Exploring Modern Intrusion Detection Systems based on Machine Learning and their implementation on SDN and IoT Networks: A Literature Survey

Ayushi Amin

***Abstract*—In today's world, the growing popularity of the Internet and increasing connectivity of end devices, has resulted in a sudden rise in the number of cyber threats and attacks. As the popularity increases, the complexity of attacks rises. Intrusion Detection System (IDS) is a security-related method that helps protect systems or networks against such attacks by regularly monitoring the network for suspicious traffic and raise alerts when needed. However, traditional IDS face issues to identify new attacks, which can go undetected.**

Hence, Machine Learning (ML) algorithms have been used as a solution to the above-mentioned issue. This survey aims to discuss how ML has improved conventional IDS and assess the effectiveness and functionalities of some of the ML-IDS models built for SDN and IoT networks.

The structure of the paper is as follows: Section I gives a brief overview on IDS, ML, IoT, and SDN. Section II investigates some of the existing ML-IDS models designed for IoT and SDN architectures. Section III discusses the benefits, disadvantages, and some suggestions for system improvement. Lastly, section IV presents the conclusion of this survey paper.

***Index Terms*—Intrusion Detection Systems, Machine Learning, Deep Learning, Internet of Things, Software Defined Networking.**

I. Introduction

When a system is subjected to intentional unauthorized access or malicious activities which compromise the confidentiality, integrity, or availability, the system is said to be under attack. Such activities that threaten and harm the security and functioning of the system can take place in various forms such as Denial of Service (DoS) attacks, installation of malware, social engineering, phishing, etc.

Malicious attacks can have harmful consequences for organizations, people, and nations. The aftermath depends on the nature of the attack or target. Leak of private information and interruptions in operations are examples of problems associated with cyberattacks. A popular security-related method that can tackle these are Intrusion Detection Systems (IDS). IDS can quickly detect suspicious traffic and mitigate them.

A. Intrusion Detection System

Similar to a Burglar Alarm [1], an Intrusion Detection System (IDS) is built to detect harmful traffic over a system. An IDS periodically monitors the network or system to identify and respond to potential intrusions resulting from internal or external attacks.

The below list displays the classification of Intrusions. They are primarily classified into two major types:

Internal Attacks: These kinds of attacks typically originate from within the organization. Such kinds of attacks are further classified into sub-types [2]: The Masquerader, The Legitimate User, and The Clandestine User.

External Attacks: These types of attacks typically originate from outside the organization. Generally, external attacks target to compromise the system or network. Some examples are Denial-of-Service (DoS), Packet Spoofing, or Malware.

As both types of attacks can cause potentially harm the three properties of security, it is essential for an IDS to recognize and respond to various types of intrusions discussed above. Ideally, an IDS should quickly identify anomalies, assess them, and take appropriate measures to mitigate potential threats.

To secure a network architecture against unseen attacks, an IDS should adapt to new attacks by using ML and Deep Learning (DL) algorithms. Hence, the design and implementation of an IDS plays a vital role in protecting a system against possible security risks.

1. Components of IDS: The design of a typical IDS involves a combination of specific rules, software and hardware components that work together to identify, analyze, and respond to potential security threats. Figure 1 describes the three core components of an IDS.

Data Collection

The first component of an IDS typically involves gathering information from the source target. The source can be the network or system. The data collected includes network traffic, system logs, port scanners, application logs and other relevant data. The event generator placed in the data collection component collects and analyzes the logged information continuously. If the IDS is network-based, the network packets are monitored. However, for host-based IDS, the focus is on the system, which include system logs, port scanners, and system security logs. At times, the data collected is usually preprocessed to a specific format before being sent to the detection component. However, even raw information (without preprocessing) can also be forwarded to the next component [3].

Detection

The detection component is responsible for examining the captured information to identify patterns or anomalies that indicate suspicious behavior or potential breaches. This component has

two distinct methods for detection: Signature-based or Anomaly-based. Signature-based detection relies on a database containing known patterns of malicious activities, while anomaly-based detection identifies abnormal behavior that vary from normal ones. From figure 1, we can see that the detection component contains an analysis engine. Often, the engine utilizes multiple detection algorithms to further enhance the process by accurately detecting intrusions [3].

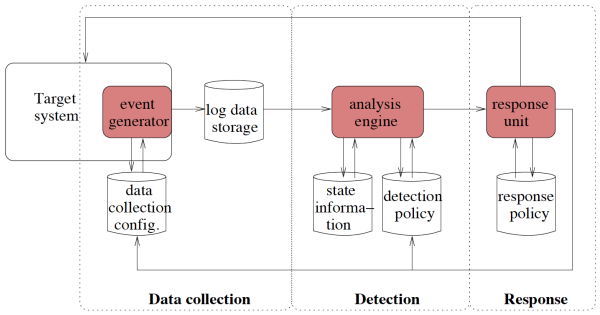

Figure 1: The components of Intrusion Detection Systems [3]

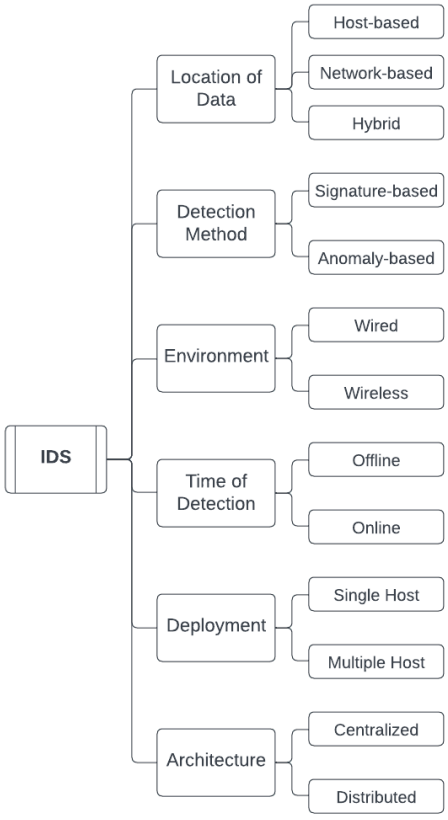

Figure 2: Classification of Intrusion Detection Systems (based on several factors)

