International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Advanced Persistent Threat Detection Performance Analysis Based on Machine Learning Models

Anil Kumar[1], Amandeep Noliya[2], Ritu Makani[3], Pardeep Kumar[4], Jagsir Singh[5]

Submitted: 18/09/2023 **Revised**: 17/11/2023 **Accepted**: 29/11/2023

Abstract: Advanced Persistent Threats (APTs) present a serious threat to modern cyber security, prompting research and evaluation of effective detection techniques. The on-going development of Advanced Persistent Threats (APTs) has motivated the investigation of novel strategies for preventing their malicious activities. The research presented here provides an in-depth investigation of machine learning-based APT detection techniques. APTs are explained in the beginning along with their features and the specifics of their attack models. By outlining their attack techniques and tactics, further analyse APTs. An extensive examination of APT attack detection strategies is covered in this study, with a focus on machine learning techniques. In the context of APT detection, Support Vector Machines (SVM), k-Nearest Neighbours (KNN), Deep Belief Networks (DBN), Decision Trees, and Convolutional Neural Networks (CNN) are considered. The underlying assumptions and applicability of each method for APT detection are evaluated. The performance study of the aforementioned machine learning approaches is the main goal of this research. To facilitate this, GiuseppeLaurenza/I_F_Identifier datasetis employed, which comprises a diverse range of network traffic scenarios. Different performance metrics, including precision, recall, F1-score, accuracy, true positive rate, and true negative rate, are employed to gauge the effectiveness of the detection techniques. The results unveiled in this study underline the superiority of Convolutional Neural Networks (CNN) over the other examined methods. The precision, recall, F1-score, accuracy, true positive rate, and true negative rate metrics collectively endorse CNN's prowess in accurately and comprehensively detecting APT attacks within network traffic. These findings not only contribute to the ongoing discourse on APT detection but also underscore the efficacy of CNNs in fortifying cyber security systems against sophisticated threats.

Keywords: *Enter APT, APT Machine Learning, SVM, KNN, CNN*

1. Introduction

Nowadays security is an essential part of every organization and agency. A few decades ago, it was only a major concern to the military and some government organizations. Every organization has its security threat monitoring, detection, and mitigation cell, which deals with the unknown threat which is dangerous to the organization. However, these monitoring cells are not sufficient as every day they received new types of malwares and a novel category of attacks [1][2]. There were times when an intruder or groups of intruder aim were to put down the organization and defamation or financial loss of the competitor's organization. Those attacks were single runs or short periods. But in the last few decades an attacker or organization of attackers, attack slow and low movements of attack to fulfil their achievements. They slowly lose the targeted organization and theft the use of full data without being trapped. These types of attackers or organizations of the attacker are called advanced persistent threat (APT) attacks. APT as its name specified that the attacker used advanced technology, methods, and tools for a persistent long time to mine the highly sensitive data of the target environment. APT intruders kept themselves low and gradually increase their foothold from a single computer to the entire network by accessing useful information and exporting authentication information on their end with a systematic approach [1][3].

APT is a fast-growing cyber security threat. APT is executed by well-trained, most experienced, financial, tools and resource-supported attackers and ambition the secret information of an organization. They attack as long as the funding agency wants or until they are caught [4].

1.1. What is APTs?

Define APT also known as an advanced persistent threat is an attack done by a trained and well-funded attacker. It is not done by the regular attacker. The funded agency

[1] *Reseach Scholar, Department of CSE, GJU S&T, Hisar and Assistant Professor of Computer Science, Indira Gandhi Govt. College, Tohana, Harayana, India*
anil.rohtagi@gmail.com
[2] *Assistant Professor, Department of Artificial Intelligence and Data Science, Guru Jambheshwar University of Science & Technology, Hisar, India*
noliyaaman@gjust.org
[3] *Associate Professor, Department of CSE, GJU S&T, Hisar, India*
ritu@gjust.org
[4] *Assistant Professor of Computer Science, Indira Gandhi Govt. College, Tohana, Harayana, India*
pardeep.phd@gmail.com
[5] *Assistant Professor of Computer Science, Indira Gandhi Govt. College, Tohana, Harayana, India*
erjagsirsingh18@gmail.com
Corresponding Author: noliyaaman@gjust.org

aims to acquire significant data on its target group or institution. APT is related to the information security name given military and accomplished by the nation-state. The acronym APT is made up of three words [1].

