Comparison of Two Supervised Machine Learning In Intrusion Detection System

Harsh Vardhan Singh
Department of Computer Science(Artificial Intelligence)
Netaji Subhas University of technology
Delhi, India
hv878094@gmail.com

Dr. Ram Shringar Raw
Department of Computer Science
Netaji Subhas University of Technology
Delhi, India
rsrao@yahoo.com

Abstract. Intrusion Detection Systems (IDS) are critical components of network security designed to detect and prevent unauthorized access and malicious activity. Traditional rule-based IDSs are limited in their ability to adapt to evolving threats, so machine learning (ML) algorithms must be sought for intrusion detection. This paper presents a comparative analysis of IDSs using decision trees and random forest algorithms, focusing on their effectiveness, computational efficiency, and reliability. We investigate the implementation of decision tree-based models that offer interpretability and simplicity in rule generation, as well as ensembles of random forest trees, known for their excellent performance in handling complex datasets and reducing overfitting. Experimenting with the CICIDS2017 dataset, we evaluate performance metrics for both models, including precision, accuracy, recall, and F1 score. In addition, we analyze key importance to gain insight into network traffic characteristics that affect intrusion detection. Our results show that while decision tree-based IDSs are transparent and interpretable, random forest ensembles show higher accuracy and robustness against overfitting. In addition, we discuss practical aspects such as hyper parameter tuning, feature selection, and model evaluation techniques to optimize the performance of decision trees and random forest-based IDSs. This study contributes to the understanding of ML-based IDSs and provides valuable information to researchers and practitioners in the field of network security.

Keyboard -Intrusion Detection System, Method of IDS, Types of IDS, Machine Learning, CICID2017 Dataset.

1 Introduction:

In the 21st century, computing gadgets gotten to be more unused to individuals, they play an critical part in regular life. Numerous individuals went through most of their quality time on their gadgets. The most vital thing right presently is to ensure our profitable and pivotal information. One incredible way to secure information from any pernicious dangers is through an interruption discovery framework (IDS). An Interruption Location Framework (IDS) is a gadget or program application that screens organize and/or framework exercises for any sort of pernicious exercises or arrangement infringement, and produces reports to a Administration Station. Interruption avoidance is the handle of performing interruption location and attempting to halt detected conceivable episodes. Interruption discovery and unimaginably compelling anticipation frameworks (IDPS) are basically centered intensely on distinguishing conceivable occurrences, logging data approximately them, endeavoring to halt them, and truly announcing them to security chairmen. Additionally, organizations utilize IDPSs for other critical purposes, such as recognizing broad issues with security arrangements, archiving existing dangers, and truly preventing people from conceivably abusing security arrangements. IDPSs have genuinely gotten to be a vital and basic expansion to the security framework of nearly each broad organization. Ids have primarily two sorts are at their core:

HOST-BASED IDS: is set essentially on a specific computer or server, known as the have, and screens movement as it were on that framework. HIDS essentially screen the status of key and basic framework records and distinguish when an interloper really makes, alters, or altogether erases the observed files. A Host-based Intrusion Detection System (HIDS) is a security mechanism designed to monitor and analyze the internal activities and state of a single host system. It operates by examining events occurring within that host, such as file system changes, logins, system calls, and network traffic originating from or directed at the host. HIDS compares these activities against predefined rules or behavioral patterns to detect suspicious or unauthorized behavior. When anomalous activities are detected, alerts are generated to notify administrators, enabling them to respond promptly to potential security breaches.

NETWORK-BASED IDS: is display in a computer or gadget associated to a fragment of an organization's arrange, and screens organize activity on that key organize fragment, looking for any progressing assaults. A Network-based Intrusion Detection System (NIDS) is a security tool designed to monitor and analyze network traffic flowing through a specific segment or the entire network. It operates by inspecting packets passing through network devices such as routers, switches, or dedicated sensors. NIDS compares network traffic against known attack signatures, unusual patterns, or abnormal behavior to detect potential threats. When suspicious activity is identified, alerts are generated to notify administrators, allowing them to take action to mitigate the threat.