Response

Upon detecting intrusions or anomalous behavior, the response component is activated. It involves taking action to mitigate the impact of the breach. Responses can be classified into two types: Active and Passive. Active responses include immediate predefined countermeasures to neutralize the attack such as system lockdown, isolation of compromised systems, or shutting down of the process. Conversely, passive responses involve alerting or notifying the security administrator. Modern intrusion detection systems utilize a combination of both response strategies, ensuring a more comprehensive and adaptive approach in mitigating potential threats. The response process consists of six phases: Preparation, Identification, Containment, Eradication, Recovery, and Follow up [4].

2. Taxonomy of Intrusion Detection System: Figure 2 illustrates a comprehensive taxonomy of IDS, as described in [5] and [6]. This classification is based on specific factors such as location of source data, method of detection, environment, time of detection, method of deployment, and type of architecture.

B. Machine Learning

Machine Learning (ML) refers to the approach of utilizing specific algorithms to train models in such a way that they can generate accurate results based on what they learned during model training [7].

There exist four types of ML algorithms: Supervised Learning, Unsupervised Learning, Reinforcement Learning, and Semi-Supervised Learning. As most ML-IDS architectures are built using supervised or unsupervised learning algorithms, this survey paper will be focusing on only these two algorithms. In supervised learning, the ML models are trained on labeled data so that they can understand and learn the relationships between features and their respective labels. This can help them categorize or predict the label of unseen data. Unsupervised learning involves training the model on unlabeled data. This is done so that the model can study and identify hidden patterns and relationships within the data.

ML has become very popular in cybersecurity, due to its ability to detect anomalies within a network or architecture. The models do so by learning patterns of both regular and suspicious traffic. ML can adapt, learn to detect unseen threats using their prior knowledge, and classify normal and abnormal traffic. As attacks continue to evolve, the integration of ML in IDS is essential in securing the system from unknown attacks.

C. Internet of Things

The Internet is referred to a global network which facilitates the connection of millions of devices worldwide, enabling exchange of data and communication. Due to its ability to provide instant access to information, online services, and real-time communication, the Internet has become an essential component of our daily lives. The idea of the Internet of Things (IoT) has emerged as a result of ongoing advances in technology, further expanding the capabilities of the Internet.

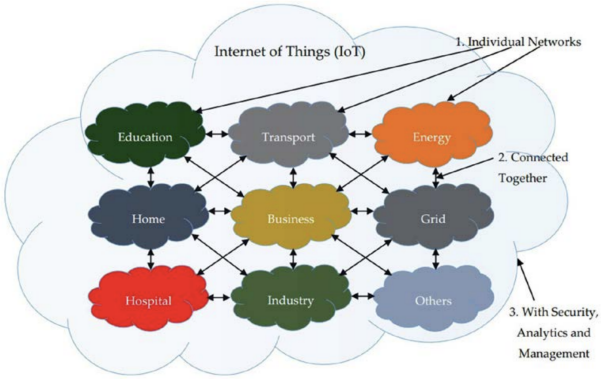

Figure 3: The Internet of Things (IoT) [8]

The Internet of Things (IoT) is described as a network of interconnected smart devices, objects, and systems embedded with sensors, software, and processing capabilities, enabling them to connect, communicate, and exchange data through the Internet. This network comprises of a diverse range of devices, including household appliances, wearable devices, industrial systems, and smart city infrastructure, all of which can transmit data via the Internet [9].

The fundamental objective of the Internet of Things is to establish a seamlessly interconnected network, facilitating devices to communicate effortlessly with each other, regardless of time or location, through a network or service. This connected network of devices holds the promise of advancing real-time monitoring, process control, tracing, and automation, thereby improving overall efficiency [9]. Additionally, it opens avenues for innovative opportunities across various industries. The potential applications of IoT are expansive, spanning across diverse industries, including healthcare, transportation, agriculture, and beyond.

D. Software Defined Networking

Traditional networking architectures utilize hardware-centric approaches, where "dedicated" network devices, such as routers and switches, are responsible for both packet forwarding and control functions. In these architectures, the control plane, which is responsible for determining how packets will be forwarded/routed, and the data plane, which forwards the packets based on the routing table constructed in the control plane, are integrated within the network devices. This integration can lead to limitations in terms of flexibility, scalability, and adaptability to changing network requirements as in such devices most of the functionality is implemented. Scaling or reconfiguring such network architectures often involves manual configuration and can be time-consuming [10].

Software Defined Networking (SDN) architectures introduce a new approach to traditional networking architectures by separating the control plane from the data plane. In SDN, the control plane is centralized in a software controller typically known as SDN Controller. The SDN architecture consists of three layers: the Application Layer, the Control Layer, and the Infrastructure Layer.

