	Flow Inter Arrival Time, the time between ~two packets sent in either direction.
active*	The amount of time a flow was active before ~going idle.
Idle*	The amount of time a flow was idle before ~becoming active.
fb_psec	Flow bytes per second.
fp_psec	Flow packets per second.
fwd IAT total	Forward Inter Arrival Time Total.
bwd IAT total	Backward Inter Arrival Time Total.
duration	Duration of a flow.

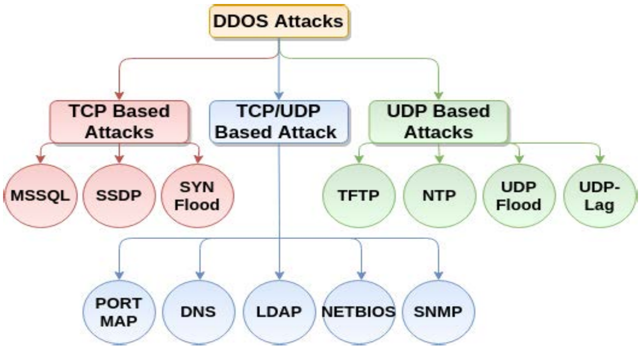


FIGURE 2. The 12 DDoS attacks categorized by TCP/UDP protocol.

that DDoS attacks involve sending large numbers of packets over a short period of time, we hypothesize that these time-based features would effectively translate to DDoS detection problems. Each of the first five features marked with an * described in the table is further broken down into four separate statistical features: mean, min, max, and standard deviation.

D. ATTACK CLASSIFICATION

As shown in Figure 2, the 12 DDoS attacks represented in the CICDDoS2019 dataset are MSSQL, SSDP, SYN Flood, PORTMAP, DNS, LDAP, NETBIOS, SNMP, TFTP, NTP, UDP Flood, and UDP-Lag. Each attack can be categorized as a Transmission Control Protocol (TCP) based attack, a User Datagram Protocol (UDP) based attack, or a TCP/UDP attack if it can be carried out using either protocol [6]. TCP is a connection-oriented protocol; therefore, a connection must be established between devices before data can be transmitted. After data is done transmitting, the connection must be closed. TCP is more reliable than UDP because it provides acknowledgment of data, packets arrive in order, and lost packets can be retransmitted [21]. On the other hand, UDP doesn't open or maintain a reliable connection. Though only basic error checking is implemented and packets can be lost, UDP is much faster than TCP [23]. The unique attributes of the protocols prove beneficial when classifying DDoS attacks.

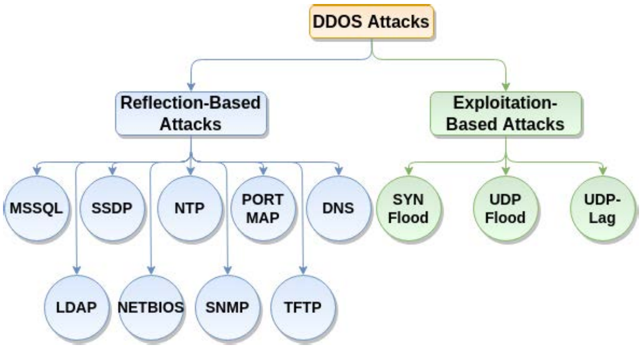


FIGURE 3. The 12 DDoS attacks are reflection or exploitation-based.

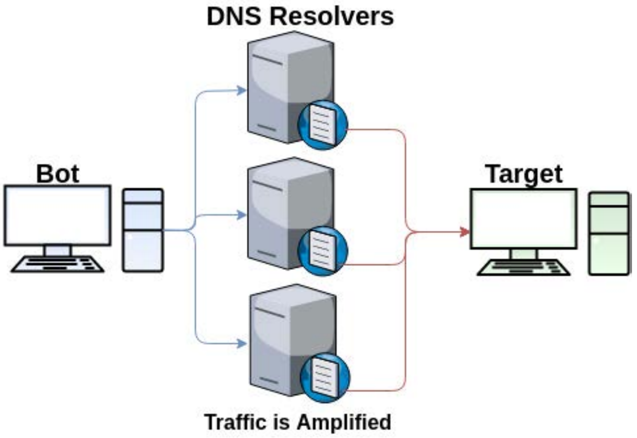


FIGURE 4. A DNS amplification attack.

The DDoS attacks are also classified as reflection-based and exploitation-based attacks. The categorization of these attack types is depicted in Figure 3. In a reflection-based attack, an attacker (or one of their bots) sends requests for information to multiple servers and systems. When sending this request, the attackers spoof their IP address so that it looks as if the request is coming from the machine that they are targeting. The server sends responses to the target overwhelming it with the bogus traffic and effectively victimizing it with the DDoS attack. Oftentimes, a reflection-based attack will be used in conjunction with an amplification attack. For example, as shown in Figure 4, DNS amplification attacks target open Domain Name System (DNS) Resolvers. Bots in a botnet will query DNS resolvers with arguments such as "ANY." The DNS Resolvers will respond to the spoofed address with a large response, essentially increasing the amount of traffic they can direct towards a victim [24].

The dataset contains three types of exploitation-based attacks. UDP Flood attacks occur when large amounts of UDP packets are sent to random ports. This can overwhelm the available bandwidth and reduce performance or cause the system to crash. SYN Flood attacks exploit the TCP-three-way handshake protocol where attackers keep sending SYN packets to the target machine without sending the SYN-ACK packet required to establish a reliable connection. Finally, UDP-Lag attacks use a hardware switch or run software that takes up the bandwidth of other users so that the connection between the server and client is disrupted.

IV. CLASSIFIERS

For our experiments, we chose classifiers proven to be effective at classifying structured data. These classifiers include boosting algorithms such as LightGBM (LGBM), XGBoost (XGB), and Adaptive Boosting (ADA) since they have been shown to outperform other models in competition [22]. We also chose to use a DNN so we could compare the results of traditional machine learning algorithms, deep learning algorithms, and boosting algorithms. Since tabular data doesn't lend well to classification by RNNs or CNNs [22], we decided to leave these out of our investigation. The following sections provide a high-level brief description of each classifier. Readers are encouraged to follow the citations for the technical detail which is beyond the scope of this work.

A. RANDOM FOREST

RF is an ensemble method made up of a collection of decision trees. Decision trees are classifiers that follow the divide-and-conquer approach. Data splits occur at every node and classification is completed at the leaves. RF utilizes bagging and randomness to aggregate the predictions of the decision trees and reduce the overall bias of a single tree [25].

B. K-NEAREST NEIGHBORS

KNN is a non-parametric supervised learning algorithm used for regression and classification problems. The algorithm selects a datapoint and finds the k nearest data entries called neighbors. The new data is classified based on the majority class of the neighbors. Generally, small k values will lead to an underfit model. Higher values will lower variance and increase bias and computing time. It's important that the optimal k value is picked for the best performance by the classifier [26].