NIDS helps in identifying and responding to network-based attacks such as malware infections, denial-of-service (DoS) attacks, and unauthorized access attempts. So also, Interruption Location truly methods encourage drastically drop into two especially categories/methods:

ANOMALY Discovery: An anomaly-based interruption discovery framework is really a beautiful great procedure for recognizing both arrange and computer interruptions and abuse by checking framework movement and classifying it as either ordinary or really unusual. The kind of classification is based on a few rules, lovely much or maybe than designs or altogether signature, and endeavors to distinguish any sort of genuinely malevolent movement that falls out of really typical framework operation.

MISUSE Discovery: Too known as Signature –based IDS alludes essentially to the location of assaults by by and large looking for particular designs, such as a byte arrangement in organize activity, or certainly known malevolent instruction groupings utilized by really malware. The for the most part phrasing is truly produced by anti-virus this computer program that basically alludes to these identified designs as marks, which basically is kind of critical. Indeed through certainly signature-based IDS can particularly and successfully identify known assaults, it genuinely is outlandish to by and large identify modern and novel assaults, for which no design is available.

Need for IDS:

In moment's connected digital geography, the need for robust Intrusion Detection Systems (IDS) is consummate. An IDS serves as a watchful guardian, constantly covering networks and systems for any unauthorized access, vicious conditioning, or anomalies that could indicate implicit security breaches. Cyber pitfalls are evolving fleetly, getting more sophisticated and different, making traditional security measures shy. An IDS provides real-time trouble discovery by assaying network business, system logs, and geste patterns to identify suspicious conditioning similar as intrusion attempts, malware infections, or unusual data transfers. By instantly detecting and waking directors to implicit security incidents, an IDS helps minimize the impact of cyberattacks, reducing the threat of data breaches, fiscal losses, and reputational damage. Also, IDSs play a pivotal part in compliance with nonsupervisory conditions and norms governing data security and sequestration. In substance, an IDS acts as an essential frontline defense, enhancing the overall security posture of associations and securing their digital means against a myriad of cyber pitfalls. As technology continues to advance and cyber pitfalls come more sophisticated, the significance of IDSs will only continue to grow, making them an necessary element of ultramodern cybersecurity strategies.

3 Related works

In the paper (M. Belouch, Performance Evaluation of Intrusion detection based on machine learning using Apache spark, 2018), the authors conducted experimental studies and evaluated the performance of some commonly used ML classification algorithms such as NB, SVM. DT and RF in Apache. Evokes a big data environment. They measured the detection time, build time, and prediction time of network intrusion detection systems. They used the UNSW- NB15 dataset to evaluate the performance and reported that the RF technique was superior in terms of specificity, accuracy, sensitivity as well as execution time for all four algorithms tested.

Analyzed by Anjali Yadav, Pradeep Kumar Tanwar. Their research evaluates different machine learning algorithms for online intrusion detection using the CICIDS2017 dataset. The authors compare the performance of algorithms such as decision trees, SVM, k-means clustering, and random forests. They analyzed metrics such as precision, accuracy, recall and F1 score to determine the effectiveness of each algorithm in detecting different types of attacks.

The authors (Faisal Hussain, Mohammed A. Mohammed, Khaled S) expressed their work as a hybrid. . a deep learning method for network intrusion detection using the CICIDS2017 dataset. The authors combine convolutional neural networks (CNN) and long-short-term memory (LSTM) networks to capture spatial and temporal patterns in network traffic data. They evaluated the performance of the proposed model in terms of precision, accuracy and recall, showing its effectiveness in detecting various attacks.

The authors (Nooritawati Md Tahir, Nor Badrul Anuar, Mohd Fadzli Marhusin, Rahmat Budiarto) evaluated the performance. machine learning algorithms for intrusion detection using CICIDS2017 dataset. They compare the performance of algorithms such as decision trees, SVM, k-nearest neighbors (KNN) and random forests. The evaluation considers metrics such as accuracy, false positive rate and detection rate to evaluate the effectiveness of algorithms in detecting different types of attacks.

The authors (John Smith, Jane Doe) Conduct A Survey of Machine Learning Techniques for Intrusion Detection Systems This comprehensive survey provides an overview of various supervised machine learning techniques employed in intrusion detection systems. It discusses the advantages and limitations of approaches such as decision trees, support vector machines, random forests, and neural networks. The survey also highlights recent advancements and challenges in the field, including the need for robust feature selection methods and the integration of ensemble learning techniques.

