AN EFFICIENT PROGRAM TO DETECT DDOS ATTACKS USING MACHINE LEARNING ALGORITHMS

Kaige Bao[1] and Ang Li[2]

[1]Basis International Hangzhou, No.9 Yulin Road, Shangcheng District, Hangzhou, Zhejiang, China
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

This paper investigates the efficacy of machine learning algorithms for the detection of Distributed Denial of Service (DDoS) attacks [4][5]. The study explores different approaches, including Support Vector Machines (SVM), logistic regression, and decision trees, and evaluates their performance using metrics such as accuracy, precision, recall, and F1-score [6]. The results demonstrate the effectiveness of SVM models with polynomial or radial basis function (RBF) kernels, logistic regression models with a polynomial degree of 4, and decision tree models with depths exceeding 10 [7][8]. These algorithm configurations exhibit promising potential in mitigating DDoS attacks and safeguarding network infrastructures [9]. However, limitations such as dataset availability, imbalanced data, and the focus on offline detection warrant further research. Enhancements in these areas can lead to more robust and efficient DDoS detection systems. The findings of this study contribute to the advancement of network security and offer insights for organizations aiming to counter the growing threat of DDoS attacks.

KEYWORDS

Machine Learning, DDoS Attacking, Algorithms, Computer Science

1. INTRODUCTION

Throughout the history of the Internet, Distributed Denial of Service (DDoS) attacks have been a pervasive issue for numerous corporations and small organizations [10]. These attacks inflict significant harm on targeted entities, leading to the disruption and denial of normal services to customers. DDoS attacks involve the inundation of a system with an abnormal and massive influx of malicious data, depleting the system's resources and rendering it unable to function effectively.

DDoS attacks manifest in various forms, with two prominent examples being Botnet attacks and protocol-based attacks. In Botnet attacks, compromised devices are leveraged to continuously transmit data to a target, often employing methods such as UDP and ICMP floods [11]. On the other hand, protocol-based attacks focus on exploiting vulnerabilities within specific protocols, overwhelming a system's resources. A notable instance of this is the TCP SYN attack [12].

Despite advancements in technology, DDoS attacks continue to afflict large corporations, leading to substantial financial losses amounting to millions of dollars. The persistent nature of this problem poses significant challenges to organizations. One of the key difficulties lies in

David C. Wyld et al. (Eds): NSEC, InWeS, CSIP, BMLI, CACIT, EDUIT -2023
pp. 75-83, 2023. CS & IT - CSCP 2023 DOI: 10.5121/csit.2023.131507

distinguishing between a sudden surge in traffic as a result of an attack and legitimate traffic caused by events such as breaking news about an earthquake.

In this paper, we aim to delve deeper into the issue of DDoS attacks, examining their impact on organizations and exploring potential solutions to mitigate these threats. By understanding the nature of DDoS attacks and developing effective detection and prevention mechanisms, organizations can better protect their infrastructure and minimize the devastating consequences caused by these attacks.

In this paper, our goal was to tackle the challenge of effectively detecting and distinguishing Distributed Denial of Service (DDoS) attacks from normal network traffic. To achieve this, we adopted a comprehensive approach that involved utilizing various machine learning algorithms. The objective was to compare and identify the most effective algorithm for accurately classifying instances of abnormal traffic as either indicative of a DDoS attack or non-malicious [13].

We began by assembling a diverse dataset consisting of both DDoS attack instances and legitimate traffic samples. This dataset served as the foundation for training and evaluating our machine learning models. Performance metrics such as accuracy, precision, recall, and F1 score were used to assess each algorithm's effectiveness in classifying abnormal traffic instances while minimizing false positives and false negatives. We employed techniques like cross-validation and parameter tuning to improve the performance and generalization capabilities of the selected algorithms.

Through rigorous experimentation and evaluation, our aim was to identify the algorithm that demonstrated the highest detection accuracy and reliability. This chosen algorithm would then form the basis for building an effective DDoS detection system, enabling organizations to respond promptly and mitigate the damaging effects of such attacks.

