A Rapid and Intelligent Statistical Mechanism for the Network Intrusion Detection System (NIDS) using ANN

Harshit Kandhwey[1] Sakshi[2] Rutvik Mangrole[3]
[1,2,3]Department of Computer Science and Engineering
[1,2,3]SRM Institute of Science and Technology, Ramapuram, Chennai,, India

Abstract — As computer networks become increasingly essential to daily life, the importance of their security is becoming more and more evident. Unfortunately, attackers are constantly designing new intrusion attacks, making it essential to be able to detect intrusions even with limited labeled data. To address this issue, we have developed an intrusion detection system based on a deep neural network using self-supervised contrastive learning. This approach allows us to use a vast amount of unlabeled data to create informative representations that can be used for various tasks, even with a limited amount of labeled data. Our paper proposes a deep learning architecture that employs a fully connected feed-forward Artificial Neural Network (ANN) for accurate intrusion detection prediction. We trained and evaluated the feed-forward ANN model using potential features.

Keywords: Intelligent Statistical Mechanism, Network Intrusion Detection System (NIDS), Artificial Neural Network (ANN)

I. INTRODUCTION

An intrusion detection system (IDS) is a security tool that tracks network activity and system events to look for suspicious activity or attempts at unauthorized access. An IDS's main job is to spot potential security breaches and notify the system administrator so they can act right away to reduce the threat. IDSs can be host-based or network-based, and they can detect malicious activity using a variety of techniques like anomaly detection, behavior analysis, or signature-based detection. IDSs are now a crucial part of any organization's security strategy due to the rise in cyber attacks and data breaches. IDSs are essential for securing sensitive data, defending networks, and preserving the integrity and availability of systems in this era of ongoing threat.

With the rise in cyber attacks and data breaches, intrusion detection systems (IDSs) have grown in significance. These assaults have the potential to significantly affect people, governments, and businesses. For instance, a successful cyber attack may cause the loss of confidential information, monetary loss, and reputational harm. Additionally, system failures and operational disruptions brought on by cyber attacks have the potential to have far-reaching effects.

IDSs employ a number of techniques to locate and recognize potential security risks. Signature-based detection, which compares network traffic or system events to a database of known attack signatures, is one of the most frequently used techniques. If a match is discovered, the IDS will send out a warning about a possible security breech.

Anomaly detection, which involves spotting suspicious or unusual activity that deviates from customary behavior, is another method used by IDSs. This method, which doesn't rely on pre-defined signatures, can be helpful in identifying previously unidentified attacks.

IDSs use behavior analysis to find unusual behavior patterns that might point to a security breach. This strategy is frequently combined with other methods to offer a more thorough picture of network and system activity.

IDSs can be set up in a number of different ways, including as a standalone device, a program running on a server or network device, or a component of a larger security system. The organization's needs and the required level of security will determine the deployment strategy.

In conclusion, an IDS is a crucial part of the security infrastructure of any organization. IDSs assist organizations in quickly responding to and mitigating potential threats by spotting and alerting the system administrator of potential security breaches. This lowers the risk of data breaches, monetary losses, and reputational damage. IDSs will continue to be essential in preserving the security and integrity of systems and networks as the threat landscape changes.

II. RELATED WORKS

In order to detect and respond to security threats more effectively and efficiently, intrusion detection systems (IDSs) are the subject of research. This entails creating new detection methods, enhancing the precision of current methods, and lowering the false positive rate. Researchers also work on enhancing IDSs' capacity to handle heavy network traffic and their capacity to detect previously unidentified attacks.

Artificial intelligence and machine learning are being used more frequently in IDS research, including the creation of automated threat response systems. Blockchain technology is another option being looked into by researchers to increase IDS security.

IDS deployment, configuration, and management are also being studied to make sure they are successfully incorporated into an organization's security infrastructure. This entails looking into ways to enhance IDS functionality and simplify their deployment and management.

Research on IDSs is generally focused on ensuring that they continue to be useful and effective in the face of a constantly changing threat environment, guarding systems and networks against potential security breaches.

The development of new detection methods, enhancing the precision of current methods, and lowering the false positive rate are just a few of the many topics that are covered in research on intrusion detection systems (IDSs). The detection of previously unidentified attacks by IDSs is one area of research that is being worked on. Anomaly detection techniques must be developed in order to spot unusual or suspicious activity that deviates from norms. Since these methods don't rely on pre-established signatures, they can be used to identify attacks that were previously undetected.

All rights reserved by *www.ijsrd.com*

In IDS research, the use of machine learning and artificial intelligence (AI) is growing. By learning from previous data and spotting patterns that are hard for humans to notice, machine learning techniques like neural networks can be used to increase the accuracy of IDSs. IDSs can respond to potential security breaches more quickly and effectively by automating the threat response process with AI.

Blockchain technology is another option being looked into by researchers to increase IDS security. A decentralized, tamper-proof log of IDS events can be made using blockchain technology, offering a safe and open method for data storage and sharing. IDSs may become more dependable and transparent as a result, earning the trust of stakeholders and organizations.