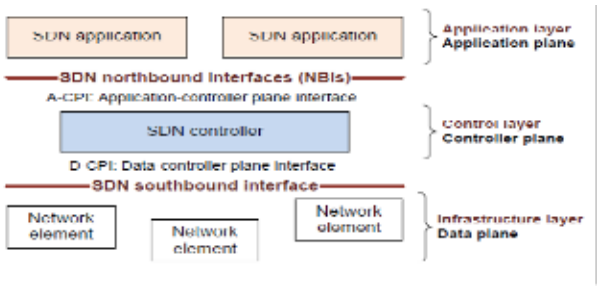

Figure 4: Software Defined Networking Architecture [10]

The Application Layer consists of the network applications that interact with the SDN controller via the Northbound APIs (eg: REST APIs). The Control Layer hosts the SDN Controller that is responsible for making decisions about network traffic forwarding as per the controller policies. Lastly, the Infrastructure Layer comprises of physical and network devices (eg: switches, routers, etc.) in the data plane. This layer deals with functions like forwarding of packets and operates based on the decisions made by the SDN controller located in the previous layer. The infrastructure layer interacts with the controller using the Southbound APIs (eg: OpenFlow Protocol) [10].

II. Machine Learning-based IDS

Machine Learning-based Intrusion Detection Systems (ML-IDS) utilize ML algorithms such as BiLSTM, SVM, KNN, etc. to examine and study network traffic patterns to detect discrepancies that indicate potential security threats. By training on a dataset, ML-IDS can autonomously learn the normal and malicious behavior that can help it identify suspicious activities. This approach of utilizing ML techniques can help traditional IDS to adapt to evolving and dynamic cyber threats. The system's ability to constantly learn and improve over-time makes it suitable for detecting both known and unknown attacks, thus boosting the overall security of the network.

A. ML-IDS based on IoT

With the growing number of attacks on IoT networks, traditional IDS encounter various challenges and are less suitable to detect these attacks, due to specific characteristics associated with IoT networks. Example: The devices on IoT systems often utilize restricted computational resources, hence, traditional IDS may overload the network devices, leading to increased consumption of energy. Other features that make traditional IDS less

sufficient for the network include, network diversity, dynamic topology, restricted capacity of bandwidth, and global span of network [11]. Researchers are now looking into leveraging ML algorithms to solve the above issues faced by traditional IDS when deployed on IoT environments. In this paper, three such ML-IDS systems built for IoT networks are discussed.

1) CNN-BiLSTM-attention and Knowledge Graph based IDS: Researchers at Wuhan University proposed an ML-based IDS that utilized Knowledge Graph and Deep Learning (DL) techniques to monitor the network traffic features of IoT Systems [12]. The proposed model focuses on capturing semantic relationships between malicious requests and the key features of standard and suspicious activities, specifically Denial of Service (DoS), U2R, Probe, and R2L kind of attacks.

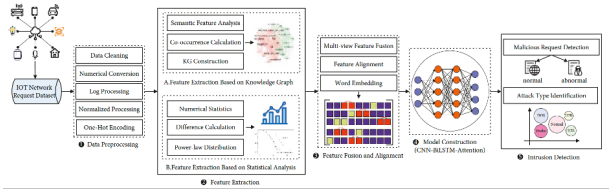

Figure 5: The architecture of CNN-BiLSTM-attention and Knowledge Graph based IDS [12]

Figure 5 displays the overall architecture of the proposed IDS. The proposed system consists of five main phases:

- **Data Pre-Processing:** The process starts off by handling empty and redundant entries. As an input pre-requisite to DL models, categorical attributes were converted into values that can be interpreted by the model. Furthermore, dimensionality reduction was implemented on specific attributes to ensure they don't affect model learning. It is essential to know that the NSL-KDD dataset is an imbalanced dataset, which can cause the DL model to favor the majority class. As a result, the authors normalized the dataset to have values in the range [0, 1]. The final step was one-hot encoding, where the labels were transformed into unique binary representations [12].
- **Feature Extraction:** This phase is essential for capturing meaningful information from the dataset. This proposed IDS employs two methods. First, a knowledge graph-based method that uses cooccurrence calculation and semantic feature analysis to construct a knowledge graph by extracting important feature pairs associated with suspicious and regular requests. The resulting set of pairs can facilitate the examination of semantic relations between features that correspond to different attack types, which can help the DL model in classification related tasks. Second, as individual features alone also can help in the enhancement of the IDS, a statistical analysis-based method was used to extract the individual key features. The median, mode, and average of each of the features (train set) were calculated, where the average was selected as the threshold. Additionally, the paper also calculates the sum of average differences across the features in network requests. A sorting algorithm was applied on the intrinsic features, time-based features, content features, and host-based features to determine the top two values of each feature type. Through this process, the authors were able to form eight key attributes to help augment the functioning of the IDS. The final step in this phase was to use the power-law distribution analysis [12] to understand the dataset's statistical characteristics. This analysis revealed a long-tail distribution that is prevalent in certain attacks.
- **Feature Fusion and Alignment:** For this phase, the paper proposes a Multiview Feature Fusion and Feature Alignment method. In this step, the feature pairs extracted through the knowledge graph generated in the previous step are transformed into sequences and aligned. Furthermore, redundant data is also eliminated (if present). Second, the knowledge graph feature pairs were fused with the features extracted through statistical analysis. Weights are then assigned to the features of the fusion based on their importance in the fusion process [12]. These serve as the initial weights for the DL model.
- **Model Construction:** This phase introduces the attention-based CNN-BiLSTM model proposed for effective intrusion detection. This model consists of six important layers [12]:

  1. Word Embedding Layer: This layer uses the Word2Vec model to transform the important network traffic features into embeddings, so that it can be read by the DL model.
  2. Convolutional Layer: This layer utilizes a convolutional kernel to obtain essential attributes of the traffic, such as those that highly indicate a specific attack.
  3. Pooling Layer: This layer uses the Max Pool function to perform dimensionality reduction while selecting the fundamental attribute(s) of both classes.
  4. BiLSTM Layer: A BiLSTM neuron comprises essential components such as, Input gate, Output gate, Forget gate, and a Memory Unit. As this paper is utilizing a BiLSTM, the output vectors from the previous model will be transformed in both the forward and backward directions, helping the model in remembering contextual information and long-distance dependencies of the traffic.
  5. Attention Layer: This layer assigns attention to features extracted in the Feature Extraction phase by adding certain attention weights. This helps the model identify the important features that can detect which packets are malicious or normal.
  6. Fully Connected Layer: This layer serves as the main classifier for the IDS, associating the features learned to "normal" or "attack" labels using the softmax function.