C. LIGHTGBM

LGBM, created by Microsoft, is a type of gradient boosted decision tree. In general, decision trees identify splits that receive the greatest change in entropy or highest information gain before and after each split. A pre-sorted or histogram-based algorithm is used to calculate the best split, and this can take a great deal of time on large datasets. LGBM attempts to address this issue by using Exclusive Feature Bundling (EFB) and Gradient One-Side Sampling (GOSS). EFB combines mutually exclusive features into a single feature. GOSS randomly drops smaller gradients because larger gradients are usually associated with larger information gain. Combining these two methods reduces the time complexity [27].

D. XGBOOST

XGB is a gradient boosted decision trees algorithm that incorporates the Gradient Boosting Machine's (GBM) framework. XGB works to optimize the GBM framework by addressing missing data with sparsity awareness, preventing overfitting with LASSO and algorithmic enhancements, and finding optimal tree splits with the Weighted Quantile sketch algorithm. Furthermore, XGBoost incorporates depth-first tree pruning, out-of-core computing, and parallelized decision tree construction [28].

E. ADABOOST

ADA was the first practical boosting algorithm and is based on the idea of combining a lot of weak and inaccurate data to create an accurate prediction rule. It employs an ensemble method that assigns higher weights to incorrectly classified instances followed by boosting to reduce variance and bias. This process is performed by growing each learner from the prior learner [29].

F. LINEAR DISCRIMINANT ANALYSIS

LD is a linear classification technique that is typically considered better than logistic regression at multiclass classification. The algorithm uses a dimensionality reduction technique to take features from a higher dimension and represent them in a lower dimension. In doing so, it works to discover differences in groups. The separate groups in lower dimensions represent separate classes [30].

G. GAUSSIAN NAIVE BAYES

GNB is a probabilistic algorithm that supports continuous data and follows the Gaussian normal distribution. GNB is an extension of Naive Bayes. Naive Bayes is a classification algorithm that is built upon Bayes theorem. The classifier assumes that features are independent of one another and are distributed according to a normal distribution. The probability density function of a distribution is used to calculate results for probability likelihoods. GNB has the advantage of working well with small amounts of data [31].

H. SUPPORT VECTOR MACHINE

SVM is a classifier that attempts to find a hyperplane in an N-dimensional space that will separate data points belonging to different classes. There are many potential hyperplanes to choose from, but the SVM chooses the plane with the maximum margin. The maximum margin represents the greatest distance between data points of both classes. SVMs have high speeds and perform well on limited data [32].

I. DEEP NEURAL NETWORK

fast.ai is a deep learning library that uses PyTorch as a backend. The tabular_learner method was used to train the

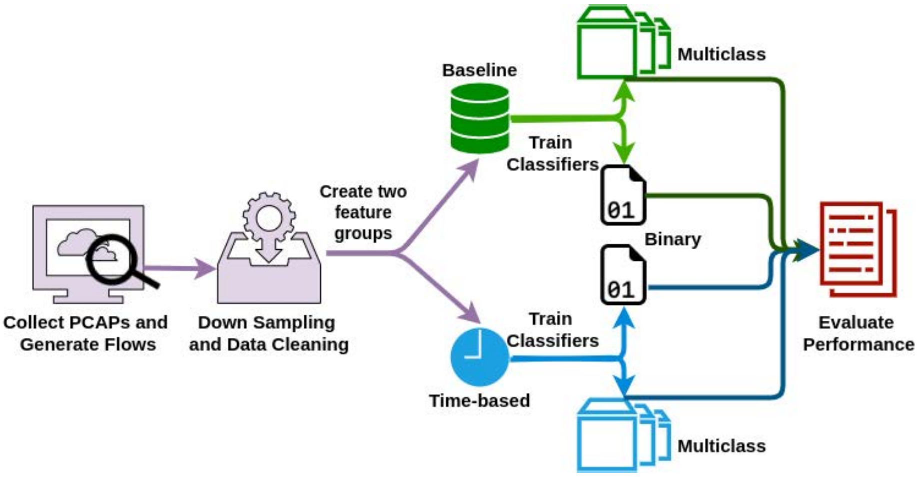


FIGURE 5. Experimental workflow.

model with 10 epochs. The rest of the hyperparameter tuning was handled by fast.ai. We trained a DNN model with 28 hidden layers each containing 50 nodes.

V. EXPERIMENTS

A. OVERVIEW

To study the efficacy of the proposed time-based features in detecting and classifying DDoS attacks, we followed the steps depicted in Figure 5. After cleaning and downsampling the original dataset, we generated a baseline and a time-based dataset for both Scenario A and Scenario B. Both datasets were then used to evaluate the performance of the time-based features.

Experiments were conducted on one machine to avoid discrepancies in model training times. The system used was Ubuntu 20.04.3 LTS with an Intel i7-7700k CPU and 16 Gigabytes of RAM. We used the Jupyter Notebook environment and the source code is published on GitHub [33].

We employed an 80/20 split for all our experiments where 80% was allocated for training and 20% was used for testing the models. The performance metrics of the models were calculated using the test set. Each experiment randomly splits the data using the same seed; therefore, the experiments can be reproduced if necessary.

B. EXPERIMENT SCENARIOS

We conduct our experiments in two separate scenarios and compare the results of several classifiers on a time-based feature set. Scenario A aims to detect DDoS attacks. This is done using binary classification where benign flows are mixed with a basket of 12 different DDoS attack types sharing the same label as DDoS flows. Control results were established by training the nine models on all 70 features in the processed and cleaned dataset. Next, the same nine models were trained on the 25 time-based features, and performance metrics were recorded to determine whether time-based features can effectively classify traffic as benign or DDoS.

Scenario B focuses on characterizing a given DDoS attack using multiclass classification. The nine models were trained to differentiate between UDP-Lag, UDP Flood, TFTP, SYN Flood, SSDP, SNMP, Portmap, NetBIOS, NTP, MSSQL, LDAP, and DNS attacks. Equal proportions of each attack type were used when training the Scenario B classifiers. Note that Scenario B does not include benign traffic as Scenario A addresses benign vs DDoS detection. Like Scenario A, the first set of models were trained on every feature and subsequent models were trained on time-based features.

C. PERFORMANCE METRICS

We use five commonly used metrics (accuracy, precision, F1-score, area under the ROC curve (AUC), and training time) to compare the performances of our models. To calculate most of these metrics, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are needed.

In the case of a DDoS attack, it is reasonable to let through some malicious traffic (FN classification). The goal isn't to screen out all malicious traffic, but a high enough percentage to allow normal operation to continue. FPs that identify benign traffic as a DDoS attack could negatively impact user experience. For instance, screening users and forcing them to prove they aren't a bot can drive away web traffic.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100 \tag{1}$$

$$Precision = \frac{TP}{TP + FP} * 100 \tag{2}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} * 100 \tag{3}$$

Accuracy is a popular performance metric that can be applied to binary and multiclass classification problems. Although accuracy can give an initial insight into the performance of a model, there is a drawback. If the dataset

TABLE 3. Results for Scenario A control experiment.

