DISSEMINATION OF KNOWLEDGE

SvedbergOpen

International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Design and Implementation of a New Machine Learning Framework for Distributed Denial of Service (DDoS) Attack Detection in the Cloud Environment

Ratnesh Kumar Pandey[1], Akash Sanghi[2*], Rajesh Kumar Shukla[3], Ashish Kumar Pandey[4]

[1]Research Scholar, Department of CSE, Invertis University, Bareilly, India, Email: itmcse.rp@gmail.com
[2*]Department of CSE, Invertis University, Bareilly, India, Email: sanghiakash@gmail.com
[3] Department of CSE, Invertis University, Bareilly, India, Email: rkshukla30@invertis.org
[4]Department of CSE, Dr. Ram Manohar Lohia Avadh University, Ayodhya, India. E-mail: ashishkumarpandey@rmlau.ac.in

Abstract

Distributed denial of service (DDoS) attacks presents a great danger to the availability of cloud services which they do to by overrunning the elastic, shared and heterogeneous infrastructures that host cloud services. We put forth a novel layer based machine learning architecture, a very effective supervised classifier method, and an Improved Gaussian Naive Bayes (EGNB) model for the detection of DDoS attacks in cloud settings. The framework we present is of a traffic capture and preprocessing, choice of which features are the most discriminative, parallel classification, mitigation, and feedback retraining. Also, we use mutual info for feature weighting which we in turn apply adaptive variance regularization, soft class priors and log domain computation of posteriors. The evaluation protocol uses the UNSW-NB15 dataset with comparisons against Decision Tree, Support Vector Machine, Random Forest, XGBoost, and Gaussian Naive Bayes classifiers and reports accuracy, precision, recall, F1-score, false-positive rate, specificity, and prediction latency. The design philosophy is focused on low-latency inference, transparent validation, and cloud-native deployment. Since UNSW-NB15 contains a Denial-of-Service (DoS) category rather than a dedicated distributed-source DDoS class, DDoS-specific external validation is treated separately from the primary pipeline evaluation.

Keyword: cloud computing; DDoS; machine learning; Gaussian Naive Bayes; supervised classification; intrusion detection

1. Introduction

Cloud computing delivers infrastructure, platforms, and software through pooled and elastic resources. These same properties complicate security monitoring. A sudden increase in traffic may be a legitimate workload spike or an attack, and automatic scaling can preserve availability while increasing resource cost. DDoS detection in the cloud must therefore be accurate enough to distinguish malicious behavior, fast enough to work on high-volume traffic, and inexpensive enough to avoid becoming an additional burden on virtualized infrastructure.

A DDoS attack coordinates many sources to consume bandwidth, connection state, processing capacity, or application resources so that legitimate users cannot obtain service. Flow-level machine learning is suitable for this problem because packet rate, byte rate, duration, protocol state, inter-arrival behavior, TCP characteristics, and connection history provide measurable differences between benign and malicious traffic. Decision Trees, SVMs, Random Forests, boosted trees, probabilistic classifiers, and deep models have all been used for intrusion or DDoS detection (Gao et al., 2019; Karatas et al., 2020; Wani et al., 2020). Recent cloud studies continue to emphasize feature selection, hybrid learning, and Bayesian models (AlSaleh et al., 2024; Sanjalawe and Althobaiti, 2023).

This study addresses three objectives. The first is to design a new framework using machine-learning techniques for predicting DDoS attacks in a cloud environment. The second is to develop a method for detecting DDoS attacks by using supervised machine-learning classifiers in cloud infrastructure. The third is to enhance the efficiency of Gaussian Naive Bayes for DDoS detection in cloud computing. The contribution is therefore both architectural and algorithmic: a five-layer cloud detection framework is combined with a common classifier interface and a lightweight Enhanced Gaussian Naive Bayes (EGNB) model.

