

An Uncertain Reasoning-Based Intrusion Detection System for DoS/DDoS Detection

Harpreet Singh¹, Habib Louafi²^a and Yiyu Yao¹^b

¹Department of Computer Science, University of Regina, Regina, SK, Canada

²Department of Science and Technology, TELUQ University, Montreal, QC, Canada
fi

Keywords: IDS, DoS, DDoS, Bayesian Networks, Markov Networks, Machine Learning, Artificial Intelligence.

Abstract: Network intrusion detection systems (NIDS) play an important role in cybersecurity, but they face obstacles such as unpredictability and computational complexity. To solve these challenges, we propose a novel probabilistic NIDS that detects DoS and DDoS attacks carried out on the TCP, UDP, and ICMP protocols. Our method incorporates knowledge from the fields of these protocols using Bayesian networks (BN) and Markov networks (MN). Inference is performed using Variable Elimination (VE) for BN and Shafer-Shenoy (SS) Propagation, as well as Lazy Propagation (LP) for MN. Extensive tests on the CAIDA dataset have yielded promising results, with higher Precision, Recall, and F1-Score metrics. Notably, both SS and LP are efficient, demonstrating the effectiveness of our proposed NIDS in improving network security.

1 INTRODUCTION

Computers and devices connected to the Internet use the Open Systems Interconnect (OSI) model to communicate with each other, whether the connection is wired (Zimmermann, 1980) or wireless (Korolkov and Kutsak, 2021). Each layer of the OSI model can be attacked differently by numerous types of attacks, such as Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, which are termed as the most catastrophic ones (Jaafar et al., 2019).


This paper focuses on securing network assets and safeguarding crucial data held on internet-connected devices and servers. DoS/DDoS attacks, which exploit basic protocols such as TCP, UDP, and ICMP with modifications, present significant issues due to their stealth and resource consumption. Intrusion Detection Systems (IDS) play an important role in minimizing such attacks by scanning network traffic for malicious activity. IDS has two detection methods: signature-based, which relies on known attack patterns, and anomaly-based, which uses models of typical behaviour to detect deviations. While signature-based detection is confined to known threats, anomaly-based detection provides greater coverage by identifying any aberrant net-


work behaviour for further investigation (Yassin et al., 2014).

In this paper, a behaviour-based IDS model is proposed for the detection of DoS/DDoS attacks, using uncertain reasoning in Artificial Intelligence (AI). Precisely, the proposed model uses Bayesian networks (BN), which are based on Probability theory and Uncertainty (Pearl, 1998). Without loss of generality, in this paper, we focus on the detection of DoS/DDoS attacks that exploit three widely used and targeted protocols, namely TCP, UDP, and ICMP. Thus, the knowledge representation of the BN is based on the fields that comprise these three protocols.

To achieve that goal, we propose a robust methodology for identifying malicious network traffic frames utilizing Bayesian network (BN) algorithms. Through rigorous implementation, testing, and validation, the paper evaluates the performance of three distinct algorithms: Variable Elimination (VE) for exact inference, Shafer-Shenoy propagation (SS) for message propagation, and Lazy Propagation (LP) for hybrid inference. These contributions collectively advance the state-of-the-art in intrusion detection systems and bolster network security measures.

The paper is organized as follows, Section 2, reviews the proposed solutions related to intrusion detection using AI algorithms. Section 3, presents the methodology of our proposed solution. Sections 4,

^a <https://orcid.org/0000-0002-3247-3115>

^b <https://orcid.org/0000-0001-6502-6226>

and 5 detail the experimental setup and results, respectively. Lastly, Section 6 concludes the paper.

2 RELATED WORK

In this section, we review the most important approaches, solutions, and frameworks proposed to design IDS using AI. We show, why probability-based approaches are preferred over fuzzy logic and rule-based methods in the design of IDS.