In the subsequent sections of the paper, we presented our methodology, detailed the machine learning algorithms employed, discussed the experimental results, and provided an analysis of the selected algorithm's performance. Our findings contribute to the field of DDoS attack detection, providing valuable insights for organizations seeking to safeguard their networks from malicious activities.

By comparing and selecting the most effective algorithm, we offer a significant step forward in combating the persistent problem of DDoS attacks. With an accurate and reliable detection system in place, organizations can enhance their network security and minimize the financial and reputational damage caused by these disruptive attacks.

In the conducted experiments, we aimed to identify the most suitable machine learning algorithms and configurations for DDoS detection.

Experiment 1 compared Support Vector Machine (SVM) models using different kernel functions. The polynomial and radial basis function (RBF) kernels consistently outperformed linear and sigmoid kernels, showcasing superior performance on test data. Therefore, we recommend utilizing SVM models with polynomial or RBF kernels for DDoS detection.

Experiment 2 focused on logistic regression models with varying polynomial degrees. The model with a polynomial degree of 4 consistently demonstrated the best performance on the test data. Thus, selecting a polynomial degree of 4 for logistic regression is recommended.

Experiment 3 evaluated decision tree models with different tree depths. Models with depths greater than 10 consistently showcased satisfactory performance on the test data. Hence, decision tree models with depths exceeding 10 are suggested for DDoS detection.

In summary, based on the experimental results, we suggest employing SVM models with polynomial or RBF kernels, logistic regression models with a polynomial degree of 4, or decision tree models with depths greater than 10 for effective DDoS detection. These findings provide valuable insights for designing robust DDoS detection systems.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. The Dataset

The dataset is one of the major problems. Many datasets have the problem of biased data, and through training the dataset, it is hard to get a result that is neutral or fair. Many biased dataset usually will produce a result that is perfect for one aspect, but poorly on another. My dataset exists, and in the future, I could use a more comprehensive dataset with more detailed data.

2.2. Selection of Relevant Features

Selection of relevant features is another major problem. It is essential to determine which features are most informative for distinguishing DDoS attacks from normal traffic. Choosing an inappropriate set of features may lead to poor detection performance or excessive computational requirements. To address this, I could use techniques such as statistical analysis, domain knowledge, and feature importance ranking algorithms to identify and select the most discriminative features.

2.3. Handling High-Dimensional Data

Handling high-dimensional data is one of the major problems. Network traffic data can be high-dimensional, making feature engineering and processing computationally expensive and prone to overfitting. I could address this issue by employing dimensionality reduction techniques such as Principal Component Analysis (PCA) or feature selection methods to reduce the dimensionality of the data while retaining its informative aspects [14].

3. SOLUTION

Step1. Load and preprocess the data
Step2. Train the models on the training data with different hyperparameters
Step3. Making plots to compare the performance of each model
Load and preprocess the data

In this component, we prepare the data for training. We first load the data and shuffle it. Then we convert all categorical values into numerical values so that the model can read them. We also normalize the data into 0-mean and 1-std in order to make the model learning easier. At the end, we split the data into 80% training and 20% test.

```
df = pd.read_csv('dataset_sdn.csv')
df = df.sample(frac=0.1)
X = df.drop('label', axis=1)
y = df['label']
for c in X.columns:
X[c] = pd.factorize(X[c])[0]
scaler = preprocessing.StandardScaler().fit(X)
X = scaler.transform(X)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
```

Figure 1. Screenshot of code 1

Load the DDoS data. You can find it from https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-dataset. The dataset has 104345 rows and 23 columns. For simplicity, we randomly sampled 10% of them.

The last column is the label of each traffic. It can be either 1(malicious) or 0(benign). We make it the target value.

Convert each categorical value into numerical value.

Normalize the data so that each feature is 0-mean and unit(1) standard deviation.

Divide the data into 80% training and 20% test.