As the model was trained using NSL-KDD, the same dataset was used for IDS evaluation. Python's TensorFlow and Keras ML libraries assisted in the experimental setup [12]. Model evaluation found that the CNN-BiLSTM-attention and Knowledge Graph based IDS achieved higher scores than the state-of-the-art IDS. The F1-Score and accuracy values were reported to be 90.71% and 90.01%, respectively. Moreover, DoS and Probe attacks had a higher recognition rate compared to U2R and R2L.

2) Man in the Middle Attack Detection IDS: The authors of [13] developed an ML-based IDS that utilized the Adaboost, Naïve Bayes, Decision Trees, and SVM algorithm to classify network requests as malicious or normal. The approach utilized a combination of hardware and software components such as Node MCU ESP8266, Kali Linux, DHT11 Sensor, and the ThinkSpeak cloud to simulate the attacks on a replica of the IoT environment. The methodology mainly focuses on detecting Man-in-the-Middle Attacks.

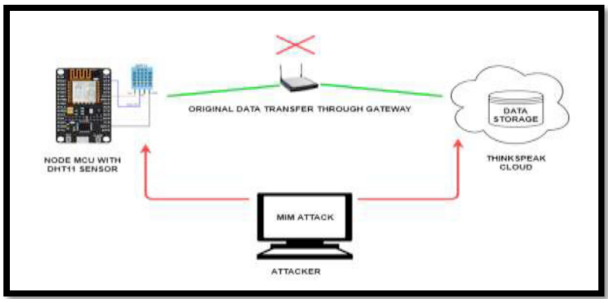

Figure 6: The architecture of Man in the Middle Attack Detection IDS [13]

Below discussed are the various phases involved in the building of the proposed IDS system [13]:

- **Construction of test bed for IoT Simulation:**
  1. IoT Platform Configuration: Setup the connection of DHT11 to the Node MCU Client.
  2. ThinkSpeak Platform Configuration: For this experiment, the ThinkSpeak platform creates a channel, which is further divided into three separate fields: due point, temperature, and humidity, to store the received sensor data. API key(s) are used for performing the read/write functions into the channels and channel id(s) are used for identification of channels.
  3. IoT Platform Coding: In this step, the client's required libraries were loaded onto the Arduino IDE. Additionally, API keys (used for read/write) are configured on the client's end to match ThinkSpeak's respective channel.
- **Adversarial System Setup:**
  1. The adversary analyzes the packets sent from the client to the server, which contain data generated by the sensor. Using WireShark, the adversary determines the active network protocols in use and identifies the IP addresses of the client, server, and themselves [13].
  2. In the second step, the "Unified Sniffing" technique [13] is used to capture packets being sent from the client to the server. Ettercap was then utilized to implement ARP Poisoning. Additionally, packets of remote connections were also sniffed and captured.
  3. Third, the Burp Suite on the attacker' system is configured to serve as a proxy listener. All network requests made on port 8080 are intercepted by Burp Suite. The attacker system's web browser is set up as a proxy server, facilitating the transmission of packets sent from Node MCU to ThinkSpeak through burp suite [13]. The Burp Suite modifies the information contained in these packets. Subsequently, the altered data is transmitted to the server, while spoofing as originating from the client.
- **Server Data Collection:** In this experiment, the ThinkSpeak server collects the data received from the client. Packets that are untouched are associated with "Normal" traffic and the ones that are modified come under the "Attack" class.
- **Machine Learning models design:** For this experiment, the authors have utilized four machine learning algorithms to perform binary classification: malicious request (will be classified to "Attack" class) and unmalicious request (will be classified to "Normal" class). The algorithms used were: Naïve Bayes, SVM (Support Vector Machine), Adaboost, and Decision Trees [13].

At the end of the experiment, model evaluation was performed by computing typical classifier metrics such as, precision, recall, F1-Score, accuracy, rate of error, false alarm rate, specificity, and detection rate. The results showed that almost all the ML algorithms achieved accuracy scores of 99%, with 1 or 2 false positives.

3) MLP-SAE based IDS: The paper [14] proposes two ML-IDS models: semi-distributed and distributed, suitable for IoT network devices with limited computation resources. To introduce distributed parallel processing into the IDS, the AWID dataset was partitioned into three parts, and three classifiers were parallelly trained. To choose these classifiers, five ML algorithms (InfoGain, J48, One Rule, SVM, and CFS) were trained on each of the three subsets of the dataset. The one with the best results for each partition was chosen. The MLP classifier was used for classifying impersonation attacks in this experiment.