In various studies, different approaches have been proposed for detecting network intrusions and mitigating cyber threats. Bringas et al. (Bringas et al., 2008) introduced ESIDE-Debian, utilizing SNORT to collect labelled packet frames and constructing a Bayesian Network (BN) for anomaly and misuse detection. They employed the Lauritzen and Spiegelhalter (LS) propagation algorithm for inference, noting its efficiency in terms of response time. Sudar et al. (Sudar et al., 2021) proposed a method using Support Vector Machine (SVM) and Decision Tree (DT) algorithms for DDoS attack detection in software-defined networks (SDN), achieving 85% accuracy with SVM and 78% with DT. Alhakami et al. (Alhakami et al., 2019) presented a non-parametric Bayesian approach integrating feature selection mechanisms to detect both known and unknown attacks effectively. Alexander et al. (Alexander, 2020) introduced a Likert scale method combined with BN to reduce cyber threat attacks, advocating for strategic integration of BN models into organizational decision-making processes. Koc et al. (Koc and Carswell, 2015) explored various Bayesian classifiers, finding that Proportion K-Interval discretization combined with Hidden Naive Bayesian (HNB) yielded high accuracy in identifying DDoS assaults. (Agostinello et al., 2023) proposed a simulation of anomaly based IDS using Deep Learning (DL) methods. They worked on three different models (DNN, CNN, RNN). They showed that in binary classification, it is possible to yield high performance when we have more feature variables in the dataset. In there experiments the DNN model achieved the best performance.

In the comparison of various intrusion detection systems (IDS) based on their features, strengths, and weaknesses. The study by Bringas et al. integrates Snort and uses Bayesian Networks (BN) for supervised learning but requires extensive computational resources. Sudar et al.'s approach employs Mininet for simulation and Support Vector Machines (SVM) for detection, though it struggles with non-probabilistic classification and unclear mathematical explanations. Alhakami et al. focus on feature selec-

tion and parameter learning but lack clarity in their algorithm and anomaly classification. Alexander's model combines Likert scales with BN for decision-making, yet its accuracy is not evaluated. Koc and Carswell utilize NB and HNB classifiers for network attack classification, benefiting from conditional independence assumptions. However, it faces challenges with small datasets and feature dependencies. Lastly, Agostinello developed an Anomaly-Based IDS with DL techniques but these DL models are not explainable, as they are not based on probabilities.

3 METHODOLOGY

To implement our uncertain reasoning-based IDS, we propose a methodology, which is comprised of several modules, as shown in Figure 1. These modules are described in the following sections.

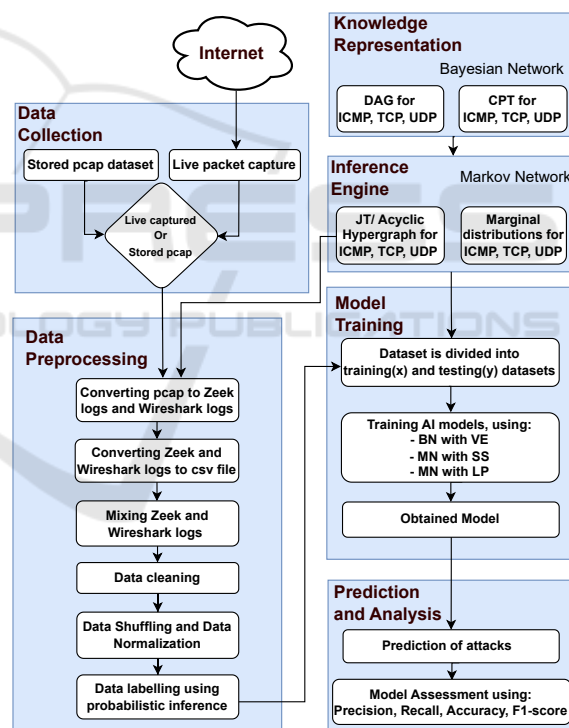


Figure 1: Proposed methodology flowchart.

