Detection of DDoS Attacks with Gaussian Mixture Model

Alessandro Cecchetto[1], Giuseppa Conte[1], and Christian Napoli[1,2,3][0000−0002−3336−5853] ⋆

[1] Department of Computer, Automation and Management Engineering, Sapienza University of Rome, Via Ariosto 25 Roma 00185 RM, Italy
[2] Institute for Systems Analysis and Computer Science, Italian National Research Council, Via dei Taurini 19 Roma, Italy
[3] Department of Computational Intelligence, Czestochowa University of Technology, ul. J.H. Dabrowskiego 69, 42-201 Czestochowa, Poland.
cecchetto.1941039@studenti.uniroma1.it, giusy.conte@uniroma1.it, cnapoli@diag.uniroma1.it

Abstract. A Distributed Denial of Service (DDoS) is an attack which aim is to stop or tamper with an online service incapacitating a server with a flood of packages or requests, using internet or intranet. The main aim of the DDoS attack is to collapse the network or server with abnormal traffic to make the service unavailable for the legitimate users. This problem is particularly profound, due to the development of emerging technologies, such as cloud computing, the Internet of Things or artificial intelligence techniques, from which attackers can take advantage by launching a huge volume of DDoS attacks at a lower cost, and it is much harder to detect and prevent DDoS attacks, because DDoS traffic is similar to normal traffic. In this paper we implement a novel technique implementing an unsupervised Gaussian Mixture Model (GMM) based algorithm. Using a real traffic dataset, the CIC-DDoS2019, for benchmark, the proposed GMM can achieve recall, precision, and accuracy up to 99%. Experiments reveal that this can be a promising solution for the detection of DDoS attacks.

Keywords: Distributed Denial of Service (DDoS) · Gaussian Mixture Model (GMM) · Machine Learning · Detection

⋆ This work has been developed at *is.Lab()* Intelligent Systems Laboratory at the Department of Computer, Control, and Management Engineering, Sapienza University of Rome (https://islab.diag.uniroma1.it). This paper has been partially supported by the Age-It: Ageing Well in an ageing society project, task 9.4.1 work package 4 spoke 9, within topic 8 extended partnership 8, under the National Recovery and Resilience Plan (PNRR), Mission 4 Component 2 Investment 1.3 - Call for tender No. 1557 of 11/10/2022 of Italian Ministry of University and Research funded by the European Union - NextGenerationEU, CUP B53C22004090006.

1 Introduction

DoS (Denial of Service) attacks deplete the network bandwidth and computing resources of a targeted system by flooding malicious traffic, preventing the target system from offering regular services to legitimate users. DDoS (Distributed Denial of Service) goes even further on a much larger scale. DDoS attacks can take over a large number of compromised systems called bots, constituting a botnet, which are used to launch coordinated attacks on the victim system, from this kind of attack behavior, DDoS attacks can be devided in several branches as reported in [1]. Along with the emergence and advancement of disruptive Internet technologies, DDoS attacks are evolving and proliferating in scale, frequency, and sophistication. Organizations face potential threats to their network environment that may cause severe impacts to their operations, such as business downtime, data breaches, or even ransom demands from hackers [17]. The detection of DDoS attacks is essential before any mitigation approaches can be taken. In the early era, the alarm of DDoS attacks was triggered by rules programmed by traffic engineers, but in the current cybersecurity scenario, the application of artificial intelligence and, more specifically, machine learning (ML) offers new and promising perspectives. Training predictive models to recognize anomalous patterns in network traffic provides a more agile and proactive means of detecting attacks. The Gaussian Mixture Model (GMM) is one such model, used for its ability to model complex distributions of data, such as those that characterize network traffic. In this paper, we propose a novel ML (Machine Learning) method based on GMM, for detecting DDoS malicious packets. The remaining part of this paper is organized as follows: Section II describes and analyze the related DDoS detection works with some proposed solutions. Section III describes and analyze the used dataset. Section IV explain all the steps that belongs to the workflow from the feature pre-processing to the GMM. Section V denotes the implementation of the project. Section VI resume the obtained results of the project also taking into account the phase of Cross-Validation. Finally, conclusions and future implementations are discussed in Section VII.