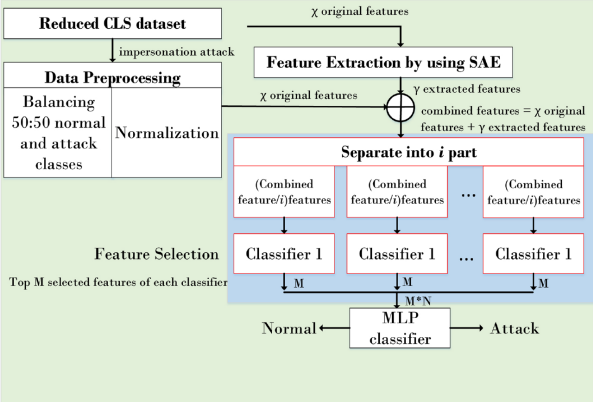

Figure 7: The architecture of the proposed Semi-Distributed IDS [14]

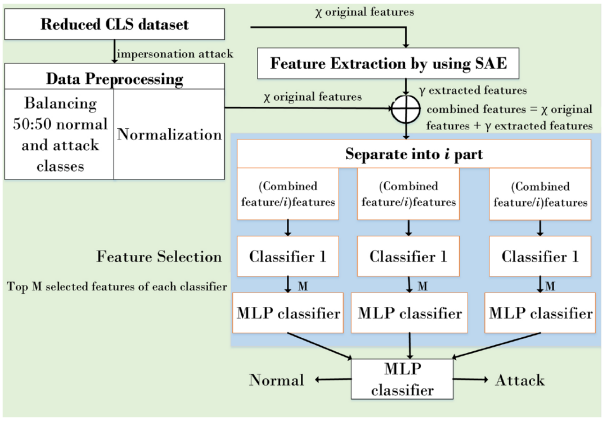

Figure 8: The architecture of the proposed Distributed IDS [14]

To construct the MLP-SAE based IDS, there were four processes involved:

- **Data Pre-Processing:** As this paper focused on only impersonation attacks, the number of instances that the experiment had initially was 97,000+, consisting of values for both normal requests and impersonation attacks. As a requirement for DL models, all dataset values were converted into numerical values. Additionally, for uniformity, the values were normalized between 0 to 1 [14] . Upon completing the pre-processing of the data, the dataset was split into train and test sets.

- **Feature Extraction:** In this phase, a Stacked Auto Encoder (SAE) was used for performing feature extraction. In SAE, multiple AEs are stacked upon each other. Each layer of the SAE is an AE, which consists of both encoder and decoder. The output of one layer (AE) becomes the input for the successive layer. For this experiment, the choice of SAE model chosen was the 154:100:50 model. There are chances of the DL model overfitting when the new extracted features are given as input to the model in addition to the training data. To ensure this doesn't happen, sparsity regularization was used along with Kulback-Leiber (KL) divergence [14] .

- **Feature Selection:** For this experiment, two major methods of feature selection were utilized: wrapper and filter-based methods. The SVM and J48 algorithms were chosen for the wrapper-based methods, whereas the filter-based methods chosen were InfoGain, CFS, OneR [14] .

- **Classification:** The authors of this paper employed an Artificial Neural Network (ANN) to classify normal traffic or impersonation attacks on an IoT Wi-Fi network. A Multilayer Perceptron (MLP) was chosen, that uses weights to help classify WiFi traffic as "Normal Attack" or "Impersonation Attack". ANNs have typically have three layers, i.e., Input, Hidden, and Output [14] . The features from the previous step is processed by these three layers. A bias function is also introduced into the MLP to identify for any potential signs of impersonation attacks on the WiFi traffic.

Similar to the above experiments, the precision, F1-Score, accuracy, false alarm rate, and detection rate metric were computed. Results indicated that for the first partition, InfoGain and OneR achieved an accuracy of 99.8%. SVM attained the highest accuracy (97.14%) for the second partition, while in the case of third partition, CFS was selected (accuracy = 96.45%). The above-mentioned metrics were also calculated for the two proposed methods. The distributed IDS attained a detection accuracy of 97.8% (CPU Time = 73.52 s), while the semi-distributed IDS had a CPU Time of 186.26 s and detection rate of 99.97%.

B. ML-IDS based on SDN

Traditional IDS implemented on SDN environments might not be able to completely secure the network as traditional systems are primarily implemented using signature-based methods [10]. Signature-based techniques involve detecting attacks based on a predefined database of signatures. Besides known attacks, SDN architectures are also vulnerable to dynamic novel attacks, which makes traditional IDS ineffective for SDN environments. Moreover, handling a huge signature-based database can increase the load [15]. Anomaly-based methods, specifically ML and DL techniques are now leveraged as a solution to the issues faced by traditional signature-based IDS. This survey discusses three ML-based IDS models executed to secure SDN architectures.

1) DDoS Detection Bloom Filter-ML based IDS: The authors of the paper [16] designed an IDS to detect Distributed Denial-of-Service (DDoS) attacks on SDN Networks. The proposed approach utilizes a Bloom Filter and three ML algorithms to detect and classify DDoS attacks. The Bloom Filter stores a set of known malicious IP addresses, while the ML algorithms trains over certain parameters to identify unknown attacks. The Kali Linux simulator, Wireshark, and Tshark were used in this experiment. Figure 9 illustrates the different layers of the SDN and their respective roles for the successive implementation of the IDS.

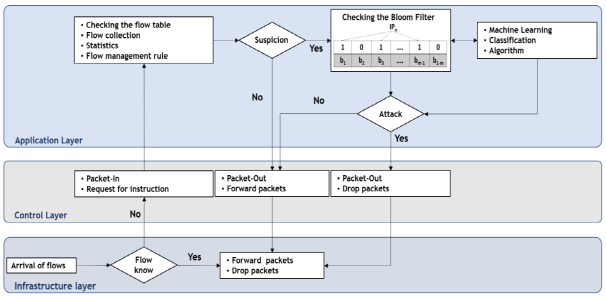

Figure 9: The DDoS Detection Bloom Filter-ML based IDS [16]

The implementation of the IDS is discussed below:

- **Infrastructure Layer (Data Plan):** Initially, when new packets reach the switch present in this layer, it examines to see if it recognizes the packet. If yes, it processes it according to the flow table rules . Else, it sends the packet to the controller.