IDS deployment, configuration, and management are also the subject of research. This entails looking into ways to enhance IDS functionality and simplify their deployment and management. To build a more complete and potent security infrastructure, researchers are also looking into integrating IDSs with other security systems like firewalls and antivirus software.

In light of the constantly changing threat landscape, research on IDSs is crucial to ensuring their effectiveness and relevance. Researchers can contribute to the defense of systems and networks against potential security breaches by creating new detection techniques, enhancing the precision of current techniques, and lowering the false positive rate. Additionally, researchers can make sure that IDSs are successfully incorporated into an organization's security infrastructure, providing a strong and thorough defense against cyberattacks, by investigating new deployment and management strategies.

IDS evaluation and benchmarking is a key focus of additional crucial IDS research. The benchmarking of IDSs offers a comparative analysis of various IDSs in terms of their performance, accuracy, and other metrics. Evaluating IDSs is essential to determining their effectiveness. The KDD Cup 99 dataset and the Common Intrusion Detection Framework (CIDF) metrics are just a couple of the datasets and metrics that researchers have created for evaluating IDSs.

Additionally, researchers are looking into ways to enhance IDSs' capacity to manage heavy network traffic. IDSs must be able to handle large amounts of data while maintaining high detection accuracy in order to keep up with the growth of network traffic and the complexity of cyberattacks. Utilizing distributed IDSs, which can process network traffic across multiple nodes or devices, is one strategy. In order to speed up the processing of IDSs, researchers are also investigating the use of hardware acceleration, such as graphics processing units (GPUs).

IDS research also extends outside of conventional IT networks. IDSs must be modified to work in these new environments as Internet of Things (IoT) devices and cyber-physical systems (CPSs) become more prevalent. IDSs that can detect attacks on IoT and CPS devices as well as attacks on the communication networks that connect them are currently being developed by researchers.

In conclusion, IDS research is a multifaceted field that examines a wide range of subjects, including new detection methods and deployment and management techniques. The accuracy, performance, and scalability of IDSs can be improved by researchers, which will benefit businesses by enhancing the security of their networks and computer systems. Additionally, researchers can ensure that IDSs remain efficient and relevant in the face of emerging threats by adapting IDSs to new environments like IoT and CPS.

III. Feature Selection and Data Preprocessing

Building effective Intrusion Detection Systems (IDSs) requires feature selection and data preprocessing. These actions are designed to narrow down the set of features that are most important and simplify the data, improving detection precision and lowering false positive rates.

The process of feature selection entails determining which features are most crucial for detecting network attacks. This is crucial because IDSs may struggle to sort through mountains of data and find the information that matters most. By reducing the dimensionality of the data and increasing the effectiveness of the detection process, feature selection techniques aid in the identification of the most pertinent features.

Filter methods, wrapper methods, and embedded methods are a few of the feature selection techniques. Filter methods rank the features in accordance with their applicability to the target variable using statistical measures. Wrapper methods evaluate feature subsets using a search algorithm, whereas embedded methods use feature selection as a step in the model building process.

To enhance the quality and simplify the data, preprocessing involves cleaning and transforming the data. This entails codifying categorical variables, scaling the data, and removing missing values. By removing noise and unimportant information from the data, data preprocessing can increase the detection process's accuracy.

Data preprocessing techniques range from normalization to discretization to dimensionality reduction. While discretization involves transforming continuous variables into discrete values, normalization entails scaling the data to a common range. Principal Component Analysis (PCA), for example, uses dimensionality reduction techniques to reduce the number of variables in the data while maintaining the most crucial information.

In conclusion, choosing features and preprocessing data are essential steps in creating efficient IDSs. The most pertinent information is identified with the aid of feature selection, which also lessens the complexity of the data and increases the effectiveness of the detection process. By removing noise and unimportant information, data preprocessing enhances the quality of the data, resulting in increased detection accuracy and decreased false positive rates. IDSs can better identify and address potential security threats by utilizing these techniques, giving organizations increased security against cyberattacks.

A. Feature Selection

Building effective Intrusion Detection Systems (IDSs) requires careful consideration of feature selection. It entails determining the features that are most crucial for detecting network attacks, bringing down the complexity of the data, and enhancing the effectiveness of the detection process.

All rights reserved by www.ijsrd.com

In IDSs, there are various methods for feature selection, such as:

1) Filter methods: Filter methods rank the features in accordance with their applicability to the target variable using statistical measures. The correlation coefficient, mutual information, and chi-squared test are the three statistical variables most frequently used in filter methods. The features with the highest scores for relevance are kept, and those with lower scores are removed.
2) Wrapper methods: Wrapper methods assess subsets of features using a search algorithm. This entails choosing subsets of features iteratively and assessing the IDS's performance using the chosen subset. The subset of features with the best performance is chosen by the algorithm.
3) Embedded methods: Feature selection is incorporated into embedded methods as a step in the creation of the model. This entails choosing the features that have the greatest impact on the model's performance during training. Support vector machines and decision tree-based algorithms are two examples of embedded techniques.

The specific requirements and characteristics of the IDS being developed determine the feature selection technique to use. To achieve better results, it is typically advised to combine several techniques.