Classifiers	Accuracy	F1	AUC	Training Time (s)
LD	.9999	1	1	1.33
LGBM	.9999	1	1	1.34
XGB	.9998	1	1	8.93
RF	.9997	1	1	12.03
DNN	.9989	1	1	185.61
ADA	.9987	1	1	12.91
SVM	.9985	1	1	203.28
KNN	.9951	1	1	30.36
GNB	.9862	.99	.99	.14

is imbalanced, the model may have high accuracy on training data and low accuracy in the real world.

Precision is the ratio of TPs over the total number of times that a positive was predicted; therefore, it determines the proportion of positive identifications that are correct. Precision is a useful metric when the cost of a FP is high. As mentioned above, FPs can negatively impact user experience when detecting DDoS attacks.

F1-score is calculated from the precision and recall metrics and is necessary when seeking a balance between precision and recall. Furthermore, F1-score works well on unbalanced datasets. It can be used as an alternative to accuracy.

ROC curves represent the TP rate (recall/sensitivity) measured against the FP rate (specificity). ROC and AUC demonstrate classification effectiveness regardless of imbalanced testing data.

Training time can be an invaluable metric when comparing models to each other. Furthermore, training time is helpful in quantifying the impact of feature selection [32]. Feature selection may potentially reduce accuracy; however, the minor decrease in accuracy could be tolerated with a significant decrease in training and testing for near-real time application of classifiers.

VI. RESULTS

In this section, we present the Scenario A and Scenario B results. First, the control results are discussed for Scenario A and Scenario B. Next, time-based results are presented and their performance metrics are compared to the control experimentation.

A. SCENARIO A CONTROL RESULTS

Every model in Scenario A performed exceptionally well on the control features. The models performed in the range of 98-99% accuracy and each model had an F1-score and AUC of at least 0.99.

Table 3 shows the models sorted in descending order by their accuracy. If two models had the same accuracy, the model with the lower training time is treated as the superior model. Based on accuracy and training time, the top five models for Scenario A with control features are LD, LGBM, XGB, RF, and ADA. Overall, GNB had the lowest accuracy and F1-score; however, its training time was much better than any other model.

TABLE 4. Results for Scenario B control experiment.

Classifiers	Accuracy	F1	AUC	Training Time (s)
XGB	0.7408	0.7342	0.98	1569.45
LGBM	0.7346	0.7257	0.97	71.96
LD	0.7334	0.7258	0.97	59.037
KNN	0.7038	0.7	0.97	27525.78
RF	0.6994	0.6858	0.97	218.77
DNN	0.6362	0.65	0.96	1566.47
SVM	0.6302	0.5958	0.96	189733.14
ADA	0.5149	0.4525	0.86	125.01
GNB	0.3849	0.29	0.92	2.94

B. SCENARIO B CONTROL RESULTS

In contrast to the Scenario A control experiments where every model performed well, the multiclass experiments saw a great divergence in model performance. Referring to table 4 (sorted in descending order by accuracies), XGB, LD, and LGBM achieved the highest accuracies and F1 scores of just over 70% and 0.7, respectively. For differentiating among 12 attack types, these are exceptional results. RF, KNN, and DNN all performed reasonably well, maintaining accuracies between 60-70%, though KNN and DNN suffered from long training times. The other models, SVM, ADA, and GNB performed poorly; ADA and GNB provided comparably terrible accuracies, and SVM provided mediocre accuracy with the longest training time by a significant margin.

The confusion matrices for the Scenario B control experiments for the top 5 performing classifiers are depicted in Figure 6. Some notable misclassification trends amongst the classifiers include misclassifying UDP as SSDP, UDP Lag as SYN, LDAP as DNS, and DNS as LDAP. These classification trends don't reflect overall poor performance, as each attack type was correctly classified the majority of the time.

C. SCENARIO A TIME-BASED RESULTS

Transitioning to the time-based data as seen in Table 5, most classifiers performed similar to their control experiment yielding only a small decrease in accuracy. GNB, DNN, and SVM classifiers saw notable deterioration in accuracy and F1 whereas the accuracy drop for GNB was an astonishing 50%. The time-based experiments improved on training time compared to the control experiments; however, all Scenario A experiments trained quickly. The training time difference between the time-based and control experiments becomes more apparent in the Scenario B experiments. This could result from the fact that differentiating among multiple classes is a more computationally demanding task compared to binary classification; therefore, any improvements in training speed will be compounded in the multiclass classification.