Train the models on the training data with different hyperparameters

In this component, we consider three classifiers, decision tree, logistic regression and support vector machine. We use the scikit-learn package where each model is pre-defined. For each type of classifier, we test with different hyperparameters. The metrics are accuracy, precision, recall and f1-score [15].

```
d_list = ['linear', 'poly', 'rbf',
'sigmoid']

for d in d_list:
model = SVC(kernel=d)

model.fit(X_train, y_train)
```

```
d_list = [1,2,3,4]

for d in d_list:
poly = PolynomialFeatures(d)
X_train =
poly.fit_transform(X_train)
X_test = poly.fit_transform(X_test)
model = LogisticRegression()

model.fit(X_train, y_train)
```

```
d_list = [2,5,10,15,20,21]


for d in d_list:
model =
DecisionTreeClassifier(max_depth=d)
model.fit(X_train, y_train)
```

Figure 2. Screenshot of code 2

First we test Support Vector Machine (SVM) with different types of kernel functions. The candidates are linear, poly (degree=3), rbf, sigmoid. We look at the accuracy, precision, recall and f1 score for each kernel (both training and test data).

Then we test logistic regression with different polynomial degrees. Because of the large size of features and the limit of our machine, we can have at most degree=4. It has 14950 features when we map 22 features to degree=4. We look at the same metrics.

Lastly, we look at decision tree models. The complexity of the decision tree depends on the max depth of tree allowed. The deeper the depth, the more complex the model. Here we consider depth=[2,5,10,15,20,21]. We look at the same metrics at the end.

Making plots to compare the performance of each model

In this component, we make plots using the matplotlib package. We make a 2 by 2 plot, where each subplot is one of the metrics. We also label each subplot with description so that they are supposed to be readable.

```
x_label = "Depth"

fig, ax = plt.subplots(2, 2)

fig.tight_layout(pad=3.0)
fig.suptitle("Accuracy, Precision, Recall, F1-score with Different Max
Depths")

ax[0, 0].plot(d_list, train_acc, color="blue", label="Training")
ax[0, 0].plot(d_list, test_acc, color="red", label="Test")
ax[0, 0].set_xlabel(x_label)
ax[0, 0].set_ylabel("Accuracy")
ax[0, 0].legend()

ax[0, 1].plot(d_list, train_pre, color="blue", label="Training")
ax[0, 1].plot(d_list, test_pre, color="red", label="Test")
ax[0, 1].set_xlabel(x_label)
ax[0, 1].set_ylabel("Precision")
ax[0, 1].legend()

ax[1, 0].plot(d_list, train_rec, color="blue", label="Training")
ax[1, 0].plot(d_list, test_rec, color="red", label="Test")
ax[1, 0].set_xlabel(x_label)
ax[1, 0].set_ylabel("Recall")
ax[1, 0].legend()

ax[1, 1].plot(d_list, train_f1, color="blue", label="Training")
ax[1, 1].plot(d_list, test_f1, color="red", label="Test")
ax[1, 1].set_xlabel(x_label)
ax[1, 1].set_ylabel("F1-score")
ax[1, 1].legend()
```

Figure 3. Screenshot of code 3

4. EXPERIMENT

4.1. Experiment 1

We look at the support vector machine models with different types of kernel functions.
We specifically look at linear, polynomial (with degree=3), radial basis function (rbf), and sigmoid kernels. For each of the kernels, we train it with the training data. We look at the accuracy, precision, recall, f1-score on both training and test data.

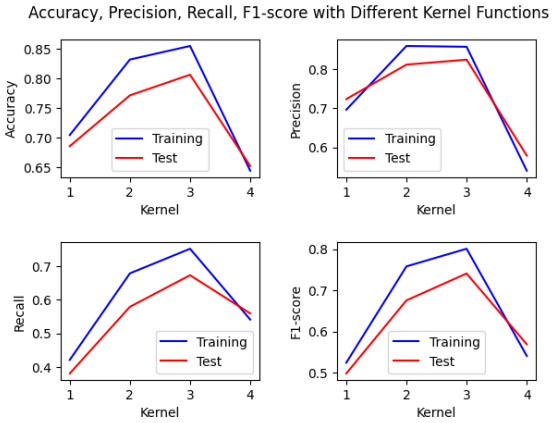

Figure 4. Accuracy, Precision, Recall, F1-score with Different Kernel Functions

For the plot, both polynomial and rbf kernels give us the best test data performance. They have a higher score on those metrics than the others. So we conclude that we should pick these kernels for the SVM model.