Advanced: The Advanced Persistent Threat attackers are very intelligent and able to evolve improved techniques and tools by using a multi-stage attack approach.

Persistent: The APT attackers are very persistent to carry out their ambition and make techniques to avoid getting detected. They follow the low and slow procedure.

Threat: The APT hacker specifically focuses on a particular organization to attain its aim. They generally have the capacity to harm an information system through denial of service, data exposure, destruction, and/or alteration of data. Generally, the hacker aims to achieve access to the selected and ex-filtrating information regarding the target from the compromised systems. Once they get access to the target organization, they stay a long time to collect the information useful for the funded agency. APT attacker frequently upgrades their techniques and involves different compromised nodes, not just one, dissimilar ordinary attack.

As per the National Institute of Standards and Technology (NIST) [2], an APT attacker group:

"(i) pursues its objectives repeatedly over an extended period.

(ii) adapts to defenders' efforts to resist it.

(iii) is determined to maintain the level of interaction needed to execute its objectives".

Chen et al. in [4], summarized the prime characteristics among ordinary attacks and Advanced persistent attacks in various features as illustrated in Table I:

Table I: Comparative difference between the ordinary and Advanced persistent attack

	ordinary Attacks	**Advanced persistent attacks**
Attacker	Primarily done by one person	Extremely organized, Experienced, strong-minded, and well-funded agency
Target	Undefined, single user or a single system	On targeted institutions, governmental bodies, economic or financial company
Purpose	Economic privilege, show skill and talent	ambitious lead planned profit
Approach	One-time implementation, appropriate, for a short while	Continue to make an effort, be slow-moving, and stay a long time,

1.2. APT Attack Model: how does APT work?

Advanced persistent attack attacks, as described above, are highly scheduled and well-coordinated to raise the possibility of an attack's accomplishment. And to meet they execute attacks in multiple stages. To know how APT works, the Author in [5] illustrated different attack tree and their advantage in executing the security of the system. This attack tree provides information on possible ways of attack happen and what security approach is applying the measure to identify and prevent the system from these attacks. APT attacks normally spread according to the following stage:

1. Reconnaissance: Any individual or group of APT attackers will start with reconnaissance. They described a survey and analysis of the intended organization at this phase. At this point, attackers gather data on the assets, staff, and connections of the target organization that can be used to reach the target. They comprehend the structure of the survey; the greater their understanding of the system being attacked, the greater the likelihood that the attack will succeed. They do a network scan to identify open services on the network, perimeter security measures in place, and users of the network who have access to the targeted data. The attacker gathers information about the employee during the reconnaissance stage, such as their interests, how often they visit websites, the social media platforms they use, and the information and communication architecture that the company uses, such as routers, hubs, switches, firewalls, network addresses, storage structures, security platforms, out-of-date systems, running machines, virtual hosts, etc. The attackers then use publicly available information in social networks where they can open accounts to create profiles for each target employee [6].