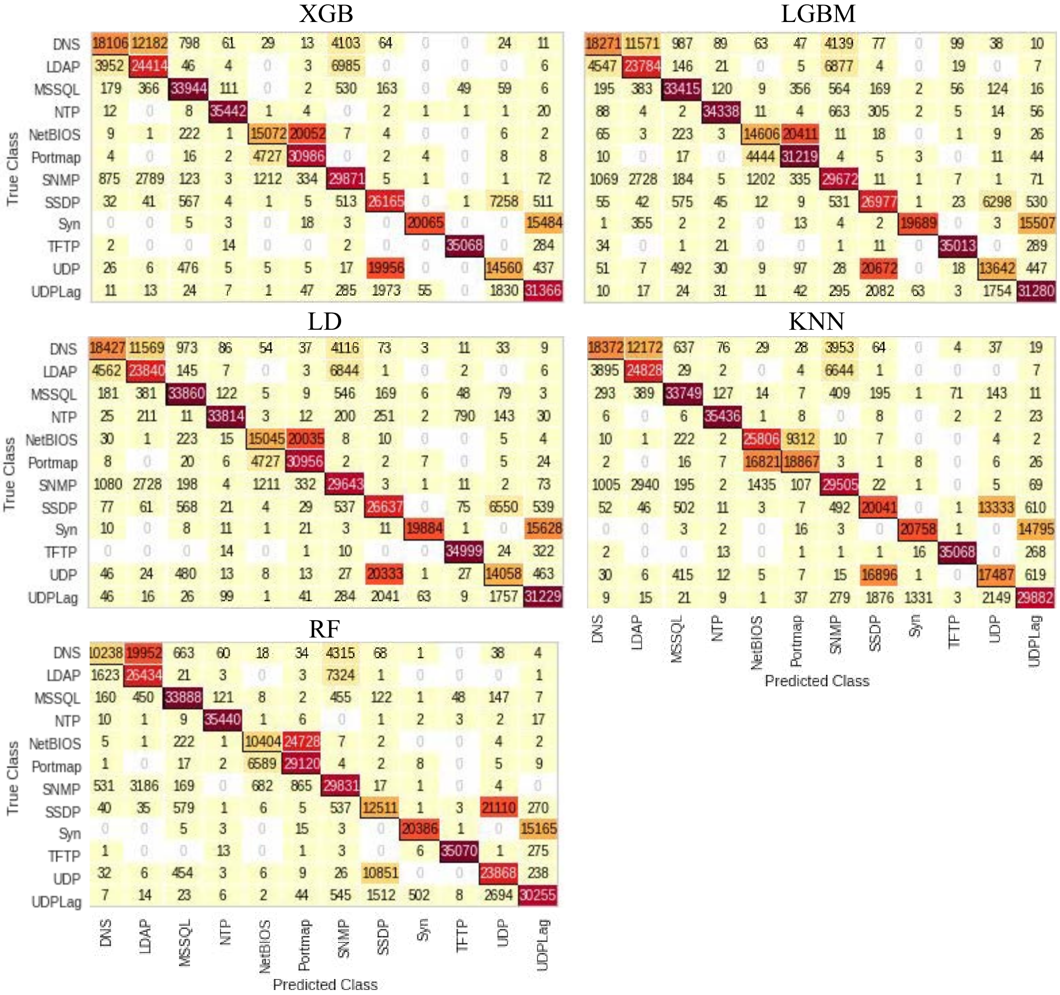


FIGURE 6. Confusion Matrices for Scenario B control experimentation.

TABLE 5. Results for Scenario A time-based experimentation.

Classifiers	Accuracy	F1	AUC	Training Time (s)
XGB	0.9858	0.99	1	5.69
RF	0.9858	0.99	1	13.06
LGBM	0.9844	0.98	1	0.65
LD	0.9843	0.98	1	0.65
KNN	0.9792	0.98	0.99	28.69
ADA	0.9578	0.96	0.99	6.27
SVM	0.9051	0.905	0.96	1728.08
DNN	0.698	0.69	0.94	187.34
GNB	0.5784	0.5	0.85	0.05

TABLE 6. Results for Scenario B time-based experimentation.

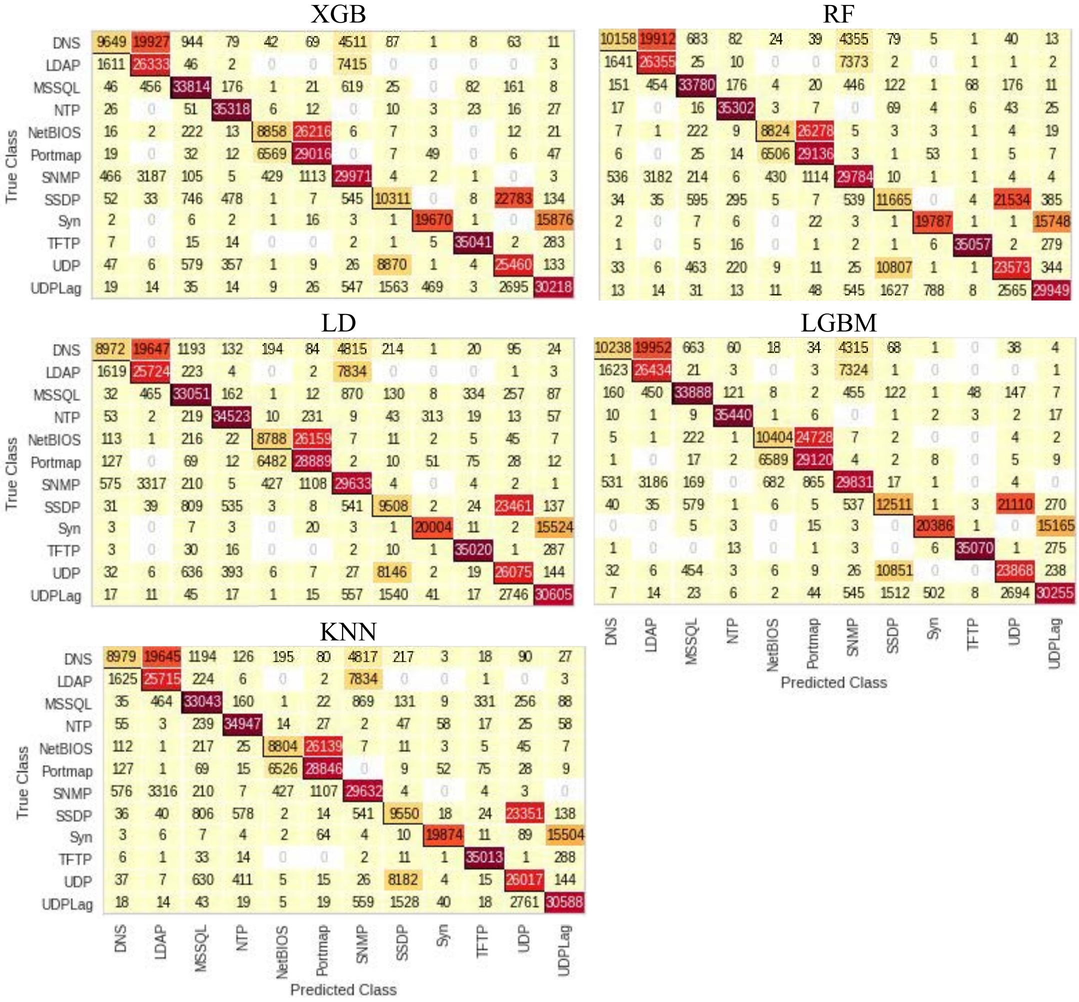
Classifiers	Accuracy	F1	AUC	Training Time (s)
XGB	0.6905	0.6733	0.97	624.01
RF	0.6898	0.6742	0.97	229.31
LD	0.6843	0.6642	0.97	44.11
LGBM	0.6838	0.6642	0.97	56.38
KNN	0.6743	0.67	0.96	13696.92
SVM	0.5427	0.495	0.88	209182.82
ADA	0.5151	0.4483	0.87	71.8
DNN	0.4955	0.48	0.87	1401.08
GNB	0.1139	0.0658	0.66	0.062

D. SCENARIO B TIME-BASED RESULTS

Observing the time-based Scenario B results seen in Table 6, accuracy and F1-score saw a slight degradation across all classifiers. Training times saw moderate to substantial improvements across all classifiers except for RF which maintained a similar time as in the control experiment. The confusion matrices in Figure 7 displayed identical trends as in the Scenario B control experiment.

VII. DISCUSSION

Figures 8 and 9 compare the individual classifier accuracies for both control and time-based experiments.