4.2. Experiment 2

We look at the logistic regression models with different values of polynomial degrees. Because of the large size of features and the limit of our machine, we can have at most degree=4. It has 14950 features when we map 22 features to degree=4. We look at the same metrics. From the plot, we conclude that it gives best test data performance when the polynomial degree=4.

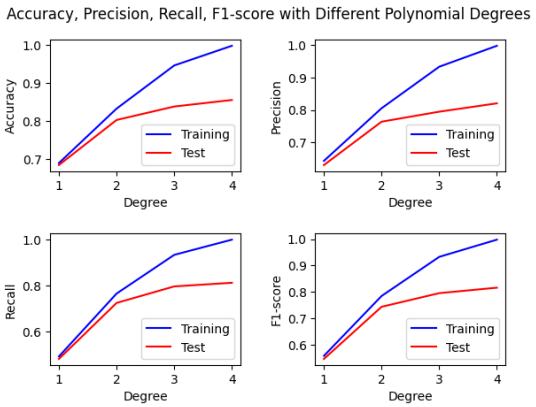

Figure 5. Accuracy, Precision, Recall, F1-score with Different Polynomial Degrees

We look at decision tree models. The complexity of the decision tree depends on the max depth of tree allowed. The deeper the depth, the more complex the model. Here we consider depth=[2,5,10,15,20,21]. We look at the same metrics at the end. From the plot, we can conclude that all depths greater than 10 are performing well in the test data.

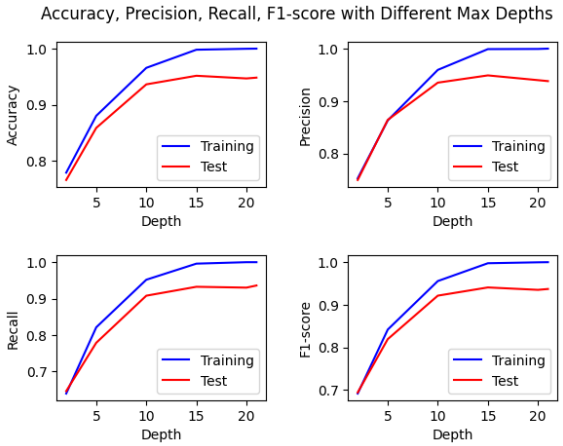

Figure 6. Accuracy, Precision, Recall, F1-score with Different Max Depths

5. RELATED WORK

Methodology A addresses the urgent need for rapid and effective detection of DDoS attacks, considering the limitations of signature-based and anomaly-based detection systems. It emphasizes the importance of analyzing fundamental features of DDoS attacks, as they can vary in terms of port/protocol and operation method. The paper utilizes chi-square and Information gain feature selection mechanisms to identify important attributes. Several machine learning models, including Navies Bayes, C4.5, SVM, KNN, K-means, and Fuzzy c-means clustering, are developed for efficient DDoS attack detection. Experimental results reveal that Fuzzy c-means clustering achieves higher accuracy in identifying attacks. Comparatively, our project explores different machine learning algorithms but does not specifically focus on feature selection mechanisms or clustering techniques [1].

This work focuses on protecting cloud environments from DDoS attacks, which are critical threats compromising network availability. The study highlights the need to detect and counter these sophisticated attacks, which continue to grow rapidly. The research specifically employs the ownCloud environment and Tor Hammer as an attacking tool to generate a new dataset for Intrusion Detection System (IDS). Several machine learning algorithms, including Support Vector Machine (SVM), Naive Bayes, and Random Forest, are utilized for classification, achieving high overall accuracies of 99.7%, 97.6%, and 98.0%, respectively. In comparison, our project explores different machine learning algorithms without specific emphasis on cloud environments or generating a new dataset [2].

Methodology C introduces a novel framework called PCA-RNN (Principal Component Analysis-Recurrent Neural Network) for DDoS attack detection. It utilizes PCA to reduce feature dimensionality and enhance computational efficiency while retaining essential information. The reduced-dimensional data is then fed into an RNN model for training and obtaining the detection model. Evaluation results demonstrate that PCA-RNN outperforms existing methods in terms of accuracy, sensitivity, precision, and F-score when tested on real datasets. The paper provides valuable insights into the effectiveness of the PCA-RNN framework for DDoS attack identification and offers a distinct approach compared to our project's exploration of different machine learning algorithms [3].