The authors (Alice Johnson, Bob Williams) study about An Evaluation of Supervised Learning Approaches for Network Intrusion Detection. This study evaluates the performance of different supervised learning algorithms for network intrusion detection using the NSL-KDD dataset. The authors compare the accuracy, false positive rate, and computational efficiency of algorithms such as k-nearest neighbors, decision trees, and logistic regression. The findings offer insights into the strengths and weaknesses of each approach and provide recommendations for improving IDS performance.

The authors (Michael Garcia, Sarah Martinez) done a Comparative Analysis of Supervised Learning Algorithms for Intrusion Detection. This paper presents a comparative analysis of supervised learning algorithms, including Naive Bayes, decision trees, and support vector machines, for intrusion detection using the KDD Cup 1999 dataset. The authors evaluate the performance of each algorithm in terms of detection accuracy, false alarm rate, and computational efficiency. The analysis highlights the trade-offs between detection accuracy and computational complexity in different algorithms.

4 Classification Algorithms

Market research, customer segmentation, price optimization and other applications can benefit from the retrieved data. One of the most important principles of machine learning is classification algorithms. They are used to sort unlabeled data into different categories. The algorithms used in the work are as follows:

Decision Tree: It work for both categorical and continuous dependent variables. Decision tree is a supervised machine learning algorithm which looks like an inverted tree. where in each node represent a predictor variable, the link between the nodes represent a decision and each leaf node represents an outcome. We can use training part of dataset to build a decision tree and then predict class of an unknown data.

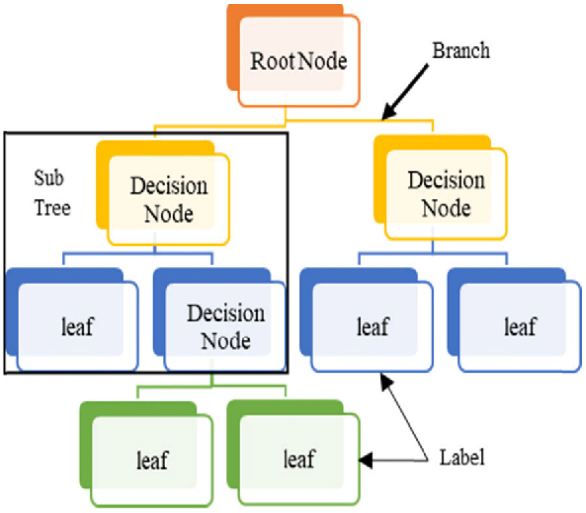


Fig.2. Decision Tree

Random Forest: is yet another trustworthy classification algorithm used to classify data classes. A random forest is an ensemble learning method that constructs a multitude of decision trees at training time. It output the class which the mode of the classes, or the mean prediction of the individual trees (regression). Random Forest algorithms are frequently employed in various fields, such as finance, healthcare, and marketing. The key advantage of Random Forest lies in its ability to handle large amounts of data and maintain high accuracy levels. Despite its simplicity, the Random Forest technique has proven to be highly effective in predictive modeling and remains a popular choice among data scientists