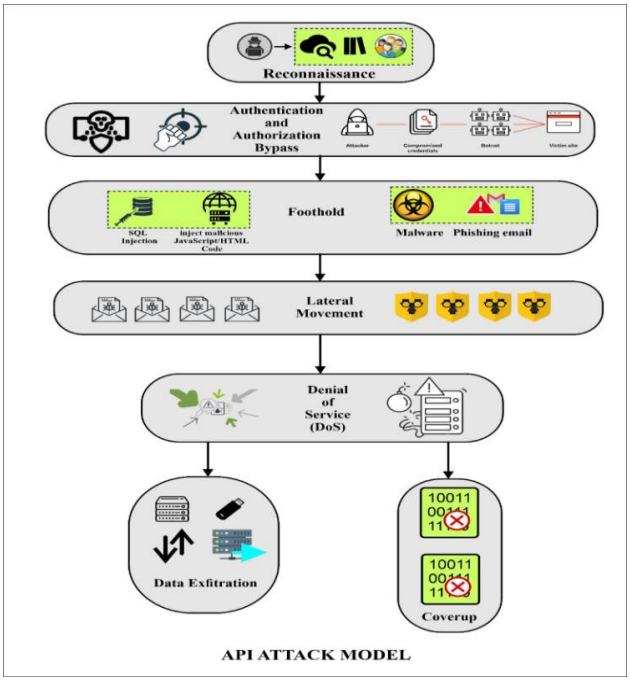


Fig 1 APT Attack Model

1) The attacker hopes to become more persistent and dig deeper into the system using this information.

2) Establish Foothold – By gathering information from reconnaissance, the attacker can successfully gain entrance into the computer networks of the targeted company. APT attackers use malware functionality in the computer system of the end user to exploit weaknesses.

I. The exploitation of known vulnerabilities: Common Vulnerabilities and Exposures List (CVE), Open Source Vulnerability Database (OSVDB) [7], and NIST National Vulnerability Database (NVD) [8] are a few well-known vulnerability databases that contain publicly revealed vulnerabilities, each of which is identifiable by a specific CVE-

ID. Additionally, in some circumstances, attackers can exchange and gather useful knowledge about discovered vulnerabilities in forums on the deep web and the black web [9]. According to a report published in [3], the bulks of Attacks by APT used known exploits. Hence, that is significant to install security fixes instantly when vulnerability has been discovered.

II. Malware: In 2016, there were 357 million distinct malware variants, and email infection rates decreased significantly from 1 in 220 emails in 2015 to 1 in 131 emails in 2016. According to Symantec, the botnets in charge of distributing spam campaigns are to blame for this increase in the virus. Our APT Attack Tree shows how malware can spread through spear-phishing, flash drives, and web access.

III. Spear-Phishing: The identical risk assessment from Symantec stated that attackers prefer targeted spear-phishing tactics, notably those that take the shape of Business Email Compromise scams, over the traditional mass-mailing phishing campaigns. Hacker uses common engineering or similar methods to gather data about the target group before sending emails that contain malware. These fraudulent emails have been skilfully created in such a way as to induce the receivers to open the attachments. By clicking on the add-on or link that heads to the installation and implementation of malware, employees who are uninformed of the infection may put the organization's network in danger [9]. When this malware is executed, it may take advantage of either known or undiscovered flaws to gain access to the organization's network.

IV. Zero-day vulnerability: A software issue known as a zero-day vulnerability is one that the maker of the computer software either doesn't know about or knew about but wasn't able to patch before an attacker could take advantage of that. APT hackers gather information regarding the organization's computer system, including the versions of the operating system, updates applied, and applications implement on those systems, such as anti-virus and anti-malware software. The next step is to find any security holes in those versions that an attacker could exploit to break into the target's network. However, the study in [3] indicates that the bulk of APT assaults was part of existing exploits and that only a small number of APT operations were carried out and accomplished through zero-day vulnerabilities. In [10], Lee and Lewis concentrated on APT carried out using malware delivered via emails. After looking at several emails that contained binaries, the authors developed a solution that entails building an undirected graph with nodes that stand in for email addresses and edges that represent the email communications that link the computers. They hoped that this chart would deliver them additional material that would be useful in investigating the selected malware. The issue with this approach is that there might be various attack nodes short of a little connection to other attacks, which could indicate that the targets of the attacks were not visible enough, or there might be distinct attacks that call for additional research before establishing the survival of an APT attack [11].

V. Web Download: As was previously noted, spear-phishing electronic messages may contain harmful documents that require being unlocked or links to a dangerous internet site that causes employees to unintentionally download a virus when they visit. Additionally, attackers may take advantage of flaws on one or more of the websites that the targeted employees visit, causing the employees to unknowingly download malware when they visit the site. The watering-hole attack is the name of this latter assault strategy.