Feature selection not only increases detection efficiency but also lowers the risk of over fitting, which happens when an IDS is trained on too many features and performs poorly on new data. An IDS can increase its accuracy in detecting network attacks by choosing the most pertinent features, leading to improved security against cyber threats.

The selection of features is an essential step in creating intrusion detection systems (IDSs), as it aids in determining which features are most crucial for the detection of network attacks. Techniques for feature selection can assist in reducing the number of features used in the IDS, which can increase detection accuracy, lessen the chance of over fitting, and lower the IDS's computational cost.

Since filter methods are quick and demand little in the way of computational power, they are frequently used in IDSs. These techniques rely on statistical measures that rank the features in relation to how important they are to the desired variable. Correlation coefficient, mutual information, and chi-squared test are a few of the frequently employed statistical measures in filter methods. The features with the highest scores for relevance are kept, and those with lower scores are removed.

Since wrapper methods use a search algorithm to evaluate subsets of features, they are more computationally expensive than filter methods. With these techniques, subsets of features are chosen iteratively, and the performance of the IDS is assessed while using the chosen subset. The subset of features with the best performance is chosen by the algorithm.

A hybrid strategy that combines feature selection and model construction is called embedded methods. By choosing features that have the greatest impact on the model's performance during training, these methods incorporate feature selection as a step in the model building process.

It is crucial to carefully choose the best feature selection method for a specific IDS. The choice of feature selection technique can be influenced by elements like the quantity and quality of data available, the network's complexity, and the available computational resources. In general, using multiple feature selection techniques may yield superior results to relying solely on one.

In conclusion, feature selection is a crucial step in creating effective IDSs because it aids in locating the most crucial features that support network attack detection. Feature selection can increase the effectiveness and accuracy of the IDS and increase security against cyber threats by reducing the complexity of the data.

B. Data Preprocessing

The creation of effective Intrusion Detection Systems (IDSs) requires the preprocessing of data as a necessary step. To prepare the raw data for analysis and modeling, it entails cleaning and transformation. The accuracy and effectiveness of the detection process are significantly impacted by the quality of the data used in the IDS.

There are several steps involved in data preprocessing, including:

1) Data cleaning: In this step, errors and inconsistencies in the data, such as missing values, incorrect data types, and outliers, are found and fixed. By lowering the impact of data noise, cleaning the data can help to increase the IDS's accuracy.
2) Data transformation: In this step, the data are transformed into a format that is better suited for analysis and modeling. Scaling, normalization, and feature engineering are some examples of transformation techniques. For instance, feature engineering can assist in developing new features that are more informative for the IDS while feature scaling can assist in modifying the range of values for a feature to improve its comparability with other features.
3) Data reduction: In this step, a subset of the most important features is chosen in order to reduce the size of the data. Features selection and dimensionality reduction are two examples of data reduction techniques. Data reduction aims to decrease the computational expense of the IDS while keeping accuracy.

The specific characteristics of the data and the IDS being developed determine the choice of data preprocessing techniques. To increase the IDS's efficacy and accuracy, it is typically advised to perform data preprocessing before training it.

In conclusion, since it helps to raise the caliber of the data used in the detection process, data preprocessing is an essential step in developing effective IDSs. The IDS can increase its accuracy and efficiency, resulting in improved security against cyber threats. This is done by cleaning, transforming, and reducing the data.

When it comes to data preprocessing in intrusion detection systems, there are additional significant considerations in addition to the three main steps previously mentioned. These consist of:

1) Handling imbalanced data: The data used in IDSs is frequently unbalanced, which means that there are not an equal number of samples in each class. As a result, there

All rights reserved by www.ijsrd.com

may be a bias in favor of the majority class and a poor ability to identify the minority class. Unbalanced data can be addressed using methods like oversampling, under sampling, and SMOTE (Synthetic Minority Over-sampling Technique).

2) Dealing with missing data: In datasets used in IDSs, missing data is a common problem. There are several methods for dealing with missing data, such as imputation, which involves substituting estimated values for missing values, and deletion, which entails eliminating samples or features that contain missing data.
3) Data standardization: Data is transformed to have a mean of zero and a standard deviation of one as part of the standardization process. Certain machine learning algorithms that presume the data is normally distributed may find use for this.
4) Data discretization: Discretization is the process of converting continuously occurring data into discrete data. When working with specific types of data, like network traffic data, which can be analyzed more quickly when grouped into distinct categories, this can be helpful.

In general, data preprocessing is essential to the efficiency of IDSs. The IDS can increase its accuracy and efficiency, resulting in more effective detection and prevention of cyber threats by properly cleaning, transforming, and reducing the data.