2 Related Works

Various ML technologies have been employed, mainly as classifiers, in the detection of DDoS attacks. Meng Wang *et al.* [2] proposed a dynamic multilayer perceptron (MLP) combined with a feature selection technique to detect DDoS attacks, where a feedback mechanism is applied to promote and reconstruct the detector system when detection is not accurate. In their model, as the complexities of traffic network increase and change, some of the selected features will not be able to distinguish the traffic and normal attacks and determine the failure therein. Nhu-Ngoc Dao *et al.* [3] proposed the approach of source based IP filtering technique to defeat DDoS attacks. The approach try to distinguish three kinds of Users. The malicious user who has fix source IP address and injects spoofed packets to the switch infinitely. The DDoS attacking user sends

spoofed packets to the switch infinitely. The frequent user acts as normal user. The method distinguishes them and processes differently according to different users. It works well when the attack traffic is not very huge, and if the attack type it is mainly a flooding one. But to use it, we need to survey the network first and initiate two parameters for the detection method, so it is necessary a kind of "setup" time in order to fix such parameters. The effect of the method may be affected by the artificial parameters. When involved with the behavior of the artificial, the uncertainty of the detecting result will increase. Seyed Mohammad Mousavi *et al.* [4] proposed a solution to detect DDoS attacks based on the entropy variation of the destination IP address. Although it is a lightweight and effective detection method, in detecting DDoS attacks, we cannot only take one factor into consideration, since there are many factors that can be used to identify DDoS attacks, which can manifest themselves in very different ways beyond the proposed control. The detection method lacks of comprehensive consideration of multi-factors. Uygar Dincalp *et al.* [5] proposed a method based on the clustering algorithm DBSCAN for analyzing the network traffic in order to catch the changes and varieties in attack vectors for showing what the attack and where the attack is based on. The proposed methodology strongly depends on a fixed threshold in order to send an alarm of a possible attack, it is not clearly explained how the threshold is defined but the results shows that the proposed system worked well with chosen attributes in their experiments. Akella *et al.* [6] proposed a detection mechanism where each intermediate router detects traffic anomalies using profiles of normal traffic. Each router keeps track of destinations whose traffic occupies greater than a fraction of the capacity of the outgoing link, and sends this information to its neighbors. Attack detection is determined by intermediate routers if the gathered traffic information on a specific destination system exceeds the predefined threshold. This scheme cannot distinguish the flash crowds provided by a spike of normal traffic from the DDoS attacks. Hence, false alarm rate will be increased.

3 Dataset

The proposed model is trained and validated on the dataset released by the Canadian Institute for Cybersecurity (CIC), namely CIC-DDoS2019 [18]. The dataset offers an extended set of Distributed Denial of Service attacks, most of which employ some form of amplification through reflection. This type of attacks are conducted concealing the attacker's identity thanks to the IP spoofing technique [7] in which packets are sent to reflector servers by attackers with the source IP address set to the target victim's. The dataset contains benign and the most up-to-date common DDoS attacks, this two classes are divided as reported in Figure 1, with a total amount of data of 431,371 records described by 88 features.

A slight imbalance is highlighted in the composition of the dataset, where the connections associated with the attacks represent approximately 77% of the total dataset, this slight lack of data for the benign connections would have been

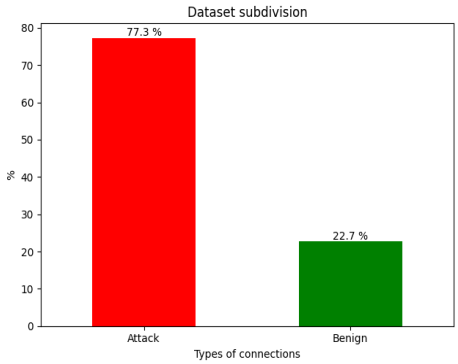


Fig. 1: Subdivision of the dataset.

possible to fix through oversampling algorithms, creating artificial data based on characteristics of the original ones, the choice in this experiment was not to make changes in this sense, as the imbalance present between the two classes was not too accentuated, equally allowing correct training of the model, the model performance evaluation is validated by applying the cross-validation step.
The following types of attacks are present: UDP, MSSQL, Portmap, Syn, NetBIOS, UDPLag, LDAP, DrDoS_DNS, UDP-lag, WebDDoS, TFTP, DrDoS_UDP, DrDoS_SNMP, DrDoS_NetBIOS, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NTP.

4 Implementation

The proposed work (Figure 2) consists of two main phases: data phase and model phase.
During data phase, the following points are developed: Feature pre-processing, Feature selection, Dimensionality reduction. During the model phase, relating to the development of the Gaussian mixture model, the following points are addressed: Model training, Performance evaluation, Cross-Validation.

4.1 Feature pre-processing

The efficiency of classification techniques have to be improved through adequate data manipulation which concerns different types of actions. Additionally, models trained on manually prepared data exhibited better performance compared to those trained on non-prepared data according to [8].