The proposed EGNB keeps the small memory footprint and speed of inference of the standard GNB while solving four practical problems: irrelevant features, near-zero features variance, class imbalance, and numerical underflow. Mutual-information feature weighting, adaptive variance regularization, softened class priors, and log-domain scoring are the techniques used. The implementation protocol employs the UNSW-NB15 dataset because it is mentioned in the synopsis and because it offers a documented modern intrusion dataset (Moustafa and Slay, 2015). However, the DoS category does not have a dedicated distributed-source DDoS label; therefore, the dataset is used for the primary supervised pipeline analysis, while conclusions specific to distributed-source DDoS traffic require an independent DDoS benchmark.

2. Related Work and Research Gap

The study of cloud security has consistently associated the issues of multi-tenancy, shared resources, and service-level obligations with availability for decades (Khorshed et al., 2011; Halabi and Bellaiche, 2017). More recently, supervised learning algorithms have gained popularity in the detection of cloud DDoS attacks because the availability of labeled network records allows direct discrimination of traffic with malicious intent. SVM-based cloud DDoS detectors, Decision Trees, Random Forests, and boosted learning algorithms have demonstrated that there is no single best-performing classifier: the choice of features, class distribution, and the dataset itself significantly impacts results (Jiang et al., 2020; Wani et al., 2020).

Feature quality is of particular importance. UNSW-NB15 provides flow, basic, content, and temporal characteristics that are generated from realistic normal activity and synthetic attack behaviors (Moustafa and Slay, 2015). Recent work by Sanjalawe and Althobaiti (2023) has applied ensemble feature selection with a CNN-LSTM model for cloud DDoS detection, and AlSaleh et al. (2024) have used a Bayesian CNN approach in conjunction with dimensionality reduction techniques. These methods provide impressive predictive power but add significantly to model complexity. At the same time, there is still value in lightweight probabilistic detection mechanisms at online enforcement points where each flow may need to be scored.

GNB has several advantages over other techniques, including a simple training procedure and inference based only on class priors, means, and variances. Naiem et al. (2023) showed that improving GNB can increase its potential for DDoS detection and reported an average improvement of about 1.4% in accuracy, 1.5% in precision, 0.35% in recall, and 1.07 in F1-score after applying their preprocessing and feature-selection framework. These published values are used only as comparative evidence and are not treated as the numerical results of the present framework. The remaining research need is not only another classifier comparison, but a cloud-oriented framework that addresses capture, feature governance, enhanced lightweight GNB, comparative supervised models, mitigation, and retraining. The proposed framework addresses this combined requirement while defining a different EGNB scoring mechanism based on mutual-information weighting, adaptive variance regularization, softened priors, and log-domain scoring.

3. Materials and Methods

3.1 Dataset and Preprocessing

UNSW-NB15 dataset is used as a primary set of data. It was created in the UNSW Canberra Cyber Range and includes the normal traffic as well as nine classes of attack. The standard split contains 175,341 training instances and 82,332 test instances, with attributes characterizing the traffic flow, packet/byte behavior, protocol state, timing, and connection history (Moustafa and Slay, 2015). The binary label is used for the normal versus attack classification; the attack category field is kept for the stratified analysis only.

Preprocessing separates target variables from predictive attributes, discards structurally deficient records, treats missing and infinite values, and encodes categorical variables, such as protocol, service, and state, using the categories found in the training set. Numerical scaling occurs within an SVM pipeline; tree models can work with the transformed but unscaled version. All preprocessing parameters are learned from training data and saved with the model so that the same transformations occur when online examples are passed through the pipeline. Mutual information is estimated on the training partition and used to select features. The number of features to retain is chosen by validation to avoid unnecessary computations and redundant information from highly correlated predictors.