IV. EXPERIMENTS

Intrusion Detection Systems (IDSs) have been the subject of numerous tests over the years with the aim of enhancing their precision, effectiveness, and efficiency in identifying and preventing cyber threats. Here are a few instances of IDS-related experiments that have been conducted:

1) Feature selection experiments: Testing various feature selection strategies to see which ones are best at spotting various types of attacks is a common experiment. For instance, researchers may test various subsets of features to determine which ones are most useful for identifying a specific kind of attack.
2) Machine learning algorithm experiments: Comparing the effectiveness of various machine learning algorithms for IDSs is a common experiment. To determine which algorithms are best at identifying various types of attacks, researchers may test decision trees, support vector machines, and neural networks.
3) Hybrid approach experiments: Testing hybrid strategies that combine various techniques to increase the precision and effectiveness of IDSs is a part of some experiments. To develop a more potent IDS, researchers might, for instance, combine feature selection methods with machine learning algorithms.
4) Real-world dataset experiments: To gauge how well IDSs work at identifying actual cyber threats, researchers frequently put them to the test using real-world datasets. Datasets like the KDD Cup 1999 dataset, which is frequently used for assessing IDSs, may be used in these experiments.
5) Evaluation metrics experiments: To gauge how well IDSs work at identifying actual cyber threats, researchers frequently put them to the test using real-world datasets. Datasets like the KDD Cup 1999 dataset, which is frequently used for assessing IDSs, may be used in these experiments.

Overall, experiments are essential to the creation and advancement of IDSs. Researchers can determine the best strategies for detecting and preventing cyber threats by putting various techniques and methods to the test. This will improve security for both individuals and organizations.

There are numerous other significant areas of research into intrusion detection systems (IDSs), in addition to the experiments mentioned above. Here are a few instances:

1) Anomaly detection: A common method for detecting cyber threats is anomaly detection, which involves spotting patterns in network traffic that differ from usual behavior. Numerous methods for anomaly detection have been developed by researchers, including statistical approaches, clustering algorithms, and machine learning strategies.
2) Signature-based detection: The process of signature-based detection entails identifying known attack patterns based on their distinctive signatures. With this method, potential threats are found by comparing network traffic to a database of known signatures. The limitation of signature-based detection, however, is that it can only identify known attacks and is ineffective against fresh or undiscovered threats.
3) Deep learning techniques: In order to increase the precision and effectiveness of IDSs, deep learning techniques like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been used. These methods enable the more accurate detection of sophisticated attacks because they can automatically learn features from unprocessed data.
4) Adversarial attacks: Network traffic is altered during malicious attacks in order to avoid being noticed by IDSs. To test the robustness of IDSs and find gaps in their detection abilities, researchers have developed methods for producing adversarial examples.
5) Online learning: As new data becomes available, machine learning models are updated in real-time for online learning. IDSs, which over time must adapt to new and evolving cyber threats, can benefit from this strategy.

IDSs are the subject of a variety of research projects, all aimed at enhancing their capacity to identify and stop cyber threats. It is essential to create efficient IDSs that can keep up with these evolving threats as cyber-attacks continue to become more sophisticated and common.

The application of ensemble methods is a significant area of research in the context of intrusion detection systems (IDSs). In ensemble methods, various models or algorithms are combined to increase the system's accuracy and robustness. Ensemble methods can be used in the context of IDSs to combine various detection strategies, such as anomaly detection and signature-based detection, to produce a more powerful and complete system.

The application of decision fusion is one typical ensemble method type. To determine whether a network event is an attack or not, decision fusion combines the results of various detection models. Decision fusion can be

All rights reserved by www.ijsrd.com

approached from a variety of angles, such as majority voting, weighted voting, and belief function theory.

The application of diversity techniques is another type of ensemble method. In order to increase the likelihood that at least one of the detection models will be successful at identifying a specific attack, diversity techniques entail developing multiple detection models that are diverse in some way, such as by using various features or algorithms. Researchers may, for instance, develop multiple machine learning models using various feature selection strategies before combining them using decision fusion.

IDS accuracy and robustness have been demonstrated to be improved by ensemble methods. They may also require more data for training and testing and be more complicated and computationally intensive than single-model approaches.

Researchers are looking into the use of other methods, such as explainable AI, transfer learning, and semi-supervised learning, in addition to ensemble methods, to increase the efficiency of IDSs. New and creative strategies to aid organizations in protecting themselves against cyber threats are probably going to be developed as the field of cyber security continues to develop.

The application of ensemble methods is a significant area of research in the context of intrusion detection systems (IDSs). In ensemble methods, various models or algorithms are combined to increase the system's accuracy and robustness. Ensemble methods can be used in the context of IDSs to combine various detection strategies, such as anomaly detection and signature-based detection, to produce a more powerful and complete system.

The application of decision fusion is one typical ensemble method type. To determine whether a network event is an attack or not, decision fusion combines the results of various detection models. Decision fusion can be approached from a variety of angles, such as majority voting, weighted voting, and belief function theory.

The application of diversity techniques is another type of ensemble method. In order to increase the likelihood that at least one of the detection models will be successful at identifying a specific attack, diversity techniques entail developing multiple detection models that are diverse in some way, such as by using various features or algorithms. Researchers may, for instance, develop multiple machine learning models using various feature selection strategies before combining them using decision fusion.

IDS accuracy and robustness have been demonstrated to be improved by ensemble methods. They may also require more data for training and testing and be more complicated and computationally intensive than single-model approaches.