3.1 Data Collection

In the first stage of our methodology, data needs to be collected ideally from real and live networks, using network traffic capture tools, such as Tshark or Wireshark (Orebaugh et al., 2006). Then, the captured traffic is usually stored in .pcap files. In this paper, we use the Caida dataset (Shirsath, 2023), which con-

tains approximately one hour of traffic traces related to DDoS attacks on the TCP, UDP, and ICMP protocols. The attack trace is split up into 5-minute pcap files. The schema of the dataset is shown in Table 1.

Table 1: Dataset Schema (Shirsath, 2023).

Attribute	Value
Data format	pcap
Total Number of Protocols	3 (TCP, UDP, ICMP)
Total Number of Features	64
Total Number of Instances	2,75,169
Data Size	18.3 MB
Start Date	Oct 24, 2022
End Date	April 15, 2023

The .pcap files are processed using Zeek (Tiwari et al., 2022), a network traffic analyzer, generating essential log files like *conn.log* and *weird.log*. While Zeek provides flow-level details, WireShark is simultaneously used to extract packet-level information from the same .pcap files, ensuring comprehensive traffic data collection.

3.2 Data Preprocessing

The second phase is data preprocessing, in which data mixing, data cleansing, normalization, labeling, and shuffling are achieved to prepare the dataset to be used by machine learning algorithms. The Zeek and WireShark logs are converted into CSV files, using pandas in Python. They are converted into data frames first for processing, then saved as CSV files.

3.2.1 Data Mixing

Since we have two CSV files, one from Zeek and the other one from WireShark, we mixed them into a single CSV file, which represents the dataset that we want to work on. In the mixing process, the source and destination IP addresses and their corresponding ports are used to match the samples from the two CSV files. The CSV files are iterated row by row and column by column to fill in the cells of the matched cases.

3.2.2 Data Cleansing

Cleaning datasets is critical because of potential discrepancies caused by network faults, device restrictions, and software problems. Several procedures are used to refine datasets, including expert-based feature selection to exclude extraneous variables required for Bayesian inference. In addition, instances representing victim host's replies to attacker hosts are eliminated, leaving just attacker source packets for

attack classification. Furthermore, removing not-a-number (NaN) fields provides compatibility with feature datatype criteria for machine learning algorithms.

3.2.3 Label Encoding, Data Shuffling, and Normalization

In this paper, the open-source *Scikit-Learn* library is used for processing the data. First, the *Label Encoder* is used to convert categorical data into numerical format, which is often required for machine learning models, including BN. Then, for data shuffling, the *Train Test Split* is used for splitting the dataset into train and test datasets. For normalization, we use the *MinMaxScaler* algorithm, which converts all the values into values between 0 and 1.

3.3 Knowledge Transformation

To capture the complex dependencies more effectively and potentially improving the computational efficiency, particularly for large datasets or complex structures, the BN representation is transformed into MN. Then, both transformations are used. Technically speaking, from the BN, the Directed Acyclic Graph (DAG) is converted into an Acyclic Hypergraph/Join Tree (JT) and the Conditional Probability Tables (CPTs) are converted into marginal distributions of the Joint Probability Distribution (JPD) for each join tree node.

3.4 Packet Classification (Labeling)

The classification of the packet frames as attack or legitimate is achieved by performing inference on the generated MN model. In this process, instances from the dataset are fed one by one to the MN model. Once the model is created, the feature values extracted from a particular instance from the dataset are used as evidence for computing the marginals on each node of the MN. This is achieved using the Shafer Shenoy Propagation algorithm (Lepar and Shenoy, 2013).

Each time a piece of evidence is obtained, its values are absorbed by the MN model, and hence the current JPD tables are updated.

3.5 Model Training

This section explains how the preprocessed dataset is used for training the prediction model, and which AI methods are used.

3.5.1 Algorithms

In this paper, the following algorithms are considered:

- Variable Elimination (VE).
- Shafer-Shenoy Propagation (SS).
- Lazy Propagation (LP).