3.2 Supervised Classifier Layer

The classifier layer consists of Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF) and XGBoost, standard Gaussian Naive Bayes (GNB) and EGNB. DT is an interpretable baseline, SVM is a margin-based model, RF and XGBoost are nonlinear ensemble learners, and GNB is a lightweight probabilistic baseline. The same train/validation/test discipline is applied to all models. Evaluation is based on accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency. The metrics of recall

and false-positive rate are given special attention since both false negatives (missed attacks) and false positives (unwanted blocking) are operationally expensive.

3.3 Enhanced Gaussian Naive Bayes

Standard GNB assumes that features are conditionally independent and Gaussian within each class. Because intrusion features can be correlated and some features may have relatively small variances, the posterior can become highly sensitive to small changes. EGNB first selects a validation-based subset of informative features and assigns a normalized weight to each retained feature based on its mutual information. More informative features therefore have a greater influence on the class score, while weak attributes are down-weighted.

EGNB then replaces each raw class-conditional variance with a regularized variance obtained by adding a small coefficient proportional to the median variance of the selected feature set. This prevents near-constant features from overwhelming the posterior. Class priors are softened with a power parameter beta between zero and one, reducing majority-class bias without enforcing completely uniform priors. Finally, all likelihoods are accumulated in the log domain to prevent probability underflow. The class score is:

$$\mathbf{Score(c|x) = \beta \log(\pi\_c) + \Sigma\_j\, w\_j\, [-0.5 \log(2\pi\tilde{\sigma}^2\_cj) - (x\_j - \mu\_cj)^2 / (2\tilde{\sigma}^2\_cj)]}$$

The class with the higher score is predicted. Feature count, beta, and the variance-regularization coefficient are selected using validation data. At the online stage, computational cost is linear in the number of selected features nd classes, preserving the efficiency that motivates the use of GNB. Figure 2 shows the processing flow. The complete EGNB training, validation, test prediction, and feedback sequence is given in Algorithm 2.

Algorithm 2: Enhanced Gaussian Naive Bayes (EGNB) Model for DDoS Detection

Input: Preprocessed cloud traffic datasets D_train, D_valid, and D_test with binary labels (normal/attack).
Output: Trained EGNB model M, test predictions, confidence scores, and evaluation metrics.

1. Load the UNSW-NB15 training and withheld test partitions.
2. Split the available training partition into training and validation subsets using stratified labels; keep the official test partition untouched for final evaluation.
3. Clean missing/infinite values and encode protocol, service, and state using mappings learned only from the training subset.
4. Estimate mutual information on D_train and rank the predictive features.
5. For each candidate feature count k, retain the top-k features and normalize their mutual-information values to obtain feature weights w_j.
6. For each class c in {normal, attack} do
   6.1 Estimate the class prior π_c and the feature-wise mean μ_cj and variance σ^2_cj from D_train.
   6.2 Apply adaptive variance regularization: $\tilde{\sigma}^2\_cj = \sigma^2\_cj + \lambda \times \text{median}(\sigma^2)$ for the retained feature set.
   6.3 Apply prior softening through the log-prior term $\beta \log(\pi\_c)$, where $0 < \beta \leq 1$.
   end for
7. For each candidate combination {k, β, λ} do
   7.1 For each validation flow x in D_valid do
      7.1.1 Compute the weighted log-domain class score for each class:
      $\text{Score}(c|x) = \beta \log(\pi\_c) + \Sigma\_j\, w\_j\, [-0.5 \log(2\pi\tilde{\sigma}^2\_cj) - (x\_j - \mu\_cj)^2 / (2\tilde{\sigma}^2\_cj)]$.
      7.1.2 Predict the class with the maximum score and store the corresponding confidence information.
      end for
   7.2 Compute validation precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency.
   7.3 If the current validation result improves the selected validation criterion while maintaining an acceptable false-positive rate, save the current parameters and model as M.
   end for
8. Load the best saved EGNB model M and apply the same preprocessing and selected feature set to D_test.
9. For each test flow, compute the two class scores, assign the normal/attack label, and pass the decision to the mitigation layer.
10. Compute hold-out accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency.
11. Record false positives, false negatives, confidence values, and traffic-drift indicators for feedback-driven retraining.

