Faceted And Cross Network Significance Assessment Cross-Layer Exposure of The Power Communication System

A.R.Geebin[1], Mrs.P.Jenifer[2], Mrs.P.Brundha[3], Dr.E.Manohar[4]
[1]Dept of CSE
[2]Assistant Professor,Dept of CSE
[3, 4]Associate Professor,Dept of CSE
[1, 2, 3, 4] Francis Xavier Engineering College, Tirunelveli.

Abstract- *As the number of devices on the internet increases, the need to protect against intrusions becomes crucial. An Intrusion Detection System (IDS) distinguishes incoming malicious network data from benign data. Traditional signature-based IDS are vulnerable to novel attacks, bringing the need for anomaly-based IDS that use machine learning to detect newer attacks.*

The thesis aims to research anomaly-based IDS focusing on deep learning methods. Traditional deep learning approaches are compared with Generative Adversarial Networks (GAN) and adversarial machine learning approaches. The methods are evaluated using statistical measures on two different datasets. During the evaluation phase, adversarial samples are considered along with benign and known attack samples. Finally, the best approach is benchmarked against existing open-source anomaly-based IDS.

An approach employing a GAN to create adversarial samples performs better than all the other considered approaches. Additionally, the approach performs on par with existing anomaly-based IDS in the case of adversarial data points. We conclude that GAN-based approaches can be further developed to create intrusion detection systems that effectively defend against novel and adversarial attacks.

Keywords:Critical Infrastructure,Machine Learning, Intrusion Detection Systems.

I. INTRODUCTION

In light of the widespread use of the internet and the new applications emerging to meet the need, cyberattacks have increased. In the future, 5G will connect even more devices worldwide and a large proportion of those will be Internet of Things (IoT) devices. Though this will lead to more excellent connectivity, it will also allow attackers to exploit. Attackers can mount exploits by sending malicious data to devices connected to a network. This malicious data can then harm the device or hijack valuable information from the device itself. Networks are secured using various technologies like firewall, anti-virus, email filtering and virtual private network (VPN). Another commonly used approach is Intrusion Detection System (IDS). An Intrusion Detection System monitors the activity of a network and detects suspicious events. Network intrusion detection can be subdivided into two categories: signature- based and anomaly-based. Signature-based IDS relies on predefined rules and signatures of attacks to make classifications [1]. On the other hand, anomaly-based or heuristic-based IDS makes the classification by using machine learning to create profiles based on available data [2]. Since signature-based IDSs rely entirely on preset rules to make the predictions, they are vulnerable to new unknown attacks or zero-day attacks. Thus in the context of modern-day intrusion detection systems, anomaly-based IDSs make more sense. Recently, machine learning has made astonishing advancements in the fields of Healthcare, Autonomous driving, Fraud Detection, Personalization, Entertainment and Robotics. The cybersecurity domain has also benefited from this development. Deep learning, a branch of machine learning, is modeled after the human brain and performs better than traditional machine learning algorithms when there is a large amount of data to analyze. Datasets associated with intrusion detection usually contain large amounts of data. Also, deep learning is better at generalizing to new data and hence can detect newer attacks better. For these reasons, deep learning is the ideal solution for intrusion detection. Within deep learning, Generative Adversarial Networks or GANs have gained a lot of popularity in recent times. GANs are primarily known for generating fake images, but their architecture allows the framing of other problems. Anomaly-based IDS is the attention of a significant number of research initiatives. Furthermore, there are sizeable amount of open-source and proprietary IDS available. However, more focus is put on the performance of the model leading to below-par performance

when introduced to unseen attacks. Hence, this thesis aims to construct robust models, focusing primarily on GAN.

DEEP LEARNING-BASED IDS :

CNN Convolutional Neural Networks (CNN) are extensively used in the field of computer vision for their ability to extract features from images. CNNs are also used for intrusion detection by converting traffic data to images and then passing the images to the CNN model [3, 4]. Both 1D-CNN and 2D-CNN schemes are employed in the literature with varying results depending on the dataset type [1]. System calls are instructions sent to the operating system kernel by a user. CNNs can be used to analyze these system calls and detect intrusions [5]. As the number of layers increases in CNNs, the increasing depth leads to the vanishing gradient problem where the change to the weights of a network is insignificant. To tackle this issue, the residual neural network (ResNet) [6] model was introduced that employs a skip connection to bypass some layers in order to avoid dilution of the weights. ResNet is also widely used for classifying network traffic [7,8].