Feature Scaling Feature scaling is a vital step in pre-processing data before building a model using machine learning [9]. The dataset used for model training in machine learning often contain unpredictable values that may have varying scales. This can result in inequalities in comparing these values. Feature scaling

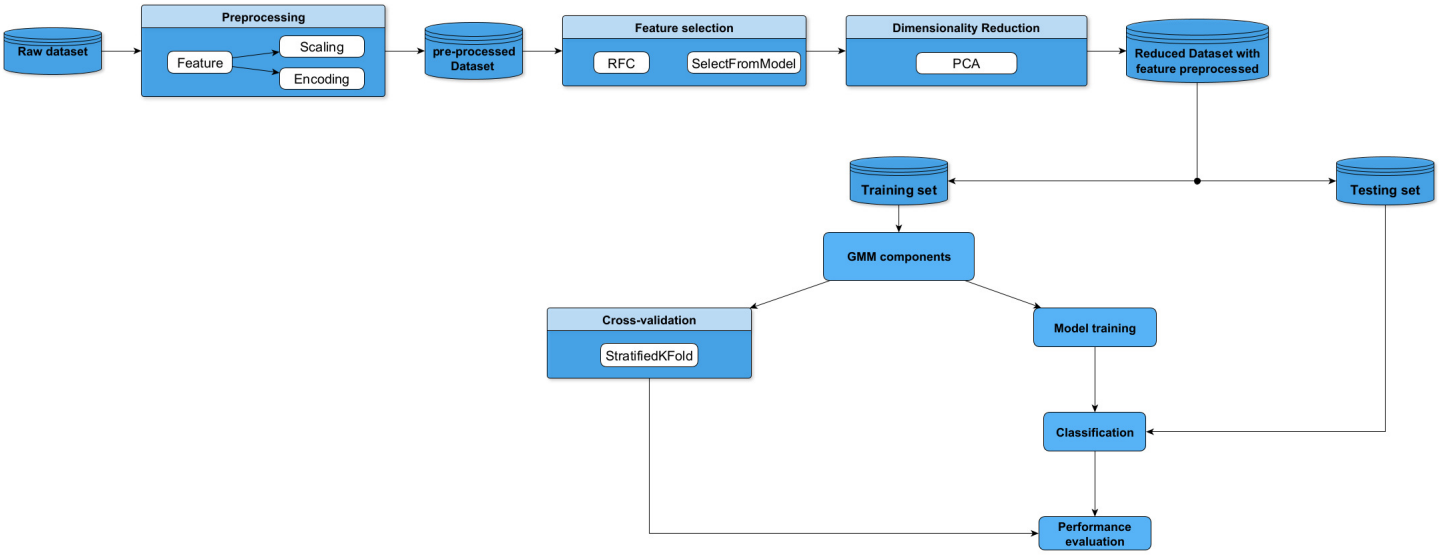


Fig. 2: Experiment workflow.

techniques can address these challenges by adjusting the values and promoting easy and fair comparisons among values.
The ML algorithm observes only numbers, and if there is a significant difference in range, it assumes that numbers in the upper ranges are superior such that features with larger numerical values have a greater effect on the distance between data and dominate other features when calculating distances. As a result, these more significant numbers play a more critical role during model training. The scaling technique used in the experiment is the Normalization, also known as Min-Max scaling, a technique in which values are shifted and rescaled to a range between 0 and 1 without distorting differences in the ranges of values or losing information.

Encoding Many statistical learning algorithms require as input a numerical feature matrix, as the case of GMM. When categorical variables are present in the data, feature engineering is needed to encode the different categories into a suitable feature vector.
Feature encoding is the process of transforming textual data into numerical values so they may be applied to ML algorithms, resulting in improved model accuracy. Researchers have used many approaches to convert textual data into numerical values, in this work is used the "Label Encoding" technique [11], in such work this process is applied to the target feature of the dataset in which there are two different values, *Attack* and *Benign*. Label encoding has the advantage that it is straightforward, yet it has the disadvantage that the numeric values can be "misinterpreted" by the machine learning algorithms since it uses

number sequencing. The problem using the number is that they introduce relation/comparison between them. Apparently, there is no relation between Attack and Benign, but when looking at the number, 'Benign' which is encoded using 1 has higher precedence over 'Attack' which is encoded using 0, this kind of problem in this work does not occur.