XGB

	DNS	LDAP	MSSQL	NTP	NetBIOS	Portmap	SNMP	SSDP	Syn	TFTP	UDP	UDPLag
DNS	9649	19927	944	79	42	69	4511	87	1	8	63	11
LDAP	1611	26333	46	2	0	0	7415	0	0	0	0	3
MSSQL	46	456	33814	176	1	21	619	25	0	82	161	8
NTP	26	0	51	35318	6	12	0	10	3	23	16	27
NetBIOS	16	2	222	13	8858	26216	6	7	3	0	12	21
Portmap	19	0	32	12	6569	29016	0	7	49	0	6	47
SNMP	466	3187	105	5	429	1113	29971	4	2	1	0	3
SSDP	52	33	746	478	1	7	545	10311	0	8	22783	134
Syn	2	0	6	2	1	16	3	1	19670	1	0	15876
TFTP	7	0	15	14	0	0	2	1	5	35041	2	283
UDP	47	6	579	357	1	9	26	8870	1	4	25460	133
UDPLag	19	14	35	14	9	26	547	1563	469	3	2695	30218

RF

	DNS	LDAP	MSSQL	NTP	NetBIOS	Portmap	SNMP	SSDP	Syn	TFTP	UDP	UDPLag
DNS	10158	19912	683	82	24	39	4355	79	5	1	40	13
LDAP	1641	26355	25	10	0	0	7373	2	0	1	1	2
MSSQL	151	454	33780	176	4	20	446	122	1	68	176	11
NTP	17	0	16	35302	3	7	0	69	4	6	43	25
NetBIOS	7	1	222	9	8824	26278	5	3	3	1	4	19
Portmap	6	0	25	14	6506	29136	3	1	53	1	5	7
SNMP	536	3182	214	6	430	1114	29784	10	1	1	4	4
SSDP	34	35	595	295	5	7	539	11665	0	4	21534	385
Syn	2	0	7	6	0	22	3	1	19787	1	1	15748
TFTP	1	0	5	16	0	1	2	1	6	35057	2	279
UDP	33	6	463	220	9	11	25	10807	1	1	23573	344
UDPLag	13	14	31	13	11	48	545	1627	788	8	2565	29949

LD

	DNS	LDAP	MSSQL	NTP	NetBIOS	Portmap	SNMP	SSDP	Syn	TFTP	UDP	UDPLag
DNS	8972	19647	1193	132	194	84	4815	214	1	20	95	24
LDAP	1619	25724	223	4	0	2	7834	0	0	0	1	3
MSSQL	32	465	33051	162	1	12	870	130	8	334	257	87
NTP	53	2	219	34523	10	231	9	43	313	19	13	57
NetBIOS	113	1	216	22	8788	26159	7	11	2	5	45	7
Portmap	127	0	69	12	6482	28889	2	10	51	75	28	12
SNMP	575	3317	210	5	427	1108	29633	4	0	4	2	1
SSDP	31	39	809	535	3	8	541	9508	2	24	23461	137
Syn	3	0	7	3	0	20	3	1	20004	11	2	15524
TFTP	3	0	30	16	0	0	2	10	1	35020	1	287
UDP	32	6	636	393	6	7	27	8146	2	19	26075	144
UDPLag	17	11	45	17	1	15	557	1540	41	17	2746	30605

LGBM

	DNS	LDAP	MSSQL	NTP	NetBIOS	Portmap	SNMP	SSDP	Syn	TFTP	UDP	UDPLag
DNS	10238	19952	663	60	18	34	4315	68	1	0	38	4
LDAP	1623	26434	21	3	0	3	7324	1	0	0	0	1
MSSQL	160	450	33888	121	8	2	455	122	1	48	147	7
NTP	10	1	9	35440	1	6	0	1	2	3	2	17
NetBIOS	5	1	222	1	10404	24728	7	2	0	0	4	2
Portmap	1	0	17	2	6589	29120	4	2	8	0	5	9
SNMP	531	3186	169	0	682	865	29831	17	1	0	4	0
SSDP	40	35	579	1	6	5	537	12511	1	3	21110	270
Syn	0	0	5	3	0	15	3	0	20386	1	0	15165
TFTP	1	0	0	13	0	1	3	0	6	35070	1	275
UDP	32	6	454	3	6	9	26	10851	0	0	23868	238
UDPLag	7	14	23	6	2	44	545	1512	502	8	2694	30255

KNN

	DNS	LDAP	MSSQL	NTP	NetBIOS	Portmap	SNMP	SSDP	Syn	TFTP	UDP	UDPLag
DNS	8979	19645	1194	126	195	80	4817	217	3	18	90	27
LDAP	1625	25715	224	6	0	2	7834	0	0	1	0	3
MSSQL	35	464	33043	160	1	22	869	131	9	331	256	88
NTP	55	3	239	34947	14	27	2	47	58	17	25	58
NetBIOS	112	1	217	25	8804	26139	7	11	3	5	45	7
Portmap	127	1	69	15	6526	28846	0	9	52	75	28	9
SNMP	576	3316	210	7	427	1107	29632	4	0	4	3	0
SSDP	36	40	806	578	2	14	541	9550	18	24	23351	138
Syn	3	6	7	4	2	64	4	10	19874	11	89	15504
TFTP	6	1	33	14	0	0	2	11	1	35013	1	288
UDP	37	7	630	411	5	15	26	8182	4	15	26017	144
UDPLag	18	14	43	19	5	19	559	1528	40	18	2761	30588

FIGURE 7. Confusion matrices for Scenario B time-based experimentation.

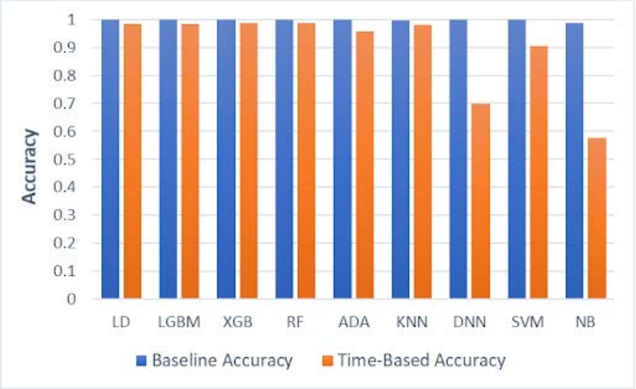


FIGURE 8. Comparison of models' accuracies in Scenario A.

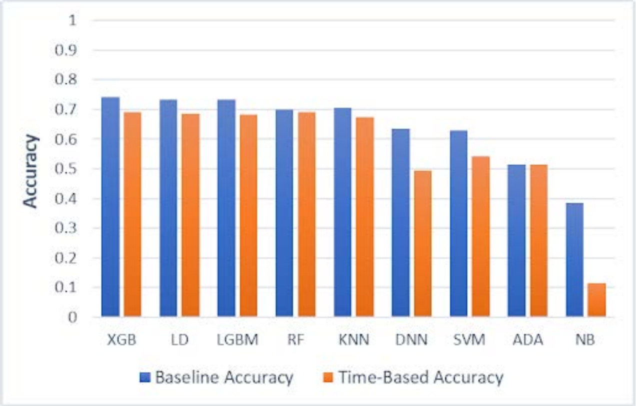


FIGURE 9. Comparison of models' accuracies in Scenario B.