LSTM Long short-term memory [9] is a type of recurrent neural network (RNN), a group of neural networks that can recall information. However, unlike other RNNs, LSTMs can remember long-term dependencies better. LSTMs are particularly known for generating excellent results with time series data [10]. Loukas et al. [11] applied an LSTM model to predict intrusion detection in vehicles in real-time. To improve the performance of LSTMs, Maya et al. [12] introduced delayed LSTM, a model for anomaly detection in time-series data. This scheme benefits from the ability to choose the best model from an array of potential models.

Autoencoder Autoencoders are unsupervised learning algorithms that learn to recreate the input from an encoded representation of the input [13]. This ability makes autoencoders ideal for anomaly detection and dimensionality reduction. One technique is to create a new feature set using an autoencoder and then passing that set for classification using traditional machine learning algorithms [14]. Conventionally, network traffic data contains numerous features that can be reduced to improve intrusion detection performance, as demonstrated by Mighan et al. [15

ADVERSARIAL MACHINE LEARNING

Adversarial Machine Learning (AML) aims at corrupting a machine learning model to output wrong predictions. The corruption can be achieved by contaminating the model or altering the dataset [17]. Another taxonomy is based on the available information about the deployed model. In white-box attacks, the adversary is assumed to have all the information about the model. In contrast, an adversary has no information about a model in black-box attacks.

Adversarial examples can be generated through specific algorithms, which can then be utilized to detect intrusions in a system. Fast gradient sign method (FGSM) proposed by Goodfellow et al. [18] exploits a model's gradient to create adversarial samples that maximise its loss. The previous example is an evasion attack where the machine learning classification can be bypassed. Gu et al. [19] discussed the possibility of introducing a backdoor during training in a deep neural network. The backdoor will initiate an error when a certain situation is satisfied. An example of a black-box attack is the OnePixel attack [20], where a single pixel is changed to fool a model. This setup can deceive various machine learning models with little knowledge of the model itself. These are examples of a poisoning attack where the training data is adulterated with incorrect datapoints leading to imprecise model training.

Another potential backdoor attack is introduced by [21] et al., where poisoned data is forwarded to a model for training. The poisoned data is hard to identify from typical inspection and the trigger, i.e., what activates the backdoor, is hidden throughout training time. Poisoning attacks can cause immense damage for models that rely on incremental learning to improve performance. In incremental learning, the data obtained during model evaluation is used to further train a model. This is a perfect use case for unmanned vehicles where adversarial attacks have been demonstrated using a poisoning scheme [22]. The same concept can be replicated in other fields that employ incremental learning in their models.

INTRUSION DETECTION SYSTEM:

As a principle, an Intrusion Detection System (IDS) usually does not block network traffic. Rather IDSs flag suspicious traffic and record the data for further examination by experts. By experts, we refer to humans or systems with superior knowledge. In contrast, an Intrusion Prevention System (IPS) blocks all suspicious traffic that have been flagged. Most modern IDSs and IPSs still have a high false alarm rate and produce alerts for benign situations [29]. This could be an issue with IPSs as normal traffic will get blocked without any reason. However, IDS will simply forward or log the entries making IDSs the better solution. Speed is another parameter that also needs to be considered in this debate. Since IDSs will not block the traffic, they will be faster than IPSs. Furthermore, the IDS will be more effective if it is

implemented to process the data in parallel to the actual operation.

Network-based IDS

Network-based Intrusion Detection System or NIDS monitors all the traffic going through a network. So instead of putting an IDS for every computer in a network, a single entity is installed to do the job. This makes NIDS more efficient and less costly to install. A NIDS cross-checks the events with events from other systems and devices to flag potential risky traffic. Network-based IDS usually monitor and analyze network traffic to detect threats including Denial-of-Service (DoS) attacks, SQL injection attacks, and password attacks [30].

Anomaly-based IDS

Anomaly-based IDSs use machine learning to detect novel attacks that are not present in the data. This is unlike signature-based IDS that flag traffic based on existing attack data. They are thus susceptible to new attacks, including zero-day attacks.

Data and Datasets

The three main pillars of a machine learning project are Data, Loss and Model. First, we will be discussing the data types and sources. Krupski et al. [31] explains the different data types in IDS. The first widely used term is raw traffic which corresponds to all the traffic received at a single point. Grouping of raw network traffic based on similar properties is called Network Flow. The commonly considered features for network flow are the source,destination IP address and port number, along with the service type. Bidirectional flow or network session describes traffic flow between two devices in either direction. In session flow, the same properties are again grouped. The session is usually initiated through a three-way handshake. Another data format in the IDS space are features that describe the traffic [32]. The traffic features can refer to statistical features such as packet size, flow duration, etc.