Feature selection Many ML models experience difficulty working with a high presence of features in input, generally, features can be categorized as: relevant, irrelevant, or redundant, the last two categories only increase the size of the input space [12] resulting in difficulty to process data further thus not contributing to the learning process. To generate the best performing model, feature selection plays a major role, which process a subset from available features data are selected for the process of learning algorithm. The best subset is the one with least number of dimensions that most contribute to learning accuracy, since an irrelevant feature does not affect describing the target concept in any way, a redundant feature does not add anything new to describing the target concept. Redundant features might possibly add more noise than useful information in describing the concept of interest. The main benefits of feature selection are follows: (i) reducing the measurement cost and storage requirements, (ii) coping with the degradation of the classification performance due to the finite size of training sample sets, (iii) reducing training and utilization time, (iv) facilitating data visualization and data understanding, (V) reducing the risk of *overfitting*. This process can be carried out into three ways [14]: filter, wrapper, and embedded.
The feature selection process of the proposed work uses a filter approach based on the Random Forest Classifier (RFC) as in according with [13], since it shows the most suitable performance among other filtering approaches, before the application of the model, the dataset is divided into train and test sets in the ratio of 80:20, since feature selection using only the training data (train set) rather than the entire dataset, is particularly important in order to avoid so-called "data leakage". The main reason behind this choice is that, when selecting features, you want the selection to be based only on the information available during the model training phase. If you also use data from the test set during the feature selection phase, you may run the risk of using future information to guide feature selection, introducing a bias into the results. The RFC takes the training dataset and resample it according to a procedure called "bootstrap". Each sample contains a random subset of the original columns and is used to fit a decision tree. Each tree of the random forest can calculate the importance of a feature according to its ability to increase the pureness of the leaves. The higher the increment in leaves purity, the higher the importance of the feature. This is done for each tree, then is averaged among all the trees and, finally, normalized to 1. So, the sum of the importance scores calculated by a Random Forest is 1. In this paper Information Gain (IG) criteria is used for feature selection by RFC. To use Information Gain for feature selection an entropy value of each attribute of the data has to be calculated. The entropy value is used for rank-

ing features that affect data classification. A feature which does not have much effect on the data classification has very small information gain and it can be ignored without affecting the detection accuracy of a classifier [15].It calculates the amount of entropy (uncertainty) that is reduced as a result of dividing the data by a specific property. Hence for each splitting attribute, information gain is calculated and the attribute with highest gain is chosen as splitting attribute. This attribute is such that it creates minimum impurity or randomness in the generated splits and hence it minimizes the information needed to classify the tuples. The entropy of a subset S is determined as follows:

$$Entropy(S) = \sum_{c=1}^{C}(p_i \cdot log_2(p_i)) \tag{1}$$

Here c = 1, .., C are the different classes in S, p_i is probability that an arbitrary tuple in S belongs to class C_i. Let A be a feature in S and $a_1, a_2, \ldots, a_v$ are different values of attribute A in S such that $S_1, \ldots, S_v$ are partitions generated based on these values. These partitions are likely to be impure. How much more information is still needed to arrive at an exact classification or pure partition is given as:

$$Entropy_A(S) = \sum_{i=1}^{v}(\frac{|S_i|}{|S|} \cdot Entropy(S_i)) \tag{2}$$

The smaller is this additional information the greater the purity of the partition.

$$IG(A) = Entropy(S) - Entropy_A(S) \tag{3}$$

Once the IG relating to all the features of the dataset has been defined via RFC, the Select From Model class is applied to the results for the selection of the main important features by defining a threshold based on the average of the values coming from the RFC. Then, the features that have the IG above the threshold are selected, which are shown in Figure 3.

Dimensionality reduction The dataset, after having gone through a preprocessing and feature selection phases, still has a high dimensionality with the presence of 28 features. In this regard, a reduction of the latter is necessary, as we want to have a representation of the data. Dimensionality reduction has been made using Principal Component Analysis (PCA) [16]. Principal Component Analysis (PCA) simplifies the complexity in high-dimensional data while retaining trends and patterns. It does this by transforming the data into fewer dimensions with minimal loss of overall dispersion, which act as summaries of features. PCA reduces data by geometrically projecting them onto lower dimensions called principal components (PCs) defined as a linear combination of the data's original variables, with the goal of finding the best summary of the data using a limited number of PCs. The first PC is chosen to minimize the total distance between the data and their projection onto the PC. By minimizing this distance, we also maximize the variance of the projected points. The second