3. Lateral Movement - After gaining access to the system,

they must find the sensitive node(s) that contain sensitive information or are essential parts of the organization's infrastructure, which necessitates that they travel further into the network. lateral data collecting movement by leveraging the account information gathered from priority users at the previous level, operators try to maintain access to the target information at this stage. Attackers create unnecessary C&C channels using sophisticated tools if there are rapid modifications to the organization's security configuration. One or more unnecessary copies of the target information are made in servers serving as "staging points" when access to the target information is achieved. Before being exfiltrated, data is segmented, defragmented, and encoded at this point. The attacker attempts to access other hosts on the target network since at this point, access to higher privileged rights is required to access crucial resources [1][3][6].

4. Exfiltration - At this point, data that has been gathered and packed on staging point servers is sent across encrypted channels to some other servers that function as drop points. When the attackers' goal is to obtain the organization's data, the procedures that include obtaining and delivering the data to the attackers' command and control center fall under this stage. At this point, the attacker manages one or more remote servers where the stolen data is transferred. If an attacker wants to steal data repeatedly, they can either leak it surreptitiously and slowly over time or totally at once [6].

5. Cover Up - An APT attack would only be successful and finished if the attackers and the organization or institution that funded the attack could not be located, identified, or tracked, necessitating the removal of any evidence that might link them. We go into further depth about each of these stages and the various assault vectors in the next sections of this section. The paradigm of an APT assault carried out in the aforementioned various stages is shown in Figure 2.

1.3 APT Attack Detection Methods

To identify and halt each stage of an APT attack at numerous places and across various network layers, a defense-in-depth approach with sufficient Defence mechanisms must be used. Any institution or individual can be secured against APT attempts by associating the information with these various protection methods. Surveillance methods depend mostly on the concept that even if intruders can prevent detection by one of the many defences which are being used, there is still another level of resistance that they should ignore. A strong defense with a systematic approach must verify that none of the protective stages could be broken. These layered defensive schemes additionally provide attackers time and a risk estimate, which would help them in adopting a mitigation strategy. The monitoring methods, detection systems, deception techniques, and mitigation procedures are the four classifications into which we classify APT protection techniques. As each one of these classifications will be extensively examined, every classification or subclass can be further subdivided into additional categories [6].

1. Monitoring Methods

Monitoring the whole information system is the first and most fundamental step in fighting against APT. No entrance point into the network should be unsupervised and should be examined from multiple aspects and stages.

I. Disk Monitoring: Every end device and element of the organization's network must be monitored for any malicious threats employing access control lists, firewalls, and antivirus programs as necessary. By eliminating existing vulnerabilities that could otherwise transmit the virus to vulnerable systems within the system, applying patches as required to the software operating on the system will help in decreasing the entry points for an attacker. Additionally, keeping an eye on each of these end systems' CPU utilization will assist in identifying any suspicious activity at the end system level [8].

II. Disk Monitoring: Processing from inside the end system's memory rather than from a file is among the techniques malware might evade detection. So this attack virus runs itself to use a program that is already running in the Main memory. Besides the unusual memory utilization by a program that can be identified if monitored, it produces nothing to record in the backdrop as there is no separate process running.

III. Packet Monitoring: The interaction with the Centre of Command and Control is probably the most important element of an APT attack. Conversation with the Command-and-control server occurs commonly more than once, usually starting whenever the system has been undermined and proceeding after that when file transfer is required. It might be possible to detect any unusual behavior within the end system by keeping an eye out after that component level for any data packet having new location IP addresses, a packet containing large sizes of data, and packets transmitted in massive quantities to an identical destination IP address.

IV. Code Monitoring: It's like seeking the dream to develop bug-free software programming. You couldn't ever promise that any program you create or any coding you publish will be error-free. Even if it can be difficult to guarantee that the code is error-free whenever executing in various contexts, it is impossible. Threat actors use such bugs as a method of system entry. Although a few of them may have been identified just before the software launch, there will always be a possibility of unexpected vulnerabilities. A threat might be detected significantly earlier before it spreads to other computers if the code is

monitored at the end-user level for its performance and to guarantee that it executes within its scope without using unexpected resources or taking up memory areas that aren't usually accessible [9].