GAN-BASED APPROACH

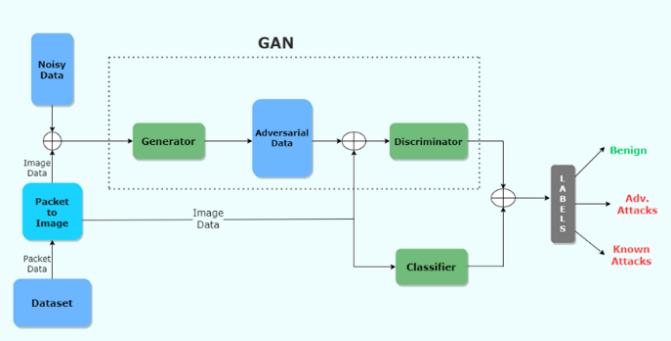
The second approach was based on an architecture comprised of a GAN network and a ResNet Classifier. This was the main approach that was focused on in this thesis. The classifier detects known attacks while the Discriminator identifies unknown attacks after training on adversarial samples.

During training, packet data from the datasets are converted to images using the data transformation scheme described earlier. the IGTD method was not considered in this approach since IGTD's performance was not satisfactory. Concurrently, random noisy data is created and concatenated with the image data. The concatenated data is then forwarded to the Generator part of the GAN network. Traditionally, the generator uses the input to create fake samples. In our case, the samples created by the generator are considered to be adversarial samples. Using this technique, the need for labeling adversarial data is avoided. The generated adversarial data is then passed to the Discriminator along with the initial image data. As the name suggests, the Discriminator distinguishes fake samples from real samples. In this scenario, the Discriminator separates the adversarial data from the benign data. Using GAN reduces the need for labeled data. The generator produces adversarial samples from the available data, removing the need for creating adversarial data samples manually.

As the GAN training continues, the generator gets better at creating adversarial data and the discriminator improves the classification of those adversarial samples. Usually, a trade-off is needed to determine when to stop the training process. However,

GAN Approach

since our primary goal is an efficient discriminator, the training was stopped when the discriminator loss plateaued. In other words, the generator loss was not considered when finding the best model combination.



Internally, GANs can be implemented with linear layers or with convolutional layers. We tried both approaches with a different number of layers. GAN architecture based on convolutional layers performs better than an architecture based on linear layers. A custom convolution GAN based on the DCGAN [39] architecture is shown in Figure6. This architecture produced the best results over all the combinations attempted in this approach. The generator had deconvolution layers that generated fake samples/images from

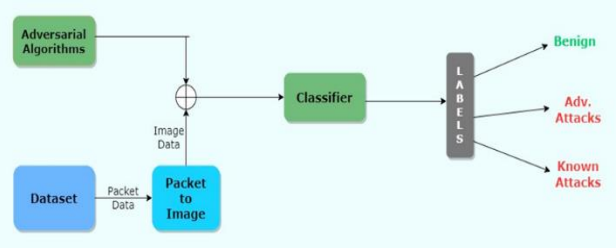
the input. The discriminator processes the fake and real samples using convolutional layers and forwards the output to a fully connected layer. The sigmoid layer then makes the final classification. As this is a binary classification of whether the discriminator input is adversarial or not, the sigmoid activation is utilized.

Simultaneously, a classifier was trained to detect known attacks from benign samples. The classifier was based on the ResNet architecture and the process was similar to the CNN ResNet approach described in Section4.2.1. The classifier inputs the image data created from the datasets and classifies benign or known attack classes.

After training, when evaluating single data points, the discriminator and the classifier would be used to determine the predicted class. Table3explains the potential scenario. Only if both discriminator and classifier categorize an input as benign, then the data point is considered to be benign. Conversely, if both outputs are anomalous, the predicted class is adversarial. This can be an example of a known attack being slightly perturbed to avoid detection.