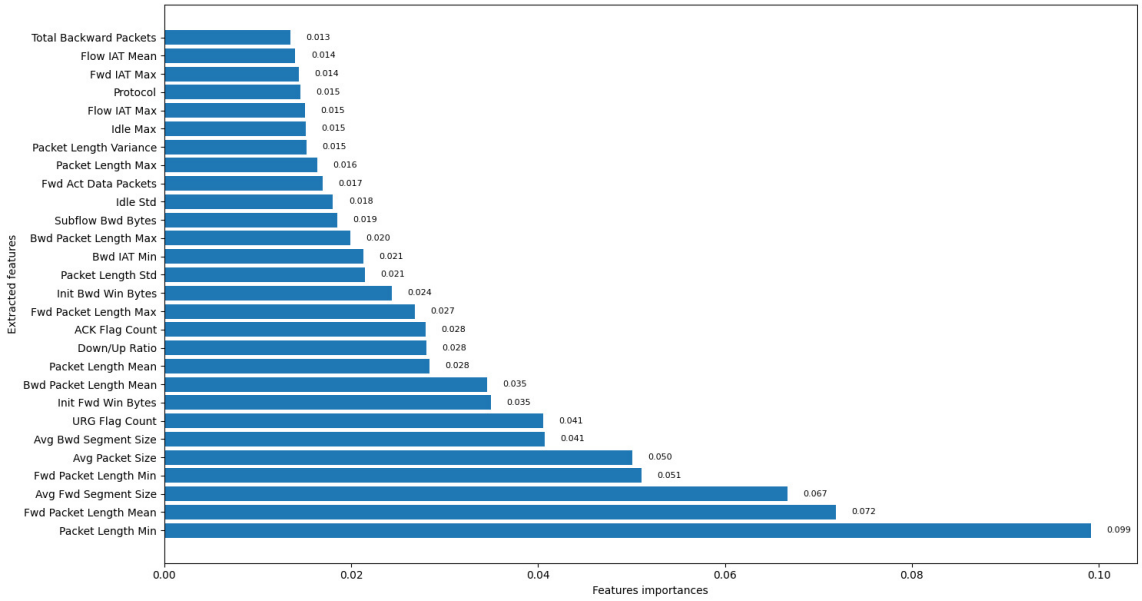


Fig. 3: Selected features.

(and subsequent) PCs are selected similarly, with the additional requirement that they be orthogonal (proving to be uncorrelated) with all previous PCs. Hence, principal components represents the directions of the data that explain a maximal amount of variance, or rather, the lines that capture most information of the data. The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more information it has. Hence, PCs are new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible. Given the dealing of the PCs with distance, a fundamental step required in order to avoid falsification of measurements is data standardization, since heterogeneous data representations going to influence the PCs constructions, given that in case of small set of variables has a much larger magnitude than others, the components in the PCA analysis are heavily weighted along those variables, while other variables are ignored. As a consequence, the PCA simply recovers the values of these high-magnitude variables for this reason standardise the scale of features variation is essential, the consequence that this aspect can have is shown in Figure 4 with a comparison on the first principal component whether features are standardized or not. The first application of PCA shown in Figure 5 is performed by defining a number of 10 PCs to see and analyze how the cumulative variance is distributed over a larger number of components. Subsequently, 3 components were extracted from the results obtained, since with 3 components it is possible

to visualize the data, reduce the computational power, finding only a data loss of 8%.

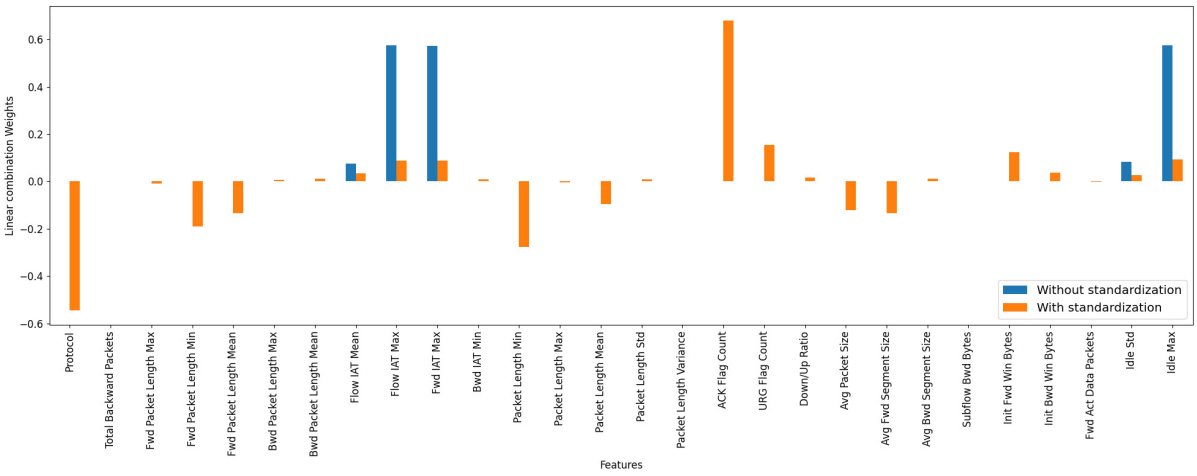


Fig. 4: Influence of the characteristics on the first principal component.

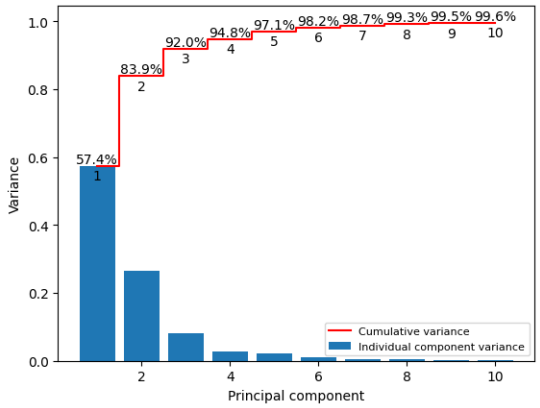


Fig. 5: Cumulative Explained Variance applying PCA.