Researchers are looking into the use of other methods, such as explainable AI, transfer learning, and semi-supervised learning, in addition to ensemble methods, to increase the efficiency of IDSs. New and creative strategies to aid organizations in protecting themselves against cyber threats are probably going to be developed as the field of cyber security continues to develop.

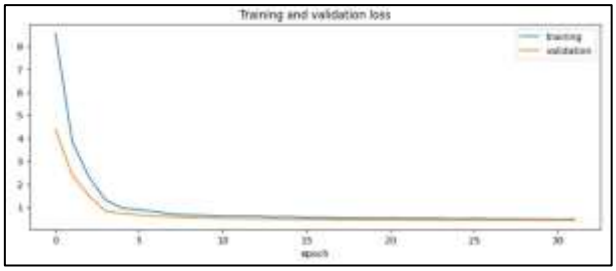

Fig. 1: Training and Validation Loss

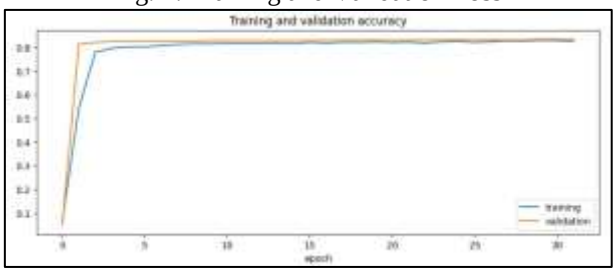

Fig. 2: Training and Validation Accuracy

V. FEATURE TRANSFERABILITY

A key idea in intrusion detection systems (IDSs) is feature transferability, which describes a model's capacity to reliably identify attacks on different datasets. In other words, it refers to how accurately features learned from one dataset can be applied to another dataset.

Because cyberattacks can be very diverse and constantly changing, the issue of feature transferability is crucial for IDSs. As a result, it's crucial to create models that can recognize a variety of attacks, including those that might not be represented in the training data.

It is possible to enhance feature transferability in IDSs in a number of ways. Transfer learning is a popular technique that uses pre-trained models on sizable and varied datasets to extract features suitable for IDSs. For instance, features from network traffic data can be extracted using a pre-trained deep neural network, and then a new classifier can be trained using these features for a particular IDS task.

Increasing the diversity of the training data through the use of data augmentation techniques is another method for enhancing feature transferability. By applying random transformations to the existing data, such as flipping, rotating, or adding noise, data augmentation involves creating new training examples.

The resulting model is likely to be more robust and have better feature transferability if the training data diversity is increased.

Finally, in order to increase feature transferability, researchers are also looking into domain adaptation techniques. By lining up the distributions of the source and target domains, domain adaptation involves customizing the model for the target domain. When the training data and the target data are very different, as when the IDS is installed in a new network or environment, this method may be helpful.

Overall, increasing feature transferability is a significant challenge for IDSs, and researchers are working to address this issue and enhance the accuracy and robustness of these systems by developing new methods.

Other methods have been investigated to enhance feature transferability in IDSs in addition to the methods

All rights reserved by www.ijsrd.com

previously mentioned. Meta-learning, which involves learning how to learn from multiple datasets, is one such method. It has been demonstrated that meta-learning enhances the transferability of features learned from one dataset to another, particularly when there is a dearth of training data.

Utilizing adversarial training, which involves educating the model with both neutral and hostile examples, is an additional strategy. Small perturbations are added to the input data to create adversarial examples, which can lead to the model predicting the wrong things. The model's robustness and feature transferability are improved by training it on adversarial examples.

Finally, to increase feature transferability in IDSs, researchers are also looking into the application of explainable AI techniques. The most crucial features for detecting attacks can be found using explainable AI techniques, which shed light on how the model generates its predictions. The model produced by concentrating on these features is more likely to have improved feature transferability.

Overall, researchers are looking into a variety of techniques to address the issue of improving feature transferability in IDSs. These techniques can help organizations better defend themselves against cyber threats and stay one step ahead of attackers by enhancing the accuracy and robustness of IDSs.

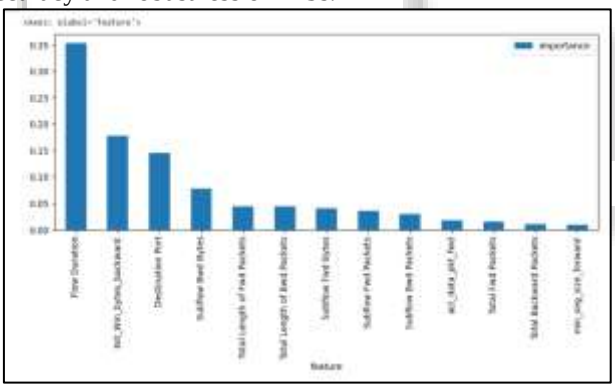

Fig. 3: Feature-Importance Graph

VI. Discussion

A. Deep Neural Network

Due to their capacity to automatically identify complex and non-linear patterns from vast amounts of data, deep neural networks (DNNs) have demonstrated promising results in intrusion detection systems (IDS).