- **Control Layer (Control Plan):** If the controller recognizes the incoming packet, it creates a new entry for the packet in the flow table. This new entry is then sent down to the switch, who processes the packet based on the entry value created by the controller. If the controller doesn't recognize the packet, through the OpenFlow protocol, the controller sends the packet header to the ADIS module located in the layer above [16].

- **Application Layer (Application Plan):** Within this layer, the SDN network traffic flow is captured and relevant features such as the source IP address, destination IP address, packet count, and so on are studied. The source IP address of the packet is inspected to see if it is present in the Bloom Filter. If no, ADIS inspects the packet further by considering an additional criteria, "The rate of packets forwarded per second from the source to the switch" [16]. The module utilizes the LDA, SVM, and KNN to determine whether the source is normal or potentially malicious.

In the last phase of this experiment, R-Studio was used to plot the findings [16]. For each of the three ML algorithms, the collected data packets were classified into three distinct classes (Normal, Malicious, and Suspicious) for evaluation. In comparison to the other methods, the SVM algorithm obtained the highest results. The error rate of the proposed IDS was reported to be 1.29%.

2) ANIDS-ML based IDS: A convolutional Neural Network (CNN) was designed to categorize the network traffic as normal or abnormal. The model was trained on certain features of the KDD 99 dataset, hence focused on identifying Denial of Service (DoS), U2R, Probe, and R2L attacks. Furthermore, this paper introduces a unique "Sensing" module to detect suspicious traffic (DoS, R2L, U2R, and Probe type of traffic), which is unfamiliar to the IDS [17].

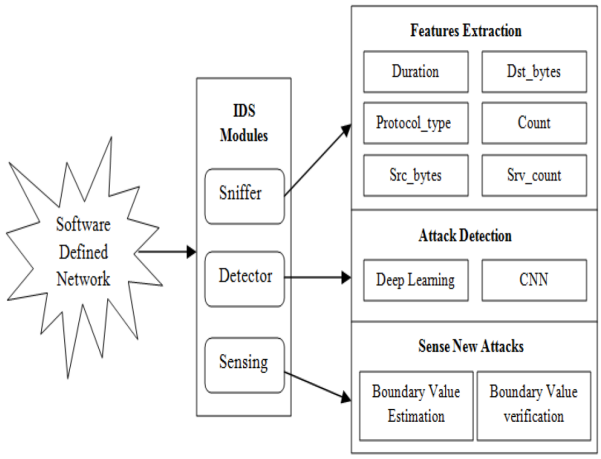

Figure 10: The proposed ANIDS model [17]

As seen from Figure 9, there are three main modules in the ANIDS-ML architecture, which are elaborated as follows:

- **Sniffer:** This module analyzes and sniffs out data packets from the SDN traffic to capture six specific features: Duration, srv_count, protocol_type, count, src_bytes, and dst_bytes [17]. These extracted attributes will be fed into the DL model to determine if the network traffic was normal or malicious.

- **Detector:** In the detector module of the ANIDS architecture, the Convolutional Neural Network (CNN) DL model was chosen as the choice of classifier. The input to the model will be the extracted attributes, which is derived from the previous module. To classify, the CNN was trained on the duration, srv_count, protocol_type, count, src_bytes, and dst_bytes attributes of the KDD 99 dataset [17]. The model generates results classifying network traffic as either malicious or normal based on its training .

- **Sensing:** The sensing component of the proposed architecture was created to categorize novel or unfamiliar threats. It utilizes the "Boundary Value Testing" hypothesis [17] which can identify known and unknown attacks on the SDN environment. For detection, the minimum and maximum features of each DoS, R2L, Probe, U2R, and Normal were extracted from the chosen dataset. When the IDS comes across a new packet, the features extracted are compared against the selected boundary values. If they are greater, the packet's extracted features are forwarded to the CNN to determine and categorize new threats.

Experimental evaluation was not carried out on the above discussed IDS as there is still scope for improvement. The authors target to optimize the algorithm and train the DL model to identify other attacks besides the above-mentioned attacks [17].

3) HFS-LGBM IDS: This paper [18] proposes an HFS-LGBM IDS for the SDN network that utilizes the concept of Hybrid Feature Selection (HFS) to generate an optimal subset of feature

and perform dimensionality reduction. Furthermore, the IDS architecture implements the LightGBM ML algorithm for model training. Like other implementations discussed above, the HFS-LGBM IDS uses the popular NSL-KDD dataset for training and evaluation.

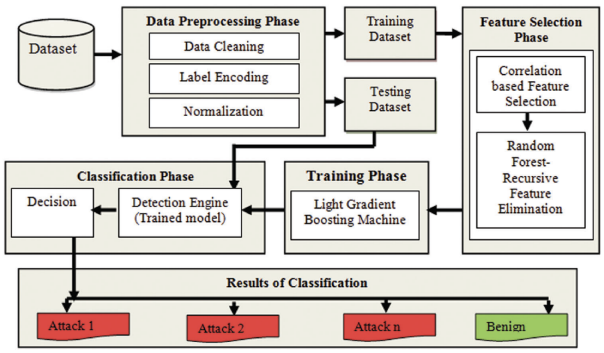

Figure 11: The architecture of HFS-LGBM IDS [18]

Similar to other architecture discussed above, the proposed IDS has three primary phases:

- **Data Preprocessing:** In the initial step of data pre- processing, empty entries were removed from the dataset. As ML models are unable to process categorical values, ordinal and one-hot encoding methods were used to transform these values into numerical representations. As the NSL-KDD dataset is imbalanced, the Min-Max normalization techniques was used to convert values within the range [0, 1] [18] .
- **Feature Selection Phase (HFS):** This feature selection step consisted of two stages:
  1. Correlation Based Feature Selector (CFS): This step is used to create the initial draft of the feature subset which won't contain any noisy or irrelevant features. The experiment chooses features that have some connection or one that can help in the prediction of a class. Those features which were excessively linked with other features are not considered. To assess the relationship between a feature and extrinsic variables, the Pearson's correlation coefficient was computed using the feature values, labels, and their averages. Lastly, the aggregation correlation coefficient was also computed to examine how relevant is the created feature group [18] .
  2. Random Forest Recursive Feature Elimination: The RF-RFE greedily finds the most optimal subset from the feature set generated in the previous step. First, random forest models are iteratively generated. At the same time, the most or least important feature is deleted in each iteration. This procedure is repeated with the remaining attributes of the feature set, until the set is empty. Upon completion, feature ranking is performed by arranging the deleted features based on their elimination order [18] . In the final step, an algorithm is used to get the top n features to create the most optimal feature subset.
- **Training Phase:** The authors employed the LightGBM ML algorithm to train the optimal feature subset generated in the previous phase. LightGBM parallelly learns and uses histogram-related methods to study the given dataset [18] .
- **Classification Phase:** In the final phase, the trained model uses the Voting Scheme and the average probability method to perform "Benign" and various types of other attacks [18] present in the given set.

In the evaluation phase, the Mininet Emulator was used for testing [18]. The standard ML model evaluation metrics: Accuracy, Precision, Recall, and F1-Score, were utilized to evaluate the IDS. Results indicated that the HFS-LGBM IDS architecture was able to outperform benchmark models, obtaining an accuracy of 98.72% and F1-Score of 98.23%.

III. Discussion

In this section, we present the advantages, disadvantages, and possible future suggestions of the above six discussed ML-IDS.

CNN-BiLSTM-attention and Knowledge Graph based IDS

- **Advantages:** The proposed DL model introduces an attention layer, which can help the model focus on important parts of the input sequence during the classification process. The use of a knowledge graph helps capture meaningful connections between malicious requests and important attributes of different attack types. The fusion of features extracted by knowledge graphs and statistical analysis allows the DL model to simultaneously consider both individual features and relationships between features, potentially enhancing capacity of understanding complex patterns and thus improving its learning.
- **Disadvantages:** The model is not generalized to the diversity of real-time network traffic as it heavily relies on the NSL-KDD dataset. Its performance might be influenced by the quality of the dataset, which can limit its performance in generalizing and detecting real-time attacks. Additionally, this approach can be computationally intensive, having a requirement of computational resources.
- **Future Suggestions:** To improve the effectiveness of the model in detecting real-time attacks, this approach can be tested on a simulation of IoT networks before testing on an actual network. Incremental learning can be implemented to improve the adaptability of the model to evolving dynamic attacks.

Man in the Middle Attack Detection IDS

- **Advantages:** The construction of a test bed for the simulation of an IoT environment, allows for the replication of real-time scenarios, providing a suitable environment for test the IDS. Furthermore, the inclusion of an adver-

sarial setup in the simulated IoT environment, assists in replicating realistic attacks typically taken place in IoT environments. By simulating attacks such as ARP poisoning, packet data modification, etc. the IDS can be tested against various attacks.

- **Disadvantages:** This approach only tests the proposed IDS on an IoT environment using the DHT11 sensor, which is connected to the Node MCU client. However, an IoT is a network of inter-connected devices. The proposed IDS might not scale well when used in a network with a number of devices as it is tested with only one sensor device.
- **Future Suggestions:** The proposed method can be extended to detect other types of attacks such as Denial-of-Service (DoS) attacks, etc. and other adversarial intrusion techniques can also be tested for the comprehensive evaluation of the IDS model. The method can also be modified to use another set of classifiers. DL models like BiLSTM and GRU can help uncover hidden patterns that contribute to the detection of malicious packets.

MLP-SAE based IDS

- **Advantages:** The paper presents two distinct approaches: Semi-distributed IDS and Distributed IDS. This allows flexibility in the implementation of IDS based on available computation resources. The dataset is also partitioned into three sets to introduce distributed parallel processing. This can be beneficial for IoT networks where data maybe distributed across multiple devices of the network.
- **Disadvantages:** The proposed approach only uses 97,000 instances of the dataset. When building an ML-based IDS for detection of impersonation attacks, it is ideal if the model is trained on a vast amount of data so that it can study the data well to understand a variety of patterns and thus improve its ability to generalize well to unknown data using the knowledge gained.
- **Future Suggestions:** The model can possibly be modified to incorporate real-time processing capabilities to enable timely attack detection. It can also incorporate the idea of federated learning to ensure secure data is kept private while being integrated on a real-time IoT network.

DDoS Detection Bloom Filter-ML based IDS

- **Advantages:** The authors of this paper have used a Bloom Filter to store a set of malicious IP addresses. The use of this structure provides a scalable and memory-efficient method of quickly identifying known attackers. Moreover, in this method, the ADIS module is implemented in the application layer and only packets that require further analysis are forwarded to the controller. This reduces the unnecessary load on the control layer and improving network efficiency.
- **Disadvantages:** The paper doesn't mention about the handling of encrypted payload. In the modern era, packets transmitted from the source to the destination are typically encrypted. Unless there is no use of keys or decryption algorithms implemented, it will be hard for the encrypted packet to be analyzed. Additionally, there are chances of false alarms taking place in anomaly detection.
- **Future Suggestions:** To improve the drawbacks discussed above, methods to handle encrypted traffic can be examined, tested, and possibly integrated into the approach. ML algorithms like Isolation Forests, etc. can be used to reduce the false positive rate.