4.2 GMM

Gaussian Mixture Model (GMM) [19] is a probabilistic model to describe subsets of data within a general population, that can be represented as a combination of normally distributed subpopulations. It is commonly used for unsupervised learning to learn the subpopulations models that can be also automatically divided. On the other hand within a supervised approach, such models can be

used in order to determine the boundaries of different subpopulations for classification purposes. In this latter case, the goal is to assign each data point to one of G preexisting unordered classes (or populations) taking into account d observed variables, X. The problem amounts to define a function that maps an arbitrary observation $x \in R^d$ to a prediction of the class from which it stems. This function, named allocation rule, has to be estimated from the training data $\{x_{ig}, \quad i = 1, ..., n_g \quad and \quad g = 1, ..., G\}$, which consist of the observations of vector X and class membership for $n = \sum_{g=1}^{G} n_g$ items. In supervised classification, their use is related to the Bayes allocation rule, which is known to minimize the expected error rate in class membership prediction. The Bayes rule suggests to allocate x to the class $\hat{g}$ having the highest posterior probability among the other classes C, that is:

$$\hat{g} = arg \underset{g=1,...,C}{max} \{\pi_g \, p_g(x)\} = arg \underset{g=1,...,C}{max} \{P(C_g) \, P(x|C_g)\} \tag{4}$$

Where π denotes the a priori probability of class g which are known or estimated from the training data, and $p_g(x)$ denotes the class-conditional probability distribution of X. For continuous predictors mixtures of multivariate normal densities are preferred, because of their computational convenience:

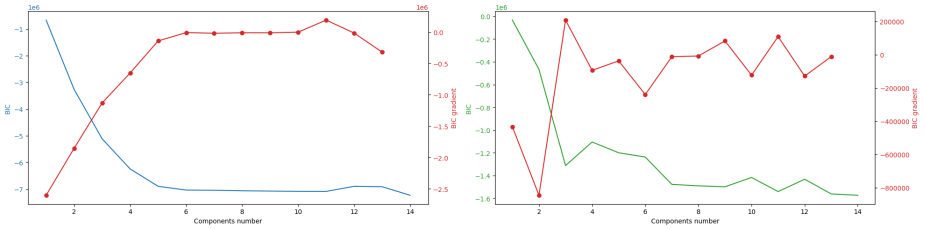
$$p_g(X) = \sum_{h=1}^{H_g} (w_{gh} \, \phi(x \,|\, \mu_{gh}, \, \Sigma_{gh})) \tag{5}$$

In mixture (5) the components $\phi(\cdot)$ are H d-dimensional Gaussian densities, each parameterized by its mean vector μ_{gh} and covariance matrix Σ_{gh}. The parameters w_{gh} $(h = 1, ..., H_g)$ are mixing proportions named also mixture coefficient, which are constrained to be positive quantities that sum to 1. These parameters are determined by fitting model (5) to the data $\{x_{ig}, \quad i = 1, ..., n_g \quad and \quad g = 1, ..., G\}$, usually by maximum likelihood via the expectation-maximization (EM) algorithm [20]. Then, the model is plugged into rule (4).

Components estimation In order to improve the performance of GMM as a classifier, we need to define correctly how many Gaussian components should be used to approximate the data distribution, a GMM with too many Gaussian components may overfit the data [22], while a GMM with too few components may not be flexible enough to approximate the true underlying density distribution, observing an underfitting of the data. The correct components estimation it was done using the Bayesian Information Criterion (BIC) [21], which is a criterion for grading models based on the posteriori probability of the models being compared. The Bayesian information criterion (BIC) is given by:

$$BIC = -2 \, \log f(x_n|\Theta) + p \, \log n, \tag{6}$$

where $f(x_n|\Theta)$ is the chosen model, p is the number of parameters to be estimated and n is the number of sample data points. Models with smaller values



(a) BIC related to Attack data. **(b)** BIC related to Benign data.

Fig. 6: Changes in BIC for different components number.

of BIC are preferable. Notice as sample size increases, BIC provides an increasingly larger penalty per parameter and thus tends to select more parsimonious models. In fact, BIC tends to overly penalize complex models, so the goal is to find the model for which the result is minimized, preferring the point of the first "elbow" of the function.