It can be observed from these figures that the time-based accuracies for each classifier are less than that of the control accuracies to varying degrees. Notably, GNB and DNN saw a considerable decrease in accuracy when trained on the time-based dataset. The other models saw a marginal decrease in accuracy. The median accuracy decrease was 1.62% for Scenario A and 7.43% for Scenario B. Median is a preferable metric for model comparison as it is less susceptible to the influence of outliers (such as GNB).

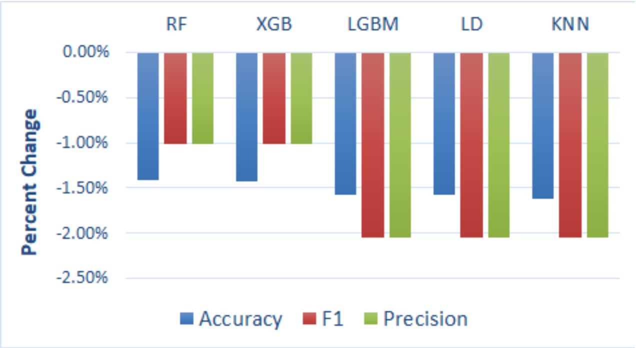


FIGURE 10. The percent change in accuracy, F1-score, and precision for models in Scenario A when trained on time-based features.

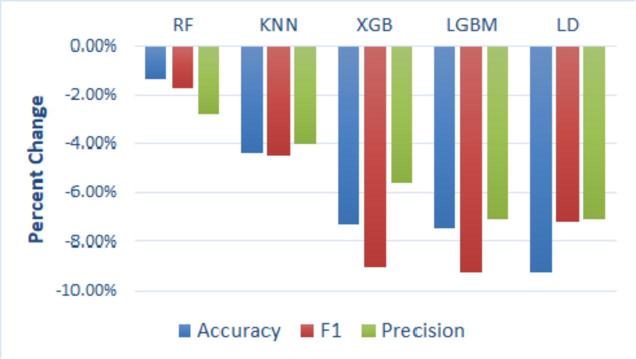


FIGURE 11. The percent change in accuracy, F1-score, and precision for models in Scenario B when trained on time-based features.

Figures 10 and 11 plot the individual performance change for the top 5 performing models. It can be observed from Figure 10 that XGB and RF saw the least accuracy, F1-score, and precision degradation in Scenario A. In Scenario B, RF and KNN saw the least accuracy degradation whereas XGB, LGBM, and LD saw greater accuracy degradation.

Between the time-based and control experiments, we observed dramatic improvements in training times across all models with the exception of RF and SVM. RF trained at approximately the same speed on both feature sets for both scenarios; whereas SVM saw a significant increase in training time in Scenario A from baseline to time-based experiments. The median training time decrease for Scenario A when training on the time-based feature set was 36.28% and 25.29% for Scenario B.

Table 7 compares the change in training for the top 5 performing classifiers. Notably, RF was an anomaly and took longer to train with the reduced feature set. In Scenario A, LGBM and LD trained in half the time. In Scenario B, XGB and KNN saw the greatest decrease in training time of 60.24% and 50.24% respectively.

From Figure 6 and 7, it can be observed that using only the time-based features did not introduce any new misclassification trends. Similar to the control results, the top five models trained on time-based features tended to misclassify UDP as SSDP, UDP Lag as SYN, LDAP as DNS, and DNS as LDAP. Overall, the models still performed with high accuracy and significantly lower training times when trained on the time-based features.

When training a model to detect DDoS attacks (Scenario A), LD, LGBM, and XGB benefit tremendously from training on time-based features. By training these models on time-based features, their training times are significantly reduced and they see a very small drop in accuracy. In contrast, RF doesn't warrant training on time-based features because the drop in accuracy doesn't justify a similar training time.

KNN sees the greatest improvement when trained on time-based features to distinguish among DDoS attack types (Scenario B). Furthermore, training XGB, LD, and LGBM on time-based features significantly decreases the overall training time, but the models are less accurate. Training RF on time-based features seems to be impractical due to lower accuracy and longer training time compared to the other models.

VIII. LIMITATIONS AND FUTURE WORK

In order to reduce computational resources, the dataset was significantly downsampled before training our classifiers. Deep learning models in particular benefit from large amounts of data; thus, training on a smaller dataset may have impacted the results [35]. With that said, our experiments used millions of samples and provided high accuracies regardless.

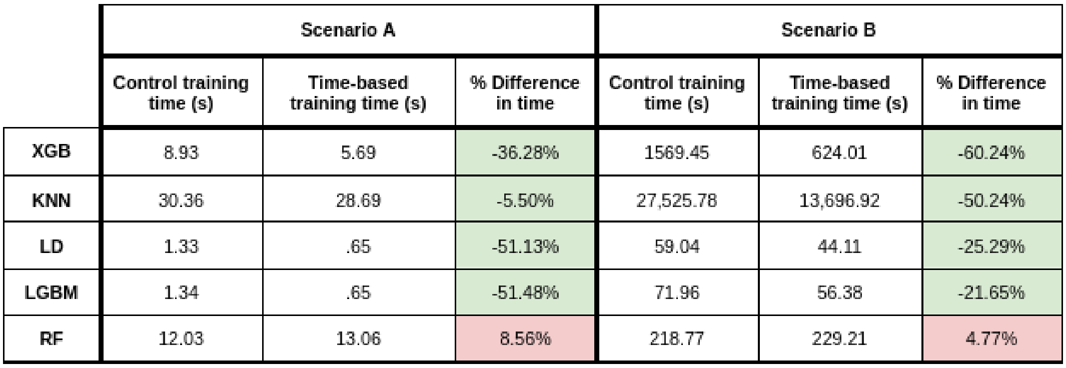
While the CICDDoS2019 dataset contains a large variety of DDoS attack types, there are still many more attacks left untested. Those untested DDoS attack types, such as low-rate DDoS attacks, may not be detectable by current classifiers. The authors of CICDDoS2019 [6] did not report the flow intervals used to generate the traffic flows. It's possible that more optimal flow intervals may exist that could better detect these attacks. Further experimentation could be conducted to investigate the optimal flow intervals to improve the performance of the time-based features by regenerating the tabular dataset from the raw pcap traffic data over different flow intervals using CIC-FlowMeter.

In our experimentation, only KNN utilized hyperparameter optimization. Future works could potentially refine the models observed in this study through greater hyperparameter tuning and applying grid search among all the classifiers.