ANIDS-ML based IDS

- **Advantages:** The proposed system introduces a sensing component that utilizes the boundary concept to detect unknown threats. It provides a technique to adapt to dynamic threats by retraining the CNN when the extracted features exceed predefined boundaries. The use of a sniffer to capture and study real-time SDN traffic allows for more accurate attack detection, enhancing the system's effectiveness when used in real-time.
- **Disadvantages:** The effectiveness of the boundary method depends on how well it can adapt to various evolving attacks. If the proposed system is not regularly updated with new training data, the boundary value can gradually become less effective overtime and also provide false positives. CNNs are known to be computationally intensive and expensive, which can impact the SDN architecture.
- **Future Suggestions:** To ensure the model is stays up to date and can adapt to evolving attacks, the ANIDS architecture should be regularly updated with new training data. The methodology can be extended to include behavioral analysis techniques to allow the ANIDS model to monitor the traffic and identify any discrepancies from the normal network behavior.

HFS-LGBM based IDS

- **Advantages:** The IDS proposed by authors of the paper utilizes the LightGBM ML algorithm for model training. This algorithm is beneficial for training due to its efficiency, less utilization of memory, speed of training, and ability to parallelly learn. Another advantage of LightGBM is that it can handle complex and huge datasets. The fusion of CFS and RF-RFE algorithms enhances the results of the feature selection process.
- **Disadvantages:** The HFS-LGBM based IDS is deployed directly on the SDN Controller. This can introduce a potential attack surface. If the SDN Controller becomes compromised, it could not only affect the IDS, leaving the network vulnerable to attacks, but also impact the overall SDN architecture. Additionally, the deployment of the complex IDS on the SDN Controller also increases the load, which could possibly lead to delay in network operations.
- **Future Suggestions:** To avoid failure of the controller and

decrease its load, the proposed IDS could be separated from the SDN Controller and deployed elsewhere within the SDN Network Architecture. In addition to the training dataset, the IDS can be trained with more diverse sets of data so that it remains effective against dynamic attacks.

IV. Conclusion

This paper discusses on the topic of Machine Learning-based Intrusion Detection Systems. Concepts of Intrusion Detection Systems, Machine Learning, Internet of Things, and Software Defined Networking are explored. We analyze and evaluate three ML-IDS built for IoT networks and three ML-IDS developed for SDN architectures. At the end of this survey, we provide future suggestions while discussing their drawbacks and benefits.

This survey aims to present how ML-IDS can refine traditional IDS in detecting attacks such as DDoS, R2L, Impersonation attacks, and so on. these models can also detect unseen threats.

V. Acknowledgements

I would like to thank Dr. Clifford Neuman for his valuable suggestions and feedback on this survey. I would also like to thank the teaching assistants of CSCI 530.

References

[1] S. Axelsson, "Intrusion detection systems: A survey and taxonomy," 2002.

[2] J. Anderson, *Computer Security Threat Monitoring and Surveillance*. 2002.

[3] E. Lundin and E. Jonsson, "Survey of intrusion detection research," 2002.

[4] J. Andress, "Chapter 1 - what is information security?," in *The Basics of Information Security (Second Edition)*, pp. 1–22, Boston: Syngress, 2014.

[5] F. Sabahi and A. Movaghar, "Intrusion detection: A survey," in *2008 Third International Conference on Systems and Networks Communications*, pp. 23–26, 2008.

[6] A. Pharate, H. Bhat, V. Shilimkar, and N. Mhetre, "Classification of intrusion detection system," *International Journal of Computer Applications*, vol. 118, pp. 23–26, 05 2015.

[7] M. Kubat, *A Simple Machine-Learning Task*, pp. 1–18. Cham: Springer International Publishing, 2017.

[8] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "Internet of nano-things, things and everything: Future growth trends," *Future Internet*, vol. 10, no. 8, 2018.

[9] A. H. Hussein, "Internet of things (iot): Research challenges and future applications," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, 2019.

[10] S. Badotra, "A review paper on software defined networking," *International Journal of Advanced Computer Research*, vol. 8, 03 2017.

[11] J. Asharf, N. Moustafa, H. Khurshid, E. Debie, W. Haider, and A. Wahab, "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, 2020.

[12] X. Yang, G. Peng, D. Zhang, and Y. Lv, "An enhanced intrusion detection system for IoT networks based on deep learning and knowledge graph," *Security and Communication Networks*, vol. 2022, p. 4748528, Apr. 2022.

[13] K. Sai Kiran, R. K. Devisetty, N. P. Kalyan, K. Mukundini, and R. Karthi, "Building a intrusion detection system for iot environment using machine learning techniques," *Procedia Computer Science*, vol. 171, pp. 2372–2379, 2020. Third International Conference on Computing and Network Communications (CoCoNet'19).

[14] M. A. Rahman, A. T. Asyhari, L. Leong, G. Satrya, M. Hai Tao, and M. Zolkipli, "Scalable machine learning-based intrusion detection system for iot-enabled smart cities," *Sustainable Cities and Society*, vol. 61, p. 102324, 2020.

[15] T. A. Tang, D. McLernon, L. Mhamdi, S. A. R. Zaidi, and M. Ghogho, *Intrusion Detection in SDN-Based Networks: Deep Recurrent Neural Network Approach*, pp. 175–195. Cham: Springer International Publishing, 2019.

[16] T. Issa and K. Tiemoman, "Intrusion detection system based on the sdn network, bloom filter and machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, 2019.

[17] Y. Hande and A. Muddana, "Intrusion detection system using deep learning for software defined networks (sdn)," in *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1014–1018, 2019.

[18] T. A. G. Logeswari, S. Bose, "An intrusion detection system for sdn using machine learning," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, pp. 867–880, 2023.