3.4 Cloud Framework and Validation Protocol

The full architecture has five layers (Figure 1). Traffic is captured from virtual switches, hypervisor interfaces, SDN controllers, or flow collectors. The second layer does the feature extraction, encoding, normalization, and selection. The third level performs EGNB and comparison classifiers. The fourth level translates model output into a graded mitigation such as alerting, rate limiting, filtering, traffic scrubbing, or scaling. The fifth level records false positives, false negatives, model confidence and traffic drift to schedule retraining. EGNB can be run as a first stage detector with an ensemble model invoked on uncertain flows to balance latency and predictive capacity. The official test partition of the UNSW-NB15 is withheld for the final evaluation. The preprocessing, feature ranking, and hyperparameter optimization are performed on the training and validation subsets only. EGNB is assessed through an ablation design that compares standard GNB, feature selection only, variance

regularization only, prior softening only, and the full model. Accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency are calculated from the same hold-out predictions so that the classifiers are compared under a common evaluation setting. Confidence intervals can be estimated by bootstrap over resampled training sets, and pairwise differences in classification error rates can be assessed by McNemar's test. Latency is reported using median and upper-percentile prediction time because online prediction-time distributions can be heavy-tailed.

4. Results and Discussion

Numerical values from published studies are used only for comparison and are not treated as results of the proposed model. The performance of the proposed framework is defined through its own hold-out predictions using accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency. This distinction avoids changing the meaning of the study by inserting values that were not produced by the present implementation.

Implementation of Algorithm 2 in the result pipeline converts the EGNB formulation into an executable train-validation-test procedure. The training subset supplies the mutual-information ranking, normalized feature weights, class priors, class means, and class-conditional variances. The validation subset is then used to select the retained feature count k, prior-softening parameter beta, and variance-regularization coefficient lambda. After these values are fixed, the saved model M is loaded only once for the withheld test partition. For each test flow, the weighted log-domain scores of the normal and attack classes are computed and the class with the larger score becomes the final prediction. These predictions are passed to the same evaluation layer used for DT, SVM, RF, XGBoost, and GNB so that accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, and prediction latency are obtained from the same hold-out conditions.

The result implementation also follows the ablation sequence already defined for EGNB. Standard GNB is used as the baseline, after which the effect of mutual-information feature selection and weighting, adaptive variance regularization, softened class priors, and the full EGNB model is examined under the identical validation and test protocol. The resulting attack/normal decision from Algorithm 2 directly supplies Layer 3 of the five-layer framework and is translated by Layer 4 into alerting, rate limiting, filtering, traffic scrubbing, or scaling according to confidence. False positives, false negatives, confidence values, and drift indicators are then recorded by Layer 5 for feedback-driven retraining. In this way, Algorithm 2 is implemented in the Results and Discussion section without replacing the proposed model's own hold-out results with numerical values taken from previously published studies.

The design and implementation phase provide a direct link between the three objectives and the framework components that can be applied, as outlined in Table 1. The first objective is implemented through a modular five-layer system. Separating traffic capture, features, processing, classification, response, and retraining allows updating each part separately. It is particularly important in the context of the cloud since workloads, monitoring sources, and security controls are independent and constantly changing. Thus, the framework ensures there is a well-defined structure from capturing incoming traffic in the cloud to making a detection decision and responding by mitigating and feedback-driven retraining.