ADVERSARIAL TRAINING:

The final approach is based on training a classifier on both adversarial data and conventional intrusion detection data. Only benign samples are selected from the dataset and combined with adversarial data that are labeled as malicious. The classifier is the same ResNet model introduced beforehand in other approaches. Combining several adversarial models to create samples is better than depending on a single algorithm. Hence, two different adversarial techniques were considered when generating the adversarial samples. The first adversarial algorithm was an evasion approach using the FGSM algorithm. The second approach is a poisoning approach utilizing the feature collision algorithm. Samples from both models were merged and processed. All similar data points were removed. Retraining the model on those samples can significantly improve the model's performance. However, this doesn't guarantee resistance from all adversarial samples. visually encompasses the



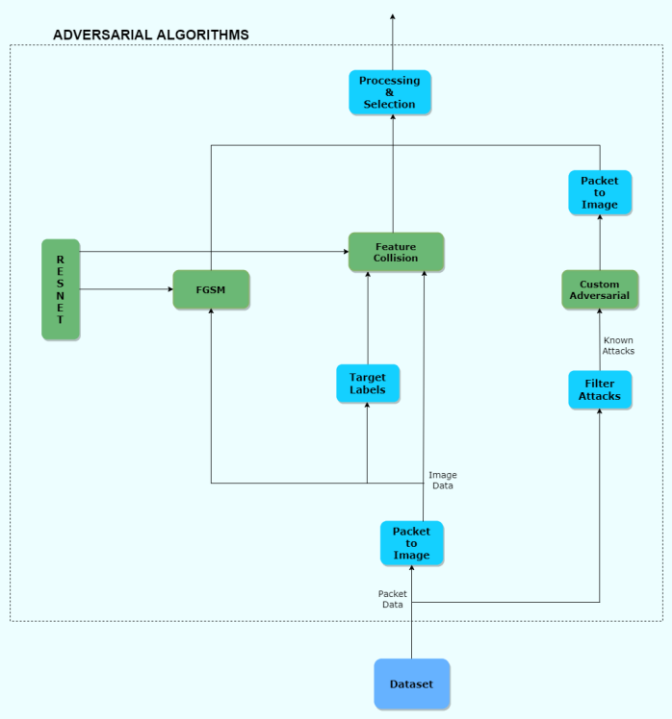
Adversarial Approach

The adversarial examples themselves were created from a combination of three different methods. The whole process of generating adversarial examples is illustrated in Figure8. The first method was the FGSM algorithm which represents an evasion attack. FGSM takes as input a base model and unadversarial samples from the dataset. The unadversarial samples are images converted from packet data. The second method to generate adversarial examples utilizes the feature collision method, which is a poisoning attack. The algorithm takes a base model and traffic data from the dataset as inputs. Additionally, the target input that will be wrongly classified at test time is also inputted. For both these cases, the base model was the best performing ResNet model from4.2.1. The third and final method was a custom adversarial sample generating scheme based on making slight changes to network data. The details of this scheme is described below. Firstly, raw packet data is filtered to only keep instances that represent known attacks. The packet data of known attacks are then passed to the adversarial generator. The generator outputs adversarial samples in network packet format, which are then converted to images. The adversarial examples from the three methods are concatenated together and processed. Duplicate data points are removed and the relevant ones are selected.

The adversarial examples themselves were created from a combination of three different methods. The whole process of generating adversarial examples is illustrated in Figure8. The first method was the FGSM algorithm which represents an evasion attack. FGSM takes as input a base model and adversarial samples from the dataset. The adversarial samples are images converted from packet data. The second method to generate adversarial examples utilizes the feature collision method, which is a poisoning attack. The algorithm takes a base model and traffic data from the dataset as inputs. Additionally, the target input that will be wrongly classified at test time is also inputted. For both these cases, the base model was the best performing ResNet model from4.2.1. The third and final method was a custom adversarial sample generating scheme based on making slight changes to network data. The details of this scheme is described below. Firstly, raw packet data is filtered to only keep instances that represent known attacks. The packet data of known attacks are then passed to the adversarial generator. The generator outputs adversarial samples in network packet format, which are then converted to images. The adversarial examples from the three methods are concatenated together and processed. Duplicate data points are removed and the relevant ones are selected.

Custom Adversarial Example Generator Adversarial examples can be generated from domain knowledge of network traffic and cybersecurity. We took packet data of known attacks and slightly altered the data to bypass the IDS.

The goal was to train the model on slightly perturbed data. Since the base data are instances of known attacks, small changes will not shift it back to a benign sample. The value of these fields can be slightly changed to create adversarial samples. An example of such a modification is illustrated in Figure9where the sequence number from the TCP layer is decreased by three. Similar changes are made to create more adversarial samples, which are then combined to create an array of custom-generated adversarial samples. The effectiveness of these samples for defense against adversarial attacks needs to be observed in detail.