One DNN model was trained and tested for this study using a standard configuration; however, there exist many varieties of deep learning classifiers that warrant experimentation in this domain and with the time-based featureset. Some of these classifiers, such as CNNs, benefit greatly from converting tabular data to image representations in order to find relationships among features [16]. Future works could investigate time-based features with these more advanced classifiers along with the data augmentation techniques.

The results of our experiments indicate that smaller feature groups are effective at improving model training speed with a minor decrease in accuracy. Feature engineering and

TABLE 7. A comparison of DDoS traffic detection works.



	Scenario A			Scenario B		
	Control training time (s)	Time-based training time (s)	% Difference in time	Control training time (s)	Time-based training time (s)	% Difference in time
XGB	8.93	5.69	-36.28%	1569.45	624.01	-60.24%
KNN	30.36	28.69	-5.50%	27,525.78	13,696.92	-50.24%
LD	1.33	.65	-51.13%	59.04	44.11	-25.29%
LGBM	1.34	.65	-51.48%	71.96	56.38	-21.65%
RF	12.03	13.06	8.56%	218.77	229.21	4.77%

experimentations with other feature groups could create a more robust dataset that may result in more effective models without performance degradation. Finally, the proposed time-based feature subset seems to be a promising avenue for lowering training times while achieving comparable accuracy perhaps in applications other than DDoS and Tor classification.

IX. CONCLUSION

We analyzed the performance of eight machine learning algorithms and one deep learning model on the CICDDoS2019 dataset for two scenarios. Scenario A classified benign vs DDoS traffic and Scenario B categorized DDoS attack types. In both scenarios, the models were initially trained on the control dataset consisting of 70 features. Next, the models were trained on the proposed subset of 25 time-based features. The top five models had accuracies of over 99% on the control features and over 98% on the time-based features in Scenario A. These results are as accurate as top DDoS classifiers seen in existing literature. Overall, the nine models had a median accuracy decrease of a mere 1.62% and a substantial median training time reduction of 36.28% when using only the proposed time-based features.

Scenario B classified specific DDoS traffic as MSSQL, SSDP, SYN Flood, PORTMAP, DNS, LDAP, NETBIOS, SNMP, TFTP, NTP, UDP Flood, or UDP-Lag. The top-performing model, XGB, had an accuracy of 74.08% on the control features and 69.05% on the time-based features. The nine classifiers had a median decrease of 7.43% in accuracy with a median training time reduction of 25.29% using only time-based features. Our top classifier's performance metrics matched or exceeded the accuracies of previously discussed multiclass works, especially considering the uniquely large number of attack types in our experiments. Overall, XGB, RF, LD, LGBM, and KNN were the top-performing models. XGB was slightly more accurate than the other four models; however, LD had the best speed vs accuracy tradeoff when using only time-based features. In certain scenarios, such as an application that incorporates incremental learning?, faster training times may be desirable over better accuracy, as the models are trained continuously, potentially in near-real time.

Lashkari *et al.* [5] established the effectiveness of the time-based feature set in detecting and classifying Tor traffic. Likewise, our work demonstrated time-based features viability in the domain of DDoS. The results found in this paper and Lashkari *et al.*'s work indicate that time-based features warrant further experimentation in traffic based research where a smaller dataset may prove beneficial to prevent overfitting and potentially improve accuracy.

ACKNOWLEDGMENT

Any opinions, findings and conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding sources.

REFERENCES

[1] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *Int. J. Distrib. Sensor Netw.*, vol. 13, no. 12, Dec. 2017, Art. no. 155014771774146, doi: 10.1177/1550147717741463.

[2] S. S. Mohammed, R. Hussain, O. Senko, B. Bimaganbetov, J. Lee, F. Hussain, C. A. Kerrache, E. Barka, and M. Z. A. Bhuiyan, "A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network," in *Proc. 14th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Oct. 2018, pp. 1–8, doi: 10.1109/WIMOB.2018.8589104.

[3] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 37–52, 3rd Quart., 2009, doi: 10.1109/SURV.2009.090304.

[4] X. Ying, "An overview of overfitting and its solutions," *J. Phys., Conf. Ser.*, vol. 1168, Feb. 2019, Art. no. 022022, doi: 10.1088/1742-6596/1168/2/022022.

[5] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *Proc. 3rd Int. Conf. Inf. Syst. Secur. Privacy*, 2017, pp. 253–262, doi: 10.5220/0006105602530262.

[6] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proc. IEEE 53rd Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8888419

[7] A. Lashkari, "CICFlowmeter-V4.0 (formerly known as ISCXFlowMeter) is a network traffic Bi-flow generator and analyser for anomaly detection," Canadian Institute of Cyber Security (CIC), Fredericton, New Brunswick, Tech. Rep., 2019. [Online]. Available: https://github.com/ISCX/CICFlowMeter, doi: 10.13140/RG.2.2.13827.20003.

[8] M. S. Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, ''DDoSNet: A deep-learning model for detecting network attacks,'' in *Proc. IEEE 21st Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Aug. 2020, pp. 391–396, doi: 10.1109/WOWMOM49955.2020.00072.

[9] A. E. Cil, K. Yildiz, and A. Buldu, ''Detection of DDoS attacks with feed forward based deep neural network model,'' *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114520, doi: 10.1016/j.eswa.2020.114520.

[10] M. A. Salahuddin, M. F. Bari, H. A. Alameddine, V. Pourahmadi, and R. Boutaba, ''Time-based anomaly detection using autoencoder,'' in *Proc. 16th Int. Conf. Netw. Service Manage. (CNSM)*, Nov. 2020, pp. 1–9, doi: 10.23919/CNSM50824.2020.9269112.

[11] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, ''Kitsune: An ensemble of autoencoders for online network intrusion detection,'' in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018.

[12] J. Chen, Y. Yang, K. Hu, H. Zheng, and Z. Wang, ''DAD-MCNN: DDoS attack detection via multi-channel CNN,'' *Proc. 11th Int. Conf. Mach. Learn. Comput. (ICMLC)* 2019, pp. 484–488, doi: 10.1145/3318299.3318329.

[13] S. S. Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, ''Machine learning based DDOS detection,'' in *Proc. Int. Conf. Emerg. Smart Comput. Informat. (ESCI)*, Mar. 2020, pp. 234–237, doi: 10.1109/ESCI48226.2020.9167642.

[14] O. Elejla, B. Belaton, M. Anbar, B. Alabsi, and A. Al-Ani, ''Comparison of classification algorithms on icmpv6-based DDoS attacks detection,'' in *Computational Science and Technology* (Lecture Notes in Electrical Engineering). Singapore: Springer, 2018, doi: 10.1007/978-981-13-2622-6_34.