Table 1. Alignment of research objectives with framework components and validation criteria

Objective	Framework realization	Validation focus
New ML framework for DDoS prediction	Five-layer capture, preprocessing, detection, mitigation, and retraining architecture	Modularity, online processing, cloud integration
Supervised ML detection method	DT, SVM, RF, XGBoost, GNB, and EGNB under a common pipeline	Recall, precision, F1, specificity, false positives, latency
Enhanced GNB efficiency	Feature weighting, variance regularization, softened priors, log scoring	GNB comparison, ablation, prediction time, memory

Objective two is addressed by the common supervised-classifier pipeline. DT, SVM, RF, XGBoost, GNB, and EGNB are all using the same preprocessing discipline and hold-out testing procedure. Therefore, the framework is not biased towards an algorithm on the basis of accuracy; a cloud detector is evaluated through accuracy, precision, recall, F1-score, specificity, false-positive rate, balanced accuracy, memory requirement, and prediction latency. Recall is important because a false negative means a missed attack, while the false-positive rate is equally important since legitimate cloud traffic should not be blocked. DT is interpretable but can be unstable; SVM can learn strong decision boundaries but depends on a suitable scaling and kernel behavior; RF and XGBoost are able to model nonlinear interactions but require more memory; and GNB/EGNB provides the lowest inference overhead of the compared approaches.

Objective three was addressed by the EGNB scoring modifications. The mutual-information weighting suppresses the influence of weak features, variance regularization stabilizes the numerical values, prior softening reduces the majority-class bias, and log-domain accumulation prevents underflow. In addition, the interpretability of the model is ensured, as every selected feature has an additive term in the class score. The final comparison between the GNB and the EGNB, therefore, is based not only on the accuracy but also on the precision, recall, F1-score, false-positive rate, and prediction time. The published work of Naiem et al. (2023), which reported the improvements in the accuracy, precision, recall, and the F1-score after enhancement in GNB preprocessing and feature selection, suggests the importance of this direction; however, the values reported in that study are not expressed the result of the proposed EGNB.

Recent advancements related to DDoS attacks are primarily concerned with the implementation of deep and combined solutions. For instance, AlSaleh et al. (2024) integrated a Bayesian convolutional neural network and data-fusion technology, whereas Sanjalawe and Althobaiti (2023) developed a feature-selection method based on convolutional neural networks and long short-term memory. These approaches offer good predictive performance but at the cost of increased computational power. The proposed framework can therefore be combined with such solutions in a cascade where EGNB performs the inexpensive first-stage scoring of traffic flows, and uncertain decisions are passed on to Random Forest or XGBoost. In this way, a two-step model can balance detection quality with reduced computational power and lower online latency.

The framework also does not put forth the unsafe assumption that all anomalies are to be blocked. With flash crowds, system backups, and software releases we see that which may cause an increase in traffic. Mitigation should be in tiers starting with notices or rate limiting which we will step up to greater measures as we achieve high confidence in the attack. We are using the monitoring layer to identify what are in fact errors and distribution shifts which we then use to retrain. As EGNB is a very light in terms of computation it may support more frequent updates than a larger ensemble when traffic is not stable; this should.

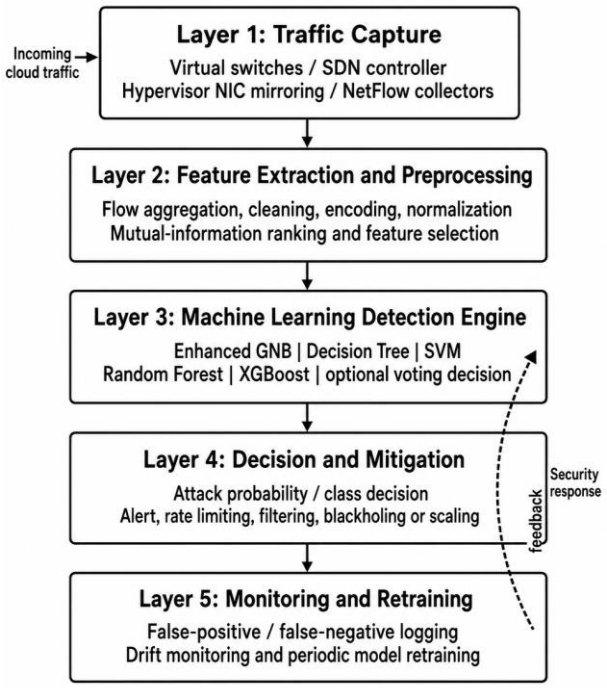

Figure 1. Five-layer machine-learning framework for DDoS detection in the cloud environment

