**Denial-of-Service Attack Detection over IPv6 Network Based on**

**KNN Algorithm**

**Methodology:**

* The experimental data set selected for DoS intrusion detection in the IPv6 network is derived from the 10% test set, a training set of the KDDCUP99 [29] data set; the normal type samples and the attack type samples related to DoS attacks are selected.
* The experimental data uses a 10% sample from KDDCUP99 dataset, containing normal and DoS attack-related samples.
* The number of discrete features in the data set is 9; the number of continuous features is 32.
* Data set has 9 discrete and 32 continuous features.
* To address limited attack samples, the study retains small groups, fine-tuning ratios and randomly selecting diverse sample sizes, while forming the test set by excluding absent types from the original training set, ensuring algorithm classification verification.
* Experiment divides into dual dimensionality reduction with information gain rates and evaluating GR-AD-KNN algorithm performance.
* The experiment is mainly divided into two parts:
  1. The initial step involves achieving dual dimensionality reduction and computing information gain rates for both primary and secondary features.
  2. The second part of the experiment is to evaluate the performance of the GR-AD-KNN algorithm.
* An average of ten experiments are recorded for optimization by comparing GR-KNN algorithm with the weighted optimization of the Euclidean distance and the GR-AD-KNN algorithm.
* Comparison between TAD-KNN and new GR-AD-KNN.
* Comparison between TAD-KNN based on average distance decision-making with the newly developed GR-AD-KNN.
* Calculation of F1-score:
* The experiment began by preprocessing data, reducing traffic features from 41D to 106D through one-level reduction, then further reducing sub feature dimensions for Service and Flag, and ultimately achieving a final 36D feature dimension for classification calculations.
* It sets up six horizontal control experiments with k, respectively [6, 8, 15, 28, 30, 31], and the experimental results. At the same time, before the algorithm performance comparison experiment starts, the data in the two data sets are normalized, respectively.
* Six control experiments are conducted with k values [6, 8, 15, 28, 30, 31], and data normalization is performed on two datasets before algorithm comparison.
* **Feature Selection and Dimensionality Reduction with Information Gain rate:** The researchers use the concept of "dual dimensionality reduction" to reduce the dimensionality of network traffic characteristics. They calculate the information gain rate to determine the importance of features and perform dimensionality reduction based on this criterion. The features are selected for dimensionality reduction to improve detection efficiency.
* **Optimizing KNN Algorithm for Intrusion Detection:** The team enhanced a commonly used classification method called K-Nearest Neighbors (KNN). They improved its accuracy by addressing issues when dealing with small groups of data and by considering the importance of different features in the classification process.
* **Developing GR-AD-KNN Algorithm:** A new algorithm, named GR-AD-KNN, was created. It combined the benefits of reducing dimensionality and optimizing KNN. This algorithm was designed to work better for detecting network attacks, especially in the context of IPv6 networks.
* **Evaluation Metrics:**

Used F1-Score for performance evaluation.

F1 − score = (Precision ∗ Recall/Precision + Recall)

**Results:**

* **Comparison of KNN Algorithms:** The researchers compared the traditional KNN with their newly developed GR-AD-KNN algorithm.
* **Algorithm Performance Evaluation:**

The GR-AD-KNN algorithm was compared to the GR-KNN algorithm with weighted optimization of Euclidean distance.

Performance was evaluated using the F1-Score as an indicator of detection performance.

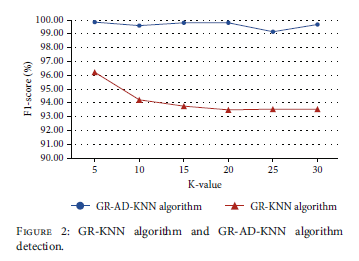


Figure 2 shows improved experimental results with the GR-AD-KNN algorithm, which is less sensitive to k values, reducing the need for precise selection and mitigating negative impacts from parameter adjustments.

A graph of different colored bars

Description automatically generated with medium confidence

This paper also compares the detection performance of traditional TAD-KNN with GR-AD-KNN algorithms; three independent rounds of ten experiments each are conducted at k = 5, resulting in average F1-Score for Teardrop attack type displayed in Figure 3.

Experimental comparison reveals that the GR-AD-KNN algorithm outperforms in detecting Teardrop attacks, highlighting the optimized algorithm's enhanced detection capability.

* **Improved DoS Attack Detection:** GR-AD-KNN performed better at identifying DoS attacks in network traffic.
* **Handling Small Groups:** The new algorithm effectively handled situations where there were only a few instances of a particular type of attack, improving the overall detection accuracy.
* **Consideration of Feature Importance:** GR-AD-KNN factored in the significance of different features during the classification process, leading to more accurate results.
* **Enhanced Detection Performance:** Overall, the proposed approach demonstrated higher accuracy in identifying specific attack types within IPv6 networks.

**Our idea:**

|  |  |  |
| --- | --- | --- |
| Aspect | Their Approach | Potential Improvements |
| Feature Selection and Dimensionality Reduction | Used “information gain rate” for selection and applied “dual dimensionality” for reduction | More advanced methods like recursive feature elimination, PCA, t-SNE could have been a better choice here. |
| Algorithm Modification | Introduced GR-AD-KNN with weighting | Explore other modified KNN variants like distance-weighted KNN or ensemble-based KNN |
| Interpretable Models | Focused on KNN-based methods | Consider interpretable models like decision trees or linear models. |
| Anomaly Detection Techniques | Focused on supervised methods | Look into unsupervised or semi-supervised anomaly detection |
| Evaluation Metrics | Used F1-Score for performance evaluation. | Include additional evaluation metrics like precision, recall, ROC curves, and AUC to provide a comprehensive view of algorithm performance. |