[15] O. E. Elejla, M. Anbar, B. Belaton, and S. Hamouda, ''Labeled flow-based dataset of ICMPv6-based DDoS attacks,'' *Neural Comput. Appl.*, vol. 31, no. 8, pp. 3629–3646, Aug. 2019. [Online]. Available: https://link.springer.com/article/10.1007/s00521-017-3319-7

[16] R. F. Fouladi, O. Ermiş, and E. Anarim, ''A novel approach for distributed denial of service defense using continuous wavelet transform and convolutional neural network for software-defined network,'' *Comput. Secur.*, vol. 112, Jan. 2022, Art. no. 102524, doi: 10.1016/j.cose.2021.102524.

[17] Y. Hussain. (2020). *Network Intrusion Detection for Distributed Denial-of-Service (DDoS) Attacks Using Machine Learning Classification Techniques*. [Online]. Available: http://hdl.handle.net/1828/11679

[18] J. P. A. Maranhão, J. P. C. L. da Costa, E. P. de Freitas, E. Javidi, and R. T. de Sousa Júnior, ''Error-robust distributed denial of service attack detection based on an average common feature extraction technique,'' *Sensors*, vol. 20, no. 20, p. 5845, Oct. 2020, doi: 10.3390/s20205845.

[19] C.-S. Shieh, W.-W. Lin, T.-T. Nguyen, C.-H. Chen, M.-F. Horng, and D. Miu, ''Detection of unknown DDoS attacks with deep learning and Gaussian mixture model,'' *Appl. Sci.*, vol. 11, no. 11, p. 5213, Jun. 2021, doi: 10.3390/app11115213.

[20] S. Sindian and S. Sindian, ''An enhanced deep autoencoder-based approach for DDoS attack detection,'' *WSEAS Trans. Syst. Control*, vol. 15, pp. 716–724, Dec. 2020, doi: 10.37394/23203.2020.15.72.

[21] R. Al-Saadi, G. Armitage, J. But, and P. Branch, ''A survey of delay-based and hybrid TCP congestion control algorithms,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3609–3638, 4th Quart., 2019, doi: 10.1109/COMST.2019.2904994.

[22] B. Sun, L. Yang, W. Zhang, M. Lin, P. Dong, C. Young, and J. Dong, ''SuperTML: Two-dimensional word embedding for the precognition on structured tabular data,'' in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–9, doi: 10.1109/CVPRW.2019.00360.

[23] G. Ajay and P. Krishnan. (2018). *A Study and Analysis of Effective Data Transmission Using UDP*. [Online]. Available: https://www.ijser.org/researchpaper/A-Study-and-Analysis-of-Effective-Data-transmission-Using-UDP.pdf

[24] K. Alieyan, M. M. Kadhum, M. Anbar, S. U. Rehman, and N. K. A. Alajmi, ''An overview of DDoS attacks based on DNS,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2016, pp. 276–280, doi: 10.1109/ICTC.2016.7763485.

[25] L. Breiman, ''Using iterated bagging to debias regressions,'' *Mach. Learn.*, vol. 45, pp. 261–277, Dec. 2001, doi: 10.1023/a:1017934522171.

[26] K. Q. Weinberger, J. Blitzer, and L. Saul, ''Distance metric learning for large margin nearest neighbor classification,'' *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Jul. 2009, doi: 10.5555/1577069.1577078.

[27] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, ''LightGBM: A highly efficient gradient boosting decision tree,'' in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2017, pp. 3149–3157, doi: 10.5555/3294996.3295074.

[28] T. Chen and C. Guestrin, ''XGBoost: A scalable tree boosting system,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[29] R. Schapire, ''Explaining AdaBoost,'' in *Empirical Inference*. Berlin, Germany: Springer, 2013, pp. 37–52, doi: 10.1007/978-3-642-41136-6_5.

[30] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, ''Linear discriminant analysis: A detailed tutorial,'' *AI Commun.*, vol. 30, no. 2, pp. 169–190, 2017, doi: 10.3233/AIC-170729.

[31] A. Mansour, ''Texture classification using Naïve Bayes classifier,'' *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, pp. 1–9, 2018. [Online]. Available: http://paper.ijcsns.org/07_book/201801/20180113.pdf

[32] T. Evgeniou and M. Pontil, ''Support vector machines: Theory and applications,'' in *Machine Learning and its Applications*, vol. 2049. Berlin, Germany: Springer, 2001, pp. 249–257, doi: 10.1007/3-540-44673-7_12.

[33] J. Halladay, D. Cullen, and N. Briner. *DDoS Time-based 881 Experimentation, Github Repository*, Jan. 2022. [Online]. Available: https://github.com/jehalladay/DDoS_Research

[34] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, ''Selecting critical features for data classification based on machine learning methods,'' *J. Big Data*, vol. 7, no. 1, pp. 1–26, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

[35] G. Amrani, A. Adadi, M. Berrada, Z. Souirti, and S. Boujraf, ''EEG signal analysis using deep learning: A systematic literature review,'' in *Proc. 5th Int. Conf. Intell. Comput. Data Sci. (ICDS)*, Oct. 2021, pp. 1–8, doi: 10.1109/ICDS53782.2021.9626707.

JAMES HALLADAY is currently pursuing the bachelor's degree in math and computer science with Colorado Mesa University. His research interest includes the application of machine learning to the domain of cyber security.

DRAKE CULLEN is currently pursuing the bachelor's degree in computer science with minors in cybersecurity and mathematics with Colorado Mesa University (CMU). He is the former President of the Cybersecurity Club and a Research Fellow with the Cybersecurity Center.

NATHAN BRINER is currently pursuing the bachelor's degree in computer science and a minor in mathematics with Colorado Mesa University. He is also a Proud Member of the Cybersecurity Club and looks forward to doing more research involving machine learning.

JACKSON WARREN is currently pursuing the bachelor's degree in computer science with Colorado Mesa University. He is also working as a Cyber Security Research Assistant with CMU and starts as a Training Security Consultant with company Cloudrise, in the Fall of 2022. He is also researching darknet and encrypted traffic identification with data generation, with an interest in solutions engineering and data management.

KARSON FYE is currently pursuing the Bachelor of Science degree in computer science and minor in mathematics with Colorado Mesa University (CMU). He is also a Cyber Security Research Assistant at CMU and is also researching the impact that data generation has on classifying encrypted traffic. His research interests include cloud security and data engineering.

RAM BASNET received the B.S. degree in computer science from Colorado Mesa University (CMU), in 2004, and the M.S. and Ph.D. degrees in computer science from New Mexico Tech, in 2008 and 2012, respectively. He is currently an Associate Professor of computer science and cybersecurity with CMU. His research interests include the areas of information assurance, machine learning, and computer science pedagogy.

JEREMY BERGEN received the M.S. degree in computer science from the Georgia Institute of Technology, in 2021, where his focus was on machine learning. He is currently an Assistant Professor of computer science at Colorado Mesa University. His current research interests include computer vision, gamification in education, and cyber security research.

TENZIN DOLECK received the Ph.D. degree from McGill University, in 2017. He is currently working as a Canada Research Chair and an Assistant Professor with Simon Fraser University.