Adversarial sample generation process in detail

II. RESULTS & EVALUATION

In this section we present the results pertaining to the approaches described in the section before. All the approaches were trained and tested on two datasets, the CIC-IDS 2018 and the ISOT-CID. For performance evaluation, accuracy and F1Score was calculated on the test set that was set apart.

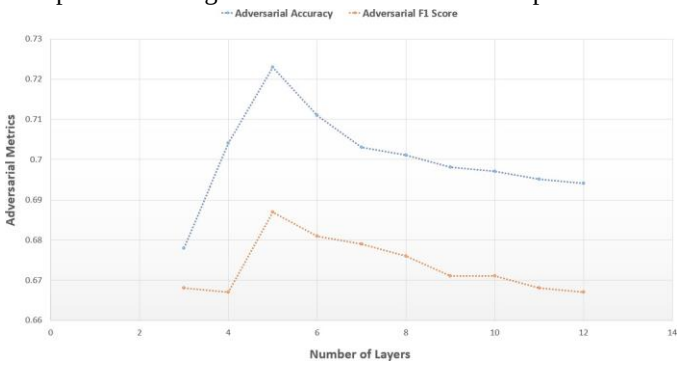
Data Transformation Schemes

In this thesis, both images and tabular data form the inputs for making predictions. Tabular data represents the statistical properties of raw network traffic data. For tab- ular data representation, the raw traffic packet is converted using the CICFlowMeter. This adds an extra overhead to the data transformation procedure. The overhead is tolerable if the training takes place directly on the tabular data. However, the network may also need image inputs, like in the case of GAN and CNN. The conversion of images from tabular data was performed using the IGTD algorithm. Then the computational burden of this algorithm also comes into the calculation.
Alternatively, the custom data transformation scheme introduced in Section4.2 can be used to transform packet network data to images directly. This avoids the extra overhead of transforming the data to and from tabular data. The conversion of data points took less time using the custom data transformation scheme. We came to this conclusion after averaging on 10000 data points over 10 iterations.

Basic Approach Results

For the basic approach, two different algorithmic processes were taken. The packet data in both datasets were converted to tabular format representing statistical features of network data. In the first process, the tabular data is fed directly to an MLP classifier to determine a baseline against which other approaches will be compared. In deep learning, a simple linear neural network, trained extensively, can outperform complex models and architectures. In order to identify whether that is the case, we have employed the basic approach. In the second approach, the tabular data is first converted to images and then passed to a CNN ResNet model. This approach helped us determine the efficacy of Convolutional architectures on network packet data. Additionally, we can also observe the effectiveness of the IGTD algorithm, which has been used to convert the tabular data to images.

The results of the approach on the CIC-IDS-2018 and the ISOT-CID dataset are visible in Table6and Table7respectively. Observing both tables, we can see that across both datasets, the MLP classifier outperforms the CNN ResNet model, even though the CNN ResNet model works with more parameters and complicated layers than the MLP classifier. We deduced that this poor performance was due to the IGTD algorithm not retaining enough useful information when converting to images. Due to this information loss, the IGTD algorithm was discarded for the next approaches and a new packet-to-image conversion method was adopted



DCGAN performance against number of CONV layers

III. CONCLUSIONS

With more gadgets connecting to the internet, it is essential to safeguard against invasions. In a network, incoming malicious network data can distinguished from legitimate data by an intrusion detection system (IDS). Traditional signature-based IDS are susceptible to new types of attacks, making anomaly-based IDS necessary. Anomaly-based IDS are implemented using machine learning making them more robust.

In this thesis, we conducted a comparative study of IDS solutions focusing on deep learning. Primary focus was on using Generative Adversarial Networks (GAN) to generate adversarial samples and train a network that can defend against both known and unknown attacks. The training of the models and the corresponding evaluation was executed on two different datasets. The CIC-IDS 2018 is a modern IDS dataset that is widely used in IDS research. The second was the ISOT-CID dataset collected in an actual cloud environment. The data contained benign, known attack and adversarial samples. Training on such varied data allowed the model to be robust and consequently be able to defend against novel attacks.

Across both datasets, the DCGAN approach provided the best results. This approach had a generator and discriminator implemented using the DCGAN architec- ture. The generator created adversarial samples from random noisy data and existing network data. The discriminator classified the adversarial samples while a CNN ResNet classifier distinguished known attacks from benign samples. The effectiveness of this model came from the duality of the two different methods working together. Benchmarking against open-source IDS validated that this idea is feasible. The next section talks about the limitations of the thesis and some of the future directions.