A DNN-based IDS's typical architecture is made up of an input layer, a number of hidden layers, and an output layer. Network traffic data is input to the input layer, where it is processed by the hidden layers to extract features useful for detecting intrusions. The output layer then generates a prediction of whether or not the traffic is malicious.

Convolutional Neural Networks are a common DNN type used in IDS. CNNs excel at tasks involving images, but they can also be used for intrusion detection systems (IDS) by treating network traffic data like a 2D image. A pattern in the traffic data, such as a particular byte sequence or a packet header that indicates an attack, can then be recognized by the CNN. Recurrent Neural Networks (RNN) are another DNN type utilized in IDS. RNNs can be used to model temporal dependencies in network traffic data because they are excellent at sequence-based tasks. An RNN can learn to recognize patterns that are suggestive of an attack by examining the packet sequence, such as a sudden increase in traffic or a sequence of packets with unusual characteristics.

DNNs have overall shown great promise in IDS and are likely to play a bigger role in the fight against online attacks. But developing a successful DNN-based IDS calls for meticulous data prepping, feature engineering, and model selection, as well as ongoing monitoring and tuning to make sure the system stays accurate and current.

B. All features

In order to identify malicious network traffic, intrusion detection systems (IDS) use a variety of features. These characteristics can be broadly divided into three groups:

1) Packet-based features: These features are based on the properties of specific network packets, including their size, protocol, source and destination IP addresses, ports, and payload information.
2) Session-based features: These characteristics, which include the length of the session, the quantity of packets exchanged, the packet rate, and the average packet size, are based on the characteristics of a sequence of packets.
3) Statistical-based features: These characteristics, such as the distribution of packet sizes, the inter-arrival time between packets, and the entropy of the payload, are based on statistical analysis of network traffic.

Some specific features commonly used in IDS include:

1) Protocol anomalies: Detection of protocol violations, such as incorrectly set TCP flags, out-of-order packet sending, or unexpected protocol behaviors.
2) Port scans: detection of attempts at port scanning, which entail connecting to a number of target machines' ports in an effort to find any potential security holes.
3) Denial of Service (DoS) attacks: detection of attacks that aim to overwhelm or crash a system or network, for example, by saturating it with traffic or taking advantage of software flaws.
4) Malware signatures: Detection of known malware signatures in network traffic, such as viruses, worms, and Trojan horses.
5) Anomalous behavior: Detection of unusual network traffic patterns, including large amounts of traffic, high packet rates, and suspicious communication patterns

All rights reserved by www.ijsrd.com

#	Column	Non-Null Count	Dtype
0	Destination Port	19997 non-null	int64
1	Flow Duration	19997 non-null	int64
2	Total Fwd Packets	19997 non-null	int64
3	Total Backward Packets	19997 non-null	int64
4	Total Length of Fwd Packets	19997 non-null	int64
5	Total Length of Bwd Packets	19997 non-null	int64
6	Fwd Packet Length Max	19997 non-null	int64
7	Fwd Packet Length Min	19997 non-null	int64
8	Fwd Packet Length Mean	19997 non-null	float64
9	Fwd Packet Length Std	19997 non-null	float64
10	Bwd Packet Length Max	19997 non-null	int64
11	Bwd Packet Length Min	19997 non-null	int64
12	Bwd Packet Length Mean	19997 non-null	float64
13	Bwd Packet Length Std	19997 non-null	float64
14	Flow Bytes/s	19997 non-null	float64
15	Flow Packets/s	19997 non-null	float64
16	Flow IAT Mean	19997 non-null	float64
17	Flow IAT Std	19997 non-null	float64
18	Flow IAT Max	19997 non-null	float64
19	Flow IAT Min	19997 non-null	float64
20	Fwd IAT Total	19997 non-null	float64
21	Fwd IAT Mean	19997 non-null	float64
22	Fwd IAT Std	19997 non-null	float64
23	Fwd IAT Max	19997 non-null	float64
24	Fwd IAT Min	19997 non-null	float64
25	Bwd IAT Total	19997 non-null	float64
57	Fwd Avg Packets/Bulk	19997 non-null	int64
58	Fwd Avg Bulk Rate	19997 non-null	int64
59	Bwd Avg Bytes/Bulk	19997 non-null	int64
60	Bwd Avg Packets/Bulk	19997 non-null	int64
61	Bwd Avg Bulk Rate	19997 non-null	int64
62	Subflow Fwd Packets	19997 non-null	int64
63	Subflow Fwd Bytes	19997 non-null	int64
64	Subflow Bwd Packets	19997 non-null	int64
65	Subflow Bwd Bytes	19997 non-null	int64
66	Init_Win_bytes_forward	19997 non-null	int64
67	Init_Win_bytes_backward	19997 non-null	int64
68	act_data_pkt_fwd	19997 non-null	int64
69	min_seg_size_forward	19997 non-null	int64
70	Active Mean	19997 non-null	float64
71	Active Std	19997 non-null	float64
72	Active Max	19997 non-null	float64
73	Active Min	19997 non-null	float64
74	Idle Mean	19997 non-null	float64
75	Idle Std	19997 non-null	float64
76	Idle Max	19997 non-null	float64
77	Idle Min	19997 non-null	float64
78	Label	19997 non-null	object
79	Flow ID	19997 non-null	object
80	Source IP	19997 non-null	object
81	Source Port	19997 non-null	float64
82	Destination IP	19997 non-null	object
83	Protocol	19997 non-null	float64
84	Timestamp	19997 non-null	object

dtypes: float64(38), int64(42), object(5)
memory usage: 13.1+ MB

Fig.4. All Features

In general, an IDS's effectiveness depends on its capacity to identify a variety of potential threats while reducing false positives and negatives. Due to the unique network environment and threat landscape, careful feature selection and tuning are necessary.