3.5.2 Training and Testing

In this phase, various AI algorithms are trained and tested using ten-fold cross-validation. The pre-processed dataset is divided into training and testing subsets, with varying proportions (e.g., 10% training, 90% testing; 20% training, 80% testing). Results are averaged across iterations for each algorithm under consideration.

3.6 Prediction and Analysis

The trained models obtained from BN and MN are used for the prediction of DoS and DDoS attacks carried out on the TCP, UDP, and ICMP protocols.

Using the aforelisted AI algorithms (i.e., VE, SS, and LP), the inferences are achieved as follows:

In Bayesian Networks (BN), inference is performed using the Variable Elimination (VE) algorithm, which sequentially eliminates one variable at a time when new evidence is introduced. On the other hand, in Markov Networks (MN), inference is conducted using the Shafer-Shenoy (SS) and Lazy Propagation (LP) algorithms. SS is faster to implement but requires more storage space due to its use of two registers for incoming and outgoing messages. LP, however, is more efficient in terms of time and storage space utilization, making it suitable for scenarios where belief update efficiency is crucial (Madsen and Jensen, 2013).

4 EXPERIMENTAL SETUP

4.1 Setup

To validate our methodology, we ran a series of experiments on a MacBook Pro computer with an M1 processor, 8 cores (4 performance and 4 efficiency), and 8GB RAM. BN and MN were used to develop various models for ICMP, TCP, and UDP based on selected features. CPTs were generated from a raw dataset, with label-encoded and normalized data used as evidence to calculate probability of events. Predicted probability were used to classify packets as either attack or legitimate. The performance of the trained models was then evaluated. Wireshark and Zeek were used to record and analyze network data, while pyCharm and Jupyter notebook were used for

implementation and testing, together with a set of libraries that included Numpy, SKLearn, Pandas, Dask, Matplotlib, Tensorflow, Pgmpy, Applescript, Pyshark, Scapy, Zat.

5 EXPERIMENTAL RESULTS

To evaluate the performance of BN and MN models, we use three widely metrics, namely Precision, Recall, Accuracy, and F1-Score.

5.1 Precision Results

The Precision results for the ICMP, TCP, and UDP protocols, employing the VE, SS, and LP algorithms, are depicted in Figure 2. Notably, the MN model utilizing the SS algorithm achieves the highest Precision scores across all three protocols, yielding 97%, 99%, and 95% for ICMP, TCP, and UDP, respectively. The second-best results, with Precision scores of 98%, 99%, and 88% for ICMP, TCP, and UDP, respectively, are obtained using the MN model with the LP algorithm. The notable Precision achieved with the MN and SS algorithm underscores the algorithm’s efficacy in accurately estimating probabilities and making precise predictions. Conversely, lower Precision scores may indicate uncertainties or inaccuracies in the inference process, warranting further investigation. Nonetheless, the results obtained with the MN and LP algorithm closely align with those from the SS algorithm, suggesting proficient handling of updates and queries, ensuring accurate results.

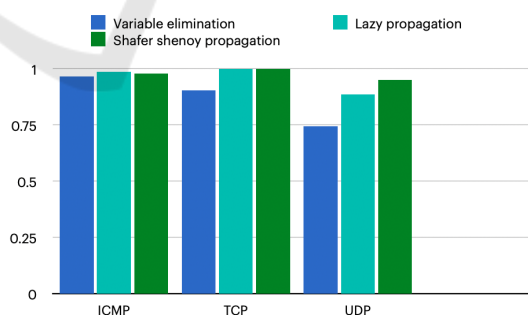


Figure 2: The Precision results, as obtained with the VE, SS, and LP algorithms.

5.2 Recall Results

The Recall results for the ICMP, TCP, and UDP protocols using the VE, SS, and LP algorithms are presented in Figure 3. The BN with the VE algorithm achieves 100% Recall for all three protocols, indicating its effectiveness in accurately identifying posi-

tive instances of the target variable. However, despite VE's exact inference capabilities, it is less efficient for large and complex models due to reasons like exponential complexity, sensitivity to variable ordering, space complexity, and lack of reusability. In contrast, the SS and LP algorithms yield similarly high Recall results, exceeding 96% for all protocols, demonstrating their effectiveness with minimal performance differences based on the dataset used.