5 Experimental setup

Model implementation In order to have a better prevision on the data, two GMMs are implemented in two different ways, one regards only Benign data type while the other one regards only Attack data type. Hence, for extracting the correct number of components of each one of the model, the BIC is applied for each case and the change for different component values is presented in Figure 6. Given the obtained results, for what concern the GMM related to Attack data, is chosen five as number of components, though corresponding to six components there is the lowest BIC value, performances are equal as using five, so even in terms of computational complexity five was chosen, on the other hand GMM related to Benign data is developed using three components given the clear result. Hence, the two different models are defined as follows:

$$f_{Attack}(x) = \sum_{i=1}^{5} (w_{Attack,i}\, \phi(x \mid \mu_{Attack,i},\, \Sigma_{Attack,i})) \tag{7}$$

$$f_{Benign}(x) = \sum_{i=1}^{3} (w_{Benign,i}\, \phi(x \mid \mu_{Benign,i},\, \Sigma_{Benign,i})) \tag{8}$$

6 Results

For the classification of a new point, the posterior probability of belonging to the two classes is calculated:

$$\underset{i \in \{Attack, Benign\}}{P(C_i|x)} = \frac{P(x|C_i)P(C_i)}{P(x)}, \tag{9}$$

then, the point is associated with the class $\hat{g}$ with the maximum posterior probability:

$$\hat{g} = arg \max_{i \in \{Attack, Benign\}} \{P(C_i)\, P(x|C_i)\} \tag{10}$$

The metric used for evaluating the quality of a model's predictions is the balanced accuracy, which is the mean of Sensitivity and Specificity. Where Sensitivity (True Positive Rate) is the probability of a positive case being accurately classed as being positive, and Specificity (True Negative Rate) is the probability of a negative case being accuracy classed as negative. This specifications helps the metric perform well with the slightly imbalanced dataset used, returning the average accuracy per class. Using the average of Sensitivity and Specificity, we are able to account for imbalanced datasets as a model will receive a worse balanced accuracy score if it only predicts accurately for the majority class in the dataset. As a result, the balanced accuracy report an accuracy equal to 99%. Following is presented a resume of the obtained result using a confusion matrix and then a classification report, over 86275 samples used as a test data:

	Attack	Benign
Attack	66361	295
Benign	292	19327

Table 1: Confusion matrix.

	Precision	Recall	F1-score	Support
Attack	1.00	1.00	1.00	66656
Benign	0.98	0.99	0.99	19619
Accuracy			0.99	86275
Macro avg	0.99	0.99	0.99	86275
Weighted avg	0.99	0.99	0.99	86275

Table 2: Classification report.

Cross-Validation Cross-validation is one of the most widely used data resampling methods to estimate the true prediction error of models. The Cross-validation technique used in the experiment is the Stratified k-fold, in which the available learning set is partitioned into k disjoint subsets of approximately equal size, where the presence of data typology is the same as the original dataset. The

word "fold" refers to the number of resulting subsets. This partitioning is performed by randomly sampling cases from the learning set without replacement. The model is trained using $k-1$ subsets, which, together, represent the training set. Then, the model is applied to the remaining subset, which is denoted as the validation set, and the performance is measured. This procedure is repeated until each of the k subsets has served as validation set. Consequently, the average of the k performances on the k validation sets is calculated, which represents the cross-validation performance and subsequently the standard deviation is also calculated considering the result of each validation set compared to the performance score.

Cross-validation performance	98.7%
Standard deviation	0.003

Table 3: Cross-Validation results.

7 Conclusions

In this paper, a procedure for fitting Gaussian mixture models (GMM) oriented to supervised classification has been proposed for classifying DDoS attacks. The proposed method demonstrates exceptional adaptability, flexibility, and performance in detecting DDoS attacks, making it a promising solution for enhancing cybersecurity in critical infrastructures. Starting from the raw data taken from the CIC-DDoS2019 dataset, a pre-processing phase was necessary in which the latter were processed with a subsequent selection of the characteristics relevant to the purpose. The results from the test data, provides an accuracy of 99% which is confirmed even by the Cross-Validation with a 98.7% accuracy, highlighting a reliable result without problems like overfitting that in some scenario can conceal the real performance. As part of our future work, first of all, we will be focusing on improving the DDoS attack data, increasing the diversity of the training data to keeps as much types of attack as possible, but even including the normal network traffic data in order to reach a more balanced dataset. This component plays a crucial role in the success of our strategy, as it needs to have a deep understanding of DDoS attacks. To achieve this goal, we plan to automate the process as much as possible. Subsequently, The GMM could be implemented to classify more than two classes, e.g. trying to classify each kind of DDoS attack or trying to model a multi-class problem where the classes could be different kinds of cyberattacks.