C. Reduced Set of Features

```
['Destination Port',
 'Flow Duration',
 'Total Fwd Packets',
 'Total Backward Packets',
 'Total Length of Fwd Packets',
 'Total Length of Bwd Packets',
 'Subflow Fwd Packets',
 'Subflow Fwd Bytes',
 'Subflow Bwd Packets',
 'Subflow Bwd Bytes',
 'Init_Win_bytes_backward',
 'act_data_pkt_fwd',
 'min_seg_size_forward']
```

Fig. 5: Selected Features

D. Comparison with Previous Work

The performance of a new intrusion detection system (IDS) is compared to that of existing systems, which may include conventional rule-based systems, statistical systems, or other machine learning-based systems. This is done in order to determine how well the new system performs.

Comparing IDSs should take into account a number of important factors, such as:

1) Detection accuracy: This refers to the IDS's capacity to accurately detect malicious traffic while reducing false positives and negatives.
2) Computational efficiency: This refers to the IDS's processing speed and resource needs, including the amount of time needed to train the model, handle input data, and produce output.
3) Scalability: This is a reference to the IDS's capacity to manage high volumes of traffic and change with the network environment.
4) Generalizability: This speaks to the IDS's capacity to identify fresh threats that weren't seen during training.
5) Explain ability: In order to help users comprehend the underlying causes of an alert, this refers to the IDS's capacity to offer concise and understandable justifications for its choices.

Use the right evaluation metrics and data sets when contrasting a new IDS with earlier work to ensure an objective and insightful comparison. Additionally, it's crucial to take into account the target environment's unique constraints and requirements and to evaluate the system in a real-world setting.

A successful IDS should, in general, be highly accurate and efficient while also being flexible, scalable, and understandable. Researchers can find areas for improvement and advance the intrusion detection field by contrasting a new IDS with earlier work.

E. Takeaways and Future Directions

The following are key conclusions from the current state of intrusion detection systems (IDS):

1) In terms of accuracy and scalability, machine learning-based IDSs, particularly deep learning-based IDSs, are displaying encouraging results.
2) IDSs need to be carefully chosen and tuned in order to be effective. They also need to be continuously monitored and adjusted to changes in the network environment and threat landscape.
3) Users are increasingly interested in learning the underlying causes of alerts and decisions, so the explainability and interpretability of IDSs is becoming more and more crucial.

Future directions for IDS research include:

1) Improved feature engineering and selection: To find and extract useful features from network traffic data, especially for IDSs based on deep learning, more work is required.
2) Better handling of class imbalance: IDSs frequently deal with highly unbalanced data sets, where the majority of traffic is legitimate and the minority is malicious. To create efficient methods for managing class imbalance in IDSs, more study is required.

All rights reserved by www.ijsrd.com

3) Real-time adaptability: Instead of relying on cyclical retraining, IDSs should be able to adapt to changes in the network environment and threat landscape in real-time.
4) Collaborative IDSs: Sharing data and alerts among various systems can increase the effectiveness of IDSs, allowing for quicker reaction times and more precise threat identification.
5) Explainability and Interpretability: In order to foster user confidence and enable efficient alert response, it is crucial for IDSs to provide concise and understandable justifications for their decisions as they become more complex and rely on more advanced machine learning techniques.

IDSs must continue to be developed in order to maintain network security despite the growing dangers posed by cyber-attacks. Researchers can create IDSs that are more effective, efficient, and interpretable and that can adapt to the changing threat landscape by utilizing advancements in machine learning, data analysis, and collaboration.

VII. Conclusion

Through the identification and notification of users of potential threats and attacks, intrusion detection systems (IDS) play a crucial part in network security. Due to the limitations of conventional rule-based IDSs in identifying unknown and complex attacks, machine learning-based IDSs, particularly deep learning-based IDSs, have been developed. These IDSs have shown promising results in terms of accuracy and scalability.

In order for IDSs to be effective, features must be carefully chosen and tuned, monitored continuously, and adjusted to changes in the network environment and threat landscape. Improved feature engineering, better handling of class imbalance, real-time adaptability, collaborative IDSs, and improved explainability and interpretability are some future research directions for IDSs.

In general, maintaining network security in the face of rising threats from cyber-attacks depends on the development of efficient IDSs. Researchers can create IDSs that are more effective, efficient, and interpretable and that can adapt to the changing threat landscape by utilizing advancements in machine learning, data analysis, and collaboration.