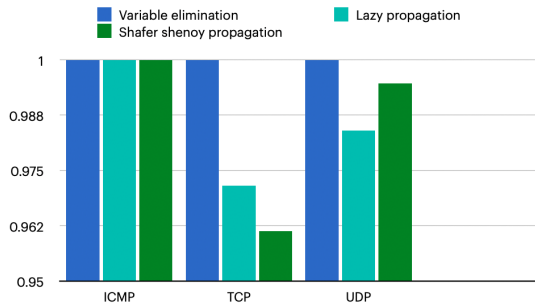


Figure 3: The Recall results, as obtained with the VE, SS, and LP algorithms.

5.3 Accuracy Results

The accuracy results for the VE, SS, and LP algorithms are provided in Figure 4. The MN utilizing the SS method works best for ICMP, TCP, and UDP, with accuracies of 97%, 96%, and 98%, respectively. The MN using the LP algorithm follows shortly behind. The SS and LP algorithms produced models with 18, 30, and 17 attributes for the three protocols, showing the same performance. Whereas the VE method, with a variable amount of attributes, affects its accuracy. As a result, the SS algorithm outperforms the VE and LP algorithms.

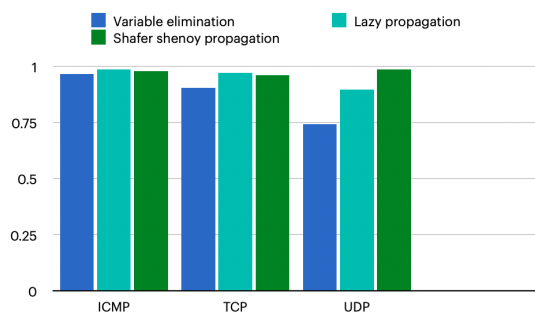


Figure 4: The Accuracy results, as obtained with the VE, SS, and LP algorithms.

5.4 F1-Score Results

The F1-Score results, illustrated in Figure 5, demonstrate that the MN with the LP algorithm produces the greatest performance, achieving 99%, 98%, and

93% for the ICMP, TCP, and UDP protocols, respectively. The VE algorithm performs slightly worse on the UDP protocol than on ICMP and TCP, and its F1-score results vary significantly between the three protocols.

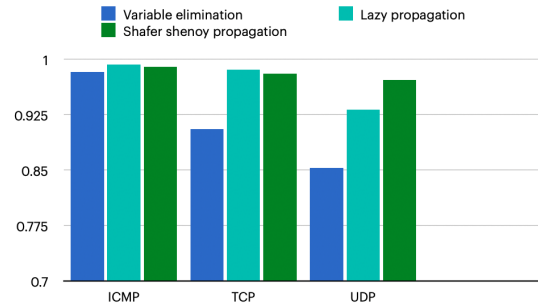


Figure 5: The F1-score results, as obtained with the VE, SS, and LP algorithms.

5.5 Complexity

The inference times for the ICMP, TCP, and UDP protocols using the VE, SS, and LP algorithms are recorded and averaged in Table 2, showing that the VE algorithm is the fastest, taking only 10.37 minutes for all three protocols. However, while VE is quicker, it is less efficient than the MN-based SS and LP algorithms, which, although slower, provide more stable, efficient, and accurate results. Once the models are generated, the MN-based algorithms have negligible prediction times, whereas the BN-based VE algorithm requires the same amount of time for each prediction as it does for initial model generation, making it less practical for frequent queries.

Table 2: Training Time (in minutes) of the inference algorithms for ICMP, TCP, and UDP.