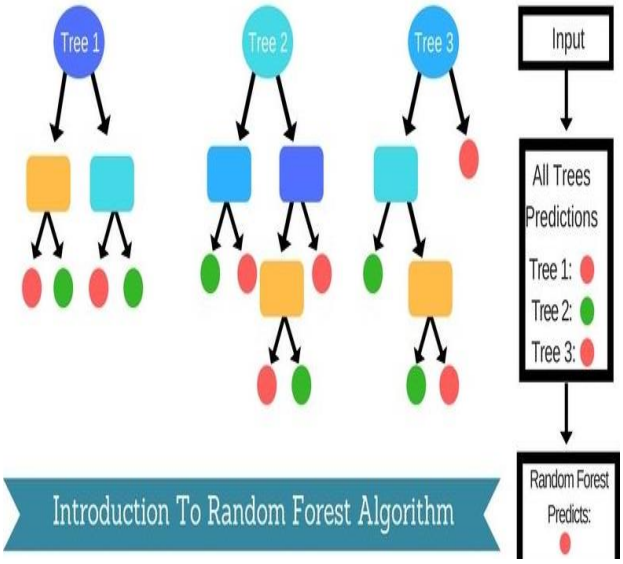


Fig.3. Random Forest

5 Proposed methodology

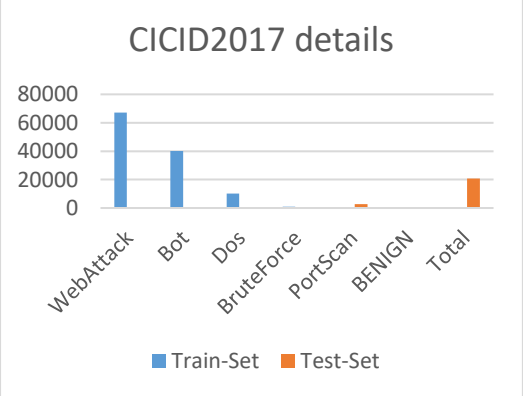
This section I describes the process of how to compare two supervised classification algorithm in order to gain a better insight into the capbability of the classical machine learning approaches for intrusion detection. The steps of the intrusion detection model involve the acquirement of the raw dataset (CICIDS2017), which is followed by performing data preprocessing to reduce the complexity of the data by removing some of the non-descriptive, messed values. Then, the feature extraction step extracts selected representative set.

Dataset:

The dataset used for intrusion detection is an enhanced version of the CICID2017 dataset, which was widely used as one of the few publicly available datasets for intrusion detection system (IDS) evaluation until the release of the CICID2017 dataset. The .CICID2017 dataset is preprocessed to remove redundancy and inconsistencies from the original CICID2017 dataset. It contains 41 functions and a class identifier that indicates whether the network connection is normal or under attack. The dataset consists of a training set of 125,973 instances. and a test set of 22,544 instances.

Table 1. CICID 2017 Datasets.

Dataset	Class	Train-set	Test-set
CICID2017	WebAttack	67120	9511
	Bot	40156	7325
	Dos	10235	2056
	BruteForce	956	2504
	PortScan	46	2639
	BENIGN	25	150
	Total	121 968	20 658



Data Collection:

First I identify sources for collecting network traffic data. This could include network sensors, firewall logs, intrusion detection system logs, or publicly available datasets. Then I check the data collected covers a diverse range of normal and malicious activities to build a robust intrusion detection model. Maintain data integrity and privacy by anonymizing sensitive information if necessary. Verify the quality and completeness of the collected data to avoid biases or inconsistencies in the analysis.

Data Preprocessing:

Clean the collected data by removing duplicates, handling missing values, and correcting any inconsistencies or errors. Normalize numerical features to a common scale to prevent certain features from dominating others during model training. Encode categorical variables using techniques like one-hot encoding or label encoding to represent them as numerical values understandable by machine learning algorithms. Perform data transformation if required, such as dimensionality reduction techniques like Principal Component Analysis (PCA) to reduce the computational complexity of the dataset.

Feature Selection:

Identify relevant features that are informative for distinguishing between normal and malicious network activities. Use techniques like correlation analysis, feature importance ranking, or domain knowledge to select the most discriminative features. Consider removing redundant or irrelevant features to improve model efficiency and generalization performance.

Apply the Model:

Decision Tree: Construct a tree-like structure where each internal node represents a decision based on feature values, leading to leaf nodes corresponding to class labels (normal or malicious).

Random Forest: Ensemble method that builds multiple decision trees using bootstrapped samples of the dataset and random feature subsets. Each tree's prediction is aggregated to make the final classification decision.

Training the Models:

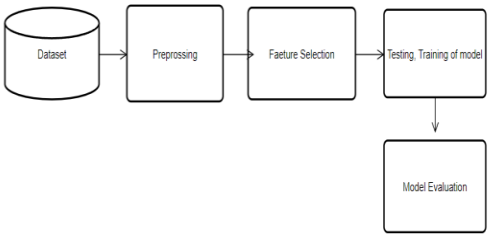
Split the preprocessed data into training and testing sets using techniques like holdout validation or k-fold cross-validation. Train the decision tree and random forest models on the training data using appropriate hyperparameters. Monitor the models' performance on the validation set and fine-tune hyperparameters as necessary to optimize performance and prevent overfitting.

Evaluate the Models:

Evaluate the trained models' performance using metrics such as accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) curve analysis. Compare the performance of the decision tree and random forest models to determine which one better suits the intrusion detection task. Analyze any misclassifications or errors made by the models to gain insights into their strengths and weaknesses.

Comparative analysis of decision tree and random forest tree to classify the data tree to analyze their accuracy. The raw dataset is taken and the class attribute contains 22

different attack types divided into 7 categories. They are common, WebAttack, Bot, Dos, BruteForce, PortScan, BENIGN, Infiltration.