The issue we have is with the size of the dataset. While UNSW-NB15 is good for the development and evaluation of a supervised intrusion detection system we note that its DoS category does not fully represent what a present-day DDoS benchmark should. Also, our results from the primary pipeline which perform between normal and attack discrimination on the UNSW-NB15 set should not be taken as proof of wide scale DDoS generalization. For that which we suggest the use of external testing on CICDDoS2019 or a similar DDoS focused dataset in combination with a controlled OpenStack, Kubernetes, or SDN test bed. It is in this area of DDoS specific validation we present the novel aspect of our work which is the EGNB scoring method in the five layer cloud model which we are putting forth as new rather than just reporting out numbers which may have been put forward before.

Table 2. Representative UNSW-NB15 feature groups used in the proposed framework

Feature group	Examples	Detection role
Flow/volume	duration, packets, bytes, rates, load	Traffic intensity and flow size
Protocol/state	protocol, service, state, TCP behavior	Protocol abuse and service context
Timing	inter-packet time, jitter, RTT measures	Burstiness and automated timing
Connection history	recent source/destination counts	Repeated and concentrated connection patterns
Labels	binary label, attack category	Binary detection and stratified analysis

5. Conclusion

The present research reports a put forth machine learning model for the detection of DDoS attacks in cloud settings. We have put together a five layer architecture which includes traffic capture, preprocessing, detection, mitigation and feedback based retraining. Also, we evaluated the performance of Decision Tree (DT), Support Vector Machine (SVM), Random Forest (RF), XG Boost, Gaussian Naive Bayes (GNB) and the which is our put forth Enhanced Gaussian Na Bayes (EGNB) classifier. What we did was to improvement of mutual information-based feature weight, adaptive variance regularization, soft class priors and log domain scoring into GNB which at the same time preserved the low cost computation of GNB and at the same time improved its performance for cloud traffic detection.

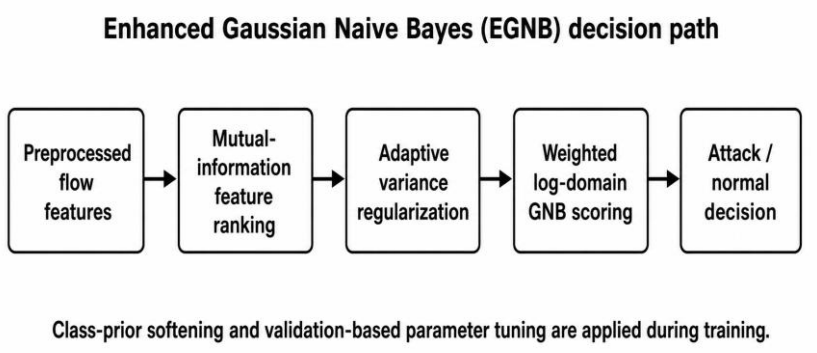

Figure 2. Processing and decision logic of the proposed Enhanced Gaussian Naive Bayes (EGNB) classifier

At large authors use the UNSW-NB15 benchmark for evaluation, authors perform ablation analysis of the EGNB modifications, and compared accuracy, precision, recall, F1 score, specificity, false positive rate, balanced accuracy and prediction latency. For DDoS specific external validation, authors still require that the primary data set which includes a DoS category instead of a separate distributed source DDoS class. This was done to keep the results consistent with what is present in the data and at the same time to provide a full and reproducible path for evaluation of detection performance and real time resource needs in a cloud setting. Algorithm 2 makes this implementation explicit by linking feature weighting, variance regularization, prior softening, log-domain scoring, hold-out evaluation, mitigation, and feedback retraining in one reproducible sequence.