ACKNOWLEDGMENT

Financial support obtained from the All India Council for Technical Education (AICTE) under Research Promotion Scheme (RPS), Sanction order no: F.No 8.9/RIFD/RPS/Policy- 1/2017-18 coordinated by Anna University Recognized Research Centre, Department of Computer Science and Engineering, Francis Xavier Engineering College, Vannarpettai, Tirunelveli 627003, Tamilnadu, India.

REFERENCES

[1] Li, S.; Da Xu, L.; Zhao, S. The internet of things: A survey. Inf. Syst. Front. 2015, 17, 243–259. [Google Scholar] [CrossRef]

[2] Sun, N.; Zhang, J.; Rimba, P.; Gao, S.; Zhang, L.Y.; Xiang, Y. Data-driven cybersecurity incident prediction: A survey. IEEE Commun. Surv. Tutor. 2018, 21, 1744–1772. [Google Scholar] [CrossRef]

[3] McIntosh, T.; Jang-Jaccard, J.; Watters, P.; Susnjak, T. The inadequacy of entropy-based ransomware detection. In Proceedings of the International Conference on Neural Information Processing, Sydney, Australia, 12–15 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 181–189. [Google Scholar]

[4] Alazab, M.; Venkatraman, S.; Watters, P.; Alazab, M. Zero-day malware detection based on supervised learning algorithms of API call signatures. In Proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1–2 December 2011. [Google Scholar]

[5] Shaw, A. Data breach: From notification to prevention using PCI DSS. Colum. JL Soc. Probs. 2009, 43, 517. [Google Scholar]

[6] Gupta, B.B.; Tewari, A.; Jain, A.K.; Agrawal, D.P. Fighting against phishing attacks: State of the art and future challenges. Neural Comput. Appl. 2017, 28, 3629–3654. [Google Scholar] [CrossRef]

[7] Geer, D.; Jardine, E.; Leverett, E. On market concentration and cybersecurity risk. J. Cyber Policy 2020, 5, 9–29. [Google Scholar] [CrossRef]

[8] Buecker, A.; Borrett, M.; Lorenz, C.; Powers, C. Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security; International Technical Support Organization: Riyadh, Saudi Arabia, 2010. [Google Scholar]

[9] Fischer, E.A. Cybersecurity Issues and Challenges: In Brief; Library of Congress: Washington, DC, USA, 2014. [Google Scholar]

[10] Chernenko, E.; Demidov, O.; Lukyanov, F. Increasing International Cooperation in Cybersecurity and Adapting Cyber Norms; Council on Foreign Relations: New York, NY, USA, 2018. [Google Scholar]

[11] Papastergiou, S.; Mouratidis, H.; Kalogeraki, E.M. Cyber security incident handling, warning and response system for the european critical information infrastructures (cybersane). In Proceedings of the International Conference on Engineering Applications of Neural Networks, Crete, Greece, 24–26 May 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 476–487. [Google Scholar]

[12] O'Connell, M.E. Cyber security without cyber war. J. Confl. Secur. Law 2012, 17, 187–209. [Google Scholar] [CrossRef]
[13] Tolle, K.M.; Tansley, D.S.W.; Hey, A.J. The fourth paradigm: Data-intensive scientific discovery [point of view]. Proc. IEEE 2011, 99, 1334–1337. [Google Scholar] [CrossRef][Green Version]
[14] Benioff, M. Data, data everywhere: A special report on managing information (pp. 21–55). The Economist, 27 February 2010. [Google Scholar]
[15] Cost of Cyber Attacks vs. Cost of Cybersecurity in 2021|Sumo Logic. Available online: https://www.sumologic.com/blog/cost-of-cyber-attacks-vs-cost-of-cyber-security-in-2021/ (accessed on 10 May 2022).
[16] Anwar, S.; Mohamad Zain, J.; Zolkipli, M.F.; Inayat, Z.; Khan, S.; Anthony, B.; Chang, V. From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions. Algorithms 2017, 10, 39. [Google Scholar] [CrossRef][Green Version]
[17] Mohammadi, S.; Mirvaziri, H.; Ghazizadeh-Ahsaee, M.; Karimipour, H. Cyber intrusion detection by combined feature selection algorithm. J. Inf. Secur. Appl. 2019, 44, 80–88. [Google Scholar] [CrossRef]