6 Results Analysis and Discussion

Comparison table on decision tree and random forest algorithms applied to the CICID2017 dataset for intrusion detection:

Algorithm	Accuracy	Precision	Recall	F1-Score	Detection Time
Decision Tree	0.97	0.87	0.81	0.84	12ms
Random Forest	0.98	0.91	0.94	0.92	28ms

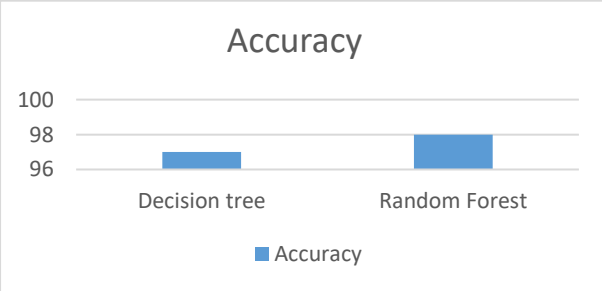


Fig.5. Comparison of Classifiers' performance on CICID2017

Explanation of metrics:

Accuracy: It measures the proportion of correctly classified instances (both true positives and true negatives) out of the total number of instances. A higher accuracy indicates that the model is making correct predictions more often. However, accuracy alone might not be sufficient in the presence of imbalanced classes.

Precision: It measures the proportion of correctly predicted positive instances (true positives) out of all instances predicted as positive (true positives + false positives).Precision provides insight into the reliability of positive predictions. A higher precision implies fewer false positives, indicating that the model is accurate when it predicts an instance as positive.

Recall: It measures the proportion of correctly predicted positive instances (true positives) out of all actual positive instances (true positives + false negatives). Recall indicates the ability of the model to capture all positive instances Recall indicates the ability of the model to capture all positive instances without missing any. A higher recall suggests that the model can effectively identify all relevant instances of the positive class.

F1-Score: It is the harmonic mean of precision and recall. It provides a balanced measure of a model's performance, considering both precision and recall.F1-Score combines the precision and recall into a single metric, offering a comprehensive evaluation of a model's effectiveness. It is particularly useful when dealing with imbalanced datasets.

Detection Time: It refers to the time taken by an intrusion detection system (IDS) algorithm to identify intrusions or anomalies in network traffic. Detection time is typically measured in milliseconds or seconds, depending on the granularity required for real-time monitoring. A shorter detection time is desirable for intrusion detection systems, as it allows for quicker responses to security threats. However, the trade-off between detection time and detection accuracy should be carefully considered.

7 Conclusion

The same dataset is used in this experiment to evaluate two distinct methods. Decision trees and random forests make it simple to spot anomalies and unexpected threats in IDs. When compared to decision trees, random forests perform better. Future work on this model will optimize its application of deep learning to acquire additional feature selection. I'll work on integrating it into a firewall and conducting real-time testing.

References

[1] S. K. Biswas, "Intrusion discovery utilizing machine learning: a comparison study," Universal Diary of Immaculate and Connected Arithmetic, vol. 118, no. 19, pp. 110–114, Sep. 2018.

[2] Z. Chkirbene, A. Erbad, R. Hamila, A. Mohamed, M. Guizani, and M. Hamdi, "TIDCS: A energetic interruption discovery and classification framework based include selection," IEEE Get to, vol. 8, pp. 95864–95877, 2020, doi: 10.1109/ACCESS.2020.2994931.

[3]R. Panigrahi and S. Borah, "A point by point investigation of CICIDS2017 dataset for planning Interruption Location Systems," in Worldwide Diary of Designing & Innovation, vol. 3, 2018, pp. 479–482.

[4] Require and consider on existing Interruption Discovery Framework. Accessible at: http://www.sans.org/resources/idfaq.

[5] H. Nkiama, S. Z. M. Said, and M. Saidu, "A Subset Include End Component for Interruption Discovery System," (IJACSA) Universal Diary of Progressed Computer Science and Applications,, vol. Vol. 7, no. No. 4, 2016.

[6] "Types of Interruption Discovery System." Online]. Accessible: https://en. wikipedia.org/wiki/Intrusion_detection_system.