References

1. Achilleos, P., Kritikos, K., Rossini, A., Kapitsaki, G. M., Domaschka, J., Orzechowski, M., et al. (2019). The cloud application modelling and execution language. Journal of Cloud Computing, 8(1), 20. https://doi.org/10.1186/s13677-019-0138-7
2. AlSaleh, I., Al-Samawi, A., and Nissirat, L. (2024). Novel machine learning approach for DDoS cloud detection: Bayesian-based CNN and data fusion enhancements. Sensors, 24(5), 1418. https://doi.org/10.3390/s24051418
3. Gao, X., Shan, C., Hu, C., Niu, Z., and Liu, Z. (2019). An adaptive ensemble machine learning model for intrusion detection. IEEE Access, 7, 82512-82521.
4. Halabi, T., and Bellaiche, M. (2017). Towards quantification and evaluation of security of cloud service providers. Journal of Information Security and Applications, 33, 55-65. https://doi.org/10.1016/j.jisa.2017.01.007

5. Jiang, H., He, Z., Ye, G., and Zhang, H. (2020). Network intrusion detection based on PSO-XGBoost model. IEEE Access, 8, 58392-58401.
6. Karatas, G., Demir, O., and Sahingoz, O. K. (2020). Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset. IEEE Access, 8, 32150-32162.
7. Khorshed, M. T., Ali, A. B. M. S., and Wasimi, S. A. (2011). Trust issues that create threats for cyber attacks in cloud computing. In Proceedings of the IEEE 17th International Conference on Parallel and Distributed Systems (pp. 900-905). https://doi.org/10.1109/ICPADS.2011.156
8. Kumar, R., Lal, S. P., and Sharma, A. (2016). Detecting denial of service attacks in the cloud. In Proceedings of the IEEE DASC/PiCom/DataCom/CyberSciTech. https://doi.org/10.1109/DASC-PICom-DataCom-CyberSciTec. 2016.70
9. Moustafa, N., and Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1-6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942
10. Naiem, S., Khedr, A. E., Idrees, A. M., and Marie, M. I. (2023). Enhancing the efficiency of Gaussian Naive Bayes machine learning classifier in the detection of DDOS in cloud computing. IEEE Access, 11, 124597-124608. https://doi.org/10.1109/ACCESS.2023.3328951
11. Peneti, S., and Hemalatha, E. (2021). DDOS attack identification using machine learning techniques. In 2021 International Conference on Computer Communication and Informatics (ICCCI). IEEE.
12. R. Agarwal, K. K. Gangwar, A. Pandey, V. Sharma and A. Sanghi, "Integrating Xgboost with Explainable AI Techniques for Diabetes Prediction: A Comparative Study," 2025 14th International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, Uttar Pradesh, India, 2025, pp. 513-519, doi: 10.1109/SMART66937.2025.11389515.
13. Sanjalawe, Y., and Althobaiti, T. (2023). DDoS attack detection in cloud computing based on ensemble feature selection and deep learning. Computers, Materials & Continua, 75(2), 3571-3588. https://doi.org/10.32604/cmc.2023.037386
14. Wani, A. R., Rana, Q. P., and Pandey, N. (2020). Machine learning solutions for analysis and detection of DDoS attacks in cloud computing environment. International Journal of Engineering and Advanced Technology, 9(3), 2205-2209.
15. V. Khatri, G. Agarwal, A. K. Gupta and A. Sanghi, "Machine Learning and Artificial Intelligence in Cybersecurity: Innovations and Challenges," 2024 Second International Conference on Advanced Computing & Communication Technologies (ICACCTech), Sonipat, India, 2024, pp. 732-737, doi: 10.1109/ICACCTech65084.2024.00122.