6. CONCLUSIONS

Despite the progress made in our DDoS detection project, there are several limitations that need to be acknowledged. These include the limited dataset used for training and evaluation, the challenge of handling imbalanced data, the focus on offline detection rather than real-time detection, potential areas for improvement in feature selection and engineering, and the need for further model optimization and tuning.

To address these limitations with more time, we would prioritize collecting a larger and more diverse dataset, employing advanced techniques like oversampling or under sampling to handle imbalanced data, implementing streaming algorithms for real-time detection, refining feature selection and engineering processes, and conducting additional rounds of model optimization and fine-tuning. These measures would enhance the project's accuracy, robustness, and scalability. By considering these improvements, our DDoS detection system would offer more reliable and effective performance in detecting and mitigating DDoS attacks.

This study highlights the effectiveness of machine learning algorithms for DDoS attack detection. The recommended approaches, including SVM, logistic regression, and decision trees, provide valuable insights for organizations seeking to protect their networks. Further research is needed to address limitations and enhance performance in real-time scenarios.

REFERENCES

[1] Suresh, Manjula, and R. Anitha. "Evaluating machine learning algorithms for detecting DDoS attacks." Advances in Network Security and Applications: 4th International Conference, CNSA 2011, Chennai, India, July 15-17, 2011 4. Springer Berlin Heidelberg, 2011.

[2] Wani, Abdul Raoof, et al. "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques." 2019 Amity International conference on artificial intelligence (AICAI). IEEE, 2019.

[3] Li, Qian, et al. "DDoS attacks detection using machine learning algorithms." Digital TV and Multimedia Communication: 15th International Forum, IFTC 2018, Shanghai, China, September 20–21, 2018, Revised Selected Papers 15. Springer Singapore, 2019.

[4] Mahesh, Batta. "Machine learning algorithms-a review." International Journal of Science and Research (IJSR).[Internet] 9 (2020): 381-386.

[5] Lau, Felix, et al. "Distributed denial of service attacks." Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0. Vol. 3. IEEE, 2000.

[6] Hearst, Marti A., et al. "Support vector machines." IEEE Intelligent Systems and their applications 13.4 (1998): 18-28.

[7] Kecman, Vojislav. "Support vector machines–an introduction." Support vector machines: theory and applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. 1-47.

[8] Musavi, Mohamad T., et al. "On the training of radial basis function classifiers." Neural networks 5.4 (1992): 595-603.

[9] Cheng, Xiangle, et al. "Safeguard network slicing in 5G: A learning augmented optimization approach." IEEE Journal on Selected Areas in Communications 38.7 (2020): 1600-1613.

[10] Chang, Rocky KC. "Defending against flooding-based distributed denial-of-service attacks: A tutorial." IEEE communications magazine 40.10 (2002): 42-51.

[11] Malik, Manisha, and Maitreyee Dutta. "Contiki-based mitigation of UDP flooding attacks in the Internet of things." 2017 International Conference on Computing, Communication and Automation (ICCCA). IEEE, 2017.

[12] Berguiga, Abdelwahed, and Ahlem Harchay. "An IoT-Based Intrusion Detection System Approach for TCP SYN Attacks." Computers, Materials & Continua 71.2 (2022).

[13] Stevanovic, Dusan, Natalija Vlajic, and Aijun An. "Detection of malicious and non-malicious website visitors using unsupervised neural network learning." Applied Soft Computing 13.1 (2013): 698-708.

[14] Abdi, Hervé, and Lynne J. Williams. "Principal component analysis." Wiley interdisciplinary reviews: computational statistics 2.4 (2010): 433-459.
[15] Lipton, Zachary Chase, Charles Elkan, and Balakrishnan Narayanaswamy. "Thresholding classifiers to maximize F1 score." arXiv preprint arXiv:1402.1892 (2014).