References

1. N. Hoque, D. K. Bhattacharyya and J. K. Kalita, "Botnet in DDoS Attacks: Trends and Challenges," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2242-2270, Fourthquarter 2015, doi: 10.1109/COMST.2015.2457491.
2. Meng Wang, Yiqin Lu, Jiancheng Qin, A dynamic MLP-based DDoS attack detection method using feature selection and feedback, Computers & Security, Volume 88, 2020, 101645, ISSN 0167-4048, https://doi.org/10.1016/j.cose.2019.101645. (https://www.sciencedirect.com/science/article/pii/S0167404819301890)
3. Nhu-Ngoc Dao, Junho Park, Minho Park and Sungrae Cho, "A feasible method to combat against DDoS attack in SDN network," 2015 International Conference on Information Networking (ICOIN), Cambodia, 2015, pp. 309-311, doi: 10.1109/ICOIN.2015.7057902.
4. S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," 2015 International Conference on Computing, Networking and Communications (ICNC), Garden Grove, CA, USA, 2015, pp. 77-81, doi: 10.1109/ICCNC.2015.7069319.
5. U. Dincalp, M. S. Güzel, O. Sevine, E. Bostanci and I. Askerzade, "Anomaly Based Distributed Denial of Service Attack Detection and Prevention with Machine Learning," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 2018, pp. 1-4, doi: 10.1109/ISMSIT.2018.8567252.
6. Aditya, A. K. E. L. L. A. (2003). Detecting DDoS Attacks on ISP Networks. In Proc. of ACM SIGMOD Workshop on Management and Processing of Data Streams, 2003.
7. Behal, Sunny & Arora, Ritika. (2010). IP Spoofing.
8. Bing Xue and Mengjie Zhang. 2017. Evolutionary feature manipulation in data mining/big data. SIGEVOlution 10, 1 (March 2017), 4–11. https://doi.org/10.1145/3089251.3089252
9. Ahsan, M. M., Mahmud, M. A. P., Saha, P. K., Gupta, K. D., & Siddique, Z. (2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. Technologies, 9(3). https://doi.org/10.3390/technologies9030052
10. Wernogtiug A. Bhandari, "Feature Scaling — Standardization Vs Normalization", Analytics Vidhya, 2020, [online] Available: https://www.analyticsvidhya.eom/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
11. Pargent, F., Pfisterer, F., Thomas, J. et al. Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. Comput Stat 37, 2671–2692 (2022). https://doi.org/10.1007/s00180-022-01207-6
12. B. Bhattacharya, D.P. Solomatine, Machine learning in sedimentation modelling, Neural Networks, Volume 19, Issue 2, 2006, Pages 208-214, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2006.01.007.
13. Salam, Mustafa Abdul, et al. "The effect of different dimensionality reduction techniques on machine learning overfitting problem." Int. J. Adv. Comput. Sci. Appl 12.4 (2021): 641-655.
14. A. Jović, K. Brkić and N. Bogunović, "A review of feature selection methods with applications," 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 2015, pp. 1200-1205, doi: 10.1109/MIPRO.2015.7160458.
15. Azhagusundari, B., and Antony Selvadoss Thanamani. "Feature selection based on information gain." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2.2 (2013): 18-21.

16. Felipe L. Gewers, Gustavo R. Ferreira, Henrique F. De Arruda, Filipi N. Silva, Cesar H. Comin, Diego R. Amancio, and Luciano Da F. Costa. 2021. Principal Component Analysis: A Natural Approach to Data Exploration. ACM Comput. Surv. 54, 4, Article 70 (May 2022), 34 pages. https://doi.org/10.1145/3447755
17. Genie-Networks. DDoS Attack Statistics and Trends Report for 2020. 2021, https://blog.cloudflare.com/ddos-threat-report-2022-q4/.
18. I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.
19. REYNOLDS, Douglas A., et al. Gaussian mixture models. Encyclopedia of biometrics, 2009, 741. 659-663.
20. B. Barazandeh and M. Razaviyayn, "ON THE BEHAVIOR OF THE EXPECTATION-MAXIMIZATION ALGORITHM FOR MIXTURE MODELS," 2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Anaheim, CA, USA, 2018, pp. 61-65, doi: 10.1109/GlobalSIP.2018.8646506.
21. Schwarz, Gideon. "Estimating the Dimension of a Model." The Annals of Statistics 6, no. 2 (1978): 461–64. http://www.jstor.org/stable/2958889.
22. NANNEN, Volker. The paradox of overfitting. 2003. PhD Thesis. Faculty of Science and Engineering.
23. Author, F.: Article title. Journal **2**(5), 99–110 (2016)
24. Siddharth Misra, Aditya Chakravarty, Pritesh Bhoumick, Chandra S. Rai, Chapter 2 - Unsupervised clustering methods for noninvasive characterization of fracture-induced geomechanical alterations, Editor(s): Siddharth Misra, Hao Li, Jiabo He, Machine Learning for Subsurface Characterization, 2020, Pages 39–64, https://doi.org/10.1016/B978-0-12-817736-5.00006-5