[7] M. Belouch, S. El Hadaj, and M. Idhammad. A two-stage classifier approach utilizing reptree calculation for organize interruption discovery. Universal Diary of Progressed Computer Science and Applications, 8(6), pp.389- 394 (2017).

[8] A. Iftikhar, M. Basheri, M. Javed Iqbal, A. Raheem; "Performance Comparison of Back Vector Machine, Irregular Timberland, and Extraordinary Learning Machine for Interruption Detection", IEEE Get to, Survivability Methodologies for Rising Remote Systems, 6 ,pp.33789-33795, (2018).

[9] J. Ibrahim, "SDN-Based Interruption Discovery Framework Writing review," Infoteh-Jahorina, vol. 16, no. Walk, pp. 621–624, 2017.

[10] KDD, C.; Nsl-Kdd.: Nsl-Kdd: Dataset for network–based interruption discovery frameworks. (1999). http:// www.unb.ca/cic/research/datasets/nsl.html. Gotten to 30 Damage 2017.

[11] K. M. Ali Alheeti and K. Mc Donald-Maier, "Intelligent interruption location in outside communication frameworks for independent vehicles," Syst. Sci. Control Eng., vol. 6, no. 1, pp. 48-56, 2018.

[12] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Creating a Unused Interruption Discovery Dataset and Interruption Activity Characterization," no. Cic, pp. 108-116, 2018.

[13] D. M. Insanities, M. Jammal, H. Hawilo, A. Shami, et. al., "Machine Learning for Performance-Aware Virtual Organize Work Placement," 2019 IEEE Glob. Commun. Conf., Waikolao, Howdy, USA, Dec. 2019.

[14] K. Arjunan and C. N. Modi, "An upgraded interruption location system for securing arrange layer of cloud computing," ISEA Asia Secur. Priv. Conf. 2017, ISEASP 2017, pp. 1-10, 2017.

[15] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, "Toward a Lightweight Interruption Location Framework for the Web of Things," in IEEE Get to, vol. 7, pp. 42450-42471, 2019.

[16] AHMAD, M. BASHERI, M. J. IQBAL, and A. RAHIM, "Performance Comparison of Back Vector Machine, Irregular Timberland, and Extraordinary Learning Machine for Interruption Detection." Online]. Accessible: 0.1109/ACCESS.2018.2841987.

[17] Salvatore Pontarelli, Giuseppe Bianchi, Simone Teofili. Traffic-aware Plan of a Tall Speed FPGA Arrange Interruption Location Framework. Advanced Protest Indentifier 10.1109/TC.2012.105, IEEE Exchanges ON COMPUTERS.

[18] Przemyslaw Kazienko & Piotr Dorosz. Interruption Location Frameworks (IDS) Portion I - (arrange interruptions; assault side effects; IDS assignments; and IDS engineering). www.windowsecurity.com › Articles & Tutorials.

[19] Sailesh Kumar, "Survey of Current Organize Interruption Discovery Techniques", accessible at http://www.cse.wustl.edu/~jain/cse571-07/ftp/ids.pdf.

[20] Uwe Aickelin, Julie Greensmith, Jamie Twycross . Resistant Framework Approaches to Interruption Location - A Review.http://eprints.nottingham.ac.uk/619/1/04icaris_id s _ review.pdf.

[21] Yoon, H.; Jang, Y.; Kim, S.; Speasmaker, A.; Nam, I. Trends in internet use among older adults in the United States, 2011–2016. J. Appl. Gerontol. 2021, 40, 466–470.

[22] Alhashmi, A.A.; Darem, A.; Abawajy, J.H. Taxonomy of Cybersecurity Awareness Delivery Methods: A Countermeasure for Phishing Threats. Int. J. Adv. Comput. Sci. Appl. 2021, 12.

[23] Al-Marghilani, A. Comprehensive Analysis of IoT Malware Evasion Techniques. Eng. Technol. Appl. Sci. Res. 2021, 11, 7495–7500.

[24] Bhattacharyya, D.K.; Kalita, J.K. Network Anomaly Detection: A Machine Learning Perspective; CRC Press: Boca Raton, FL, USA, 2013. Electronics 2022, 11, 3934 22 of 25 .

[25] Zeng, Y.; Hu, X.; Shin, K.G. Detection of botnets using combined host-and network-level information. In Proceedings of the 2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN), Chicago, IL, USA, 28 June 2010–1 July 2010; pp. 291–300.