References

[1] Kismet. Kismet. Accessed: Apr. 11, 2022. [Online]. Available: https://www.kismetwireless.net/
[2] AirSnort. Airsnort. Accessed: Apr. 11, 2022. [Online].Available: https://ftp.unpad.ac.id/orari/library/library-sw-hw/linux-1/airsnort/AirSnort%20Homepage.htm
[3] Arubaos. Arubaos. Accessed: Apr. 11, 2022. [Online]. Available: https://www.arubanetworks.com/products/network-management-operations/arubaos/
[4] C. Kolias, G. Kambourakis, A. Stavrou, and S. Gritzalis, ‘‘Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset,’’ IEEE Commun. Surveys Tuts, vol. 18, no. 1, pp. 184–208,1st. Quart, 2016.
[5] S. Bhandari, A. K. Kukreja, A. Lazar, A. Sim, and K. Wu, ‘‘Feature selection improves tree-based classification for wireless intrusion detection,’’ in Proc. 3rd Int. Workshop Syst. Netw. Telemetry Anal., Jun. 2020, pp. 19–26.
[6] M. E. Aminanto and K. Kim, ‘‘Improving detection of Wi-Fi impersonation by fully unsupervised deep learning,’’ in Proc. Int. Workshop Inf. Secur.Appl. Cham, Switzerland: Springer, 2017, pp. 212–223.
[7] F. D. Vaca and Q. Niyaz, ‘‘An ensemble learning based Wi-Fi network intrusion detection system (WNIDS),’’ in Proc. IEEE 17th Int. Symp. Netw. Comput. Appl. (NCA), Nov. 2018, pp. 1–5.
[8] A. A. Reyes, F. D. Vaca, G. A. C. Aguayo, Q. Niyaz, and V. Devabhaktuni,‘‘A machine learning based two-stage Wi-Fi network intrusion detection system,’’ Electronics, vol. 9, no. 10, p. 1689, Oct. 2020.
[9] R. Abdulhammed, M. Faezipour, A. Abuzneid, and A. Alessa, ‘‘Effective features selection and machine learning classifiers for improved wireless intrusion detection,’’ in Proc. Int. Symp. Netw., Comput. Commun. (ISNCC), Jun. 2018, pp. 1–6.
[10] J. Ran, Y. Ji, and B. Tang, ‘‘A semi-supervised learning approach to IEEE 802.11 network anomaly detection,’’ in Proc. IEEE 89th Veh. Technol.Conf., Apr. 2019, pp. 1–5.
[11] S. Rezvy, Y. Luo, M. Petridis, A. Lasebae, and T. Zebin, ‘‘An efficient deep learning model for intrusion classification and prediction in 5G and IoT networks,’’ in Proc. 53rd Annu. Conf. Inf. Sci. Syst. (CISS), Mar. 2019,pp. 1–6.
[12] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, ‘‘Building an efficient intrusion detection system based on feature selection and ensemble classifier,’’Comput. Netw., vol. 174, Jun. 2020, Art. no. 107247.
[13] M. E. Aminanto, R. Choi, H. C. Tanuwidjaja, P. D. Yoo, and K. Kim,‘‘Deep abstraction and weighted feature selection for Wi-Fi impersonation detection,’’ IEEE Trans. Inf. Forensics Security, vol. 13, no. 3, pp. 621–636, Mar. 2018.
[14] A. Lazar, A. Sim, and K. Wu, ‘‘GPU-based classification for wireless intrusion detection,’’ in Proc. Syst. Netw. Telemetry Anal., Jun. 2020, pp. 27–31.
[15] A. Agrawal, U. Chatterjee, and R. R. Maiti, ‘‘CheckShake: Passively detecting anomaly in Wi-Fi security handshake using gradient boosting based ensemble learning,’’ Cryptol. ePrint Arch., Tech. Rep. 2021/1702,2021. [Online]. Available: https://ia.cr/2021/1702
[16] S. M. Kasongo and Y. Sun, ‘‘A deep learning method with filter based feature engineering for wireless intrusion detection system,’’ IEEE Access, vol. 7, pp. 38597–38607, 2019.
[17] S. M. Kasongo and Y. Sun, ‘‘A deep learning method with filter based feature engineering for wireless intrusion detection system,’’ IEEE Access,vol. 7, pp. 38597–38607, 2019.
[18] S. M. Kasongo and Y. Sun, ‘‘A deep gated recurrent unit based model for wireless intrusion detection system,’’ ICT Exp., vol. 7, no. 1, pp. 81–87, Mar. 2021.

All rights reserved by www.ijsrd.com

[19] E. Chatzoglou, G. Kambourakis, and C. Kolias, ‘‘how is your Wi-Fi connection today? DoS attacks on WPA3-SAE,’’ J. Inf. Secur. Appl., vol. 64, Feb. 2022, Art. no. 103058.
[20] O. Kanhere and T. S. Rappaport, ‘‘Position locationing for millimeter wave systems,’’ in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2018, pp. 206–212.

All rights reserved by *www.ijsrd.com*