Algorithm	ICMP	TCP	UDP	Total
VE	19.68	11.39	0.044	10.37
LP	128.77	170.07	0.064	99.63
SS	101.91	135.43	0.050	79.13

6 CONCLUSION

In this paper, we proposed a probabilistic-based NIDS for detecting DoS and DDoS attacks on TCP, UDP, and ICMP protocols, utilizing BN and MN models with inference implemented via VE, SS, and LP algorithms. VE was applied to BN, while SS and LP were used with MN. Experiments on the CAIDA dataset showed promising results across metrics like Precision, Recall, and F1-Score, revealing that both SS and LP are efficient, with LP slightly outperforming SS.

Despite LP's longer training time, its superior performance makes it the best algorithm based on our experiments. However, further validation on additional datasets is necessary to confirm these findings. Since the training is usually done offline (in the system's idle time), this does not represent an issue.

REFERENCES

- Agostinello, D., Genovese, A., Piuri, V., et al. (2023). Anomaly-based intrusion detection system for ddos attack with deep learning techniques. In *Proceedings of the 20th International Conference on Security and Cryptography. 1*, pages 267–275. SCITEPRESS.
- Alexander, R. (2020). Reducing Threats by Using Bayesian Networks to Prioritize and Combine Defense in Depth Security Measures. *Journal of Information Security*, 11(3):121–137.
- Alhakami, W., ALharbi, A., Bourouis, S., Alroobaea, R., and Bouguila, N. (2019). Network anomaly intrusion detection using a nonparametric Bayesian approach and feature selection. *IEEE Access*, 7:52181–52190.
- Bringas, P. G., Penya, Y. K., Paraboschi, S., and Salvaneschi, P. (2008). Bayesian-Networks-Based Misuse and Anomaly Prevention System. In *ICEIS (2)*, pages 62–69. Citeseer.
- Jaafar, G. A., Abdullah, S. M., and Ismail, S. A. (2019). Review of Recent Detection Methods for HTTP DDoS Attack. *Journal of Computer Networks and Communications*, 2019:1283472:1–1283472:10.
- Koc, L. and Carswell, A. D. (2015). Network intrusion detection using a HNB binary classifier. In *2015 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim)*, pages 81–85.
- Korolkov, R. Y. and Kutsak, S. (2021). Analysis of attacks in IEEE 802.11 networks at different levels of OSI model. *Natsional'nyi Hirnychiy Universytet. Naukovyi Visnyk*, (2):163–169.
- Lepar, V. and Shenoy, P. P. (2013). A comparison of Lauritzen-Spiegelhalter, Hugin, and Shenoy-Shafer architectures for computing marginals of probability distributions. *arXiv preprint arXiv:1301.7394*.
- Madsen, A. L. and Jensen, F. V. (2013). Lazy propagation in junction trees. *arXiv preprint arXiv:1301.7398*.
- Orebaugh, A., Ramirez, G., and Beale, J. (2006). *Wire-shark & Ethereal network protocol analyzer toolkit*. Syngress.
- Pearl, J. (1998). *Bayesian Networks*, page 149–153. MIT Press, Cambridge, MA, USA.
- Shirsath, V. (2023). CAIDA UCSD DDoS 2007 Attack Dataset. <https://dx.doi.org/10.21227/dvp9-s124>. Accessed: 2024-03-05.
- Sudar, K., Beulah, M., Deepalakshmi, P., Nagaraj, P., and Chinnasamy, P. (2021). Detection of distributed denial of service attacks in SDN using machine learning techniques. In *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–5.
- Tiwari, A., Saraswat, S., Dixit, U., and Pandey, S. (2022). Refinements In Zeek Intrusion Detection System. In *8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 974–979.
- Yassin, W., Udzir, N. I., Abdullah, A., Abdullah, M. T., Zulzalil, H., and Muda, Z. (2014). Signature-based anomaly intrusion detection using integrated data mining classifiers. In *2014 International Symposium on Biometrics and Security Technologies (ISBAST)*, pages 232–237.
- Zimmermann, H. (1980). OSI reference model-the ISO model of architecture for open systems interconnection. *IEEE Transactions on communications*, 28(4):425–432.