V. Logs Monitoring: Logs are a crucial component not only for forensic investigation but when used effectively, they may also support early-stage threat detection or even mitigation. Instead of just having the specific logs, which frequently finish up in a stack to be subsequently decided to search for proof of an intrusion, the relationship of such log files, including memory utilization logs, Cpu utilization logs, program execution logs, and event logs, might give extensive significant amounts of data that will also understand and support attempts to defend networks or systems against anomalous behavior.

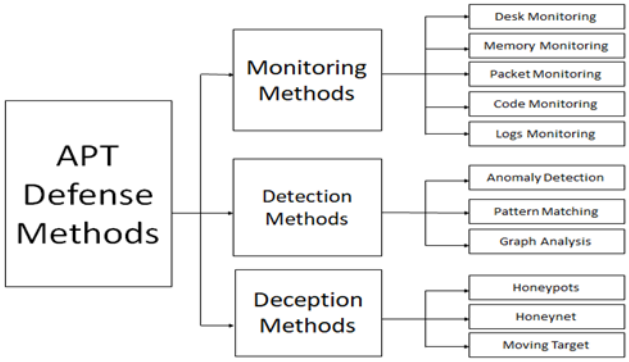


Fig 2 APT Defense Methods

2. APT Defence Methods

Anomaly-based detection, pattern-matching detection, and graph-based detection are three categories into which APT detection methods can be characterized.

I. Anomaly Detection: The ability to respond to the defender's efforts to counter it constitutes one of the key attributes of an advanced persistent threat. And to counter such a threat, the used prevention strategies must pick up on and adjust to the offenders' actions. These techniques should include gathering information from many sources, studying from that information, and predicting the future based on that knowledge to determine and address potential future actions.

II. Pattern Matching: A common method used by intrusion detection and prevention systems is pattern matching. However, this method offers benefits to itself. Inappropriate behavior could be determined by looking for patterns in a process's or user's activity. a method for applying structured intrusion detection to find APTs. Some pattern-matching strategy is built on high-level structured data which has been acquired in network traffic time-series data. Pattern matching methods use Natural Language Processing (NLP) for activity recognition in remote sensing and geographical information challenges [11].

III. Graph Analysis: Among the domains that are currently explored is graph analysis since it can enhance the analysis of complicated organizations and identify malicious activities. Johnson et al. [68] offered a creative way to use graph analysis to assess a cyber-network's susceptibility. Their technique particularly identifies threats that would employ Move the hash approaches to involve lateral movement and privilege escalations to accomplish the attack objective. A straightforward metric that assesses on a graph the probability that a node can be accessed from some other randomized location, possibly exposing the network sensitivity, can be utilized to detect the threat.

3. APT Deception Methods

The degree of complexity being used to carry out the attacks is an important feature of APT attacks. We cannot promise that every one of the existing protective factors will safeguard us against malicious software and emerging system vulnerabilities. However, one of the finest ways to safeguard yourself is to be well-prepared. Therefore, the power for deception appears. In this defense approach, the defenders confuse the intruders by constructing traps in the style of forged information or networks or platforms that replicate the organization's production conditions but aren't a component of this one.

I. Honeypots Defence Approach

The first tool used for deception was honeypots. Beginning in the 1990s, Information cyber security experts employed them to confuse cybercriminals who already had acquired unauthorized access to the network into engaging with a fake system. Honeypots might obtain information and analyze the actions of the hackers in this way. They weren't designed to identify threats. Nevertheless, a lot has happened since honeypots were invented, including deception technologies. It has been discovered that honeypots are completely ineffective at detection due to their regular scope limitations and ease of detection by skilled malicious actors. Successful hackers rapidly conclude that they are fake. The deception technology of today has a lot of potential, particularly for quick and accurate threat detection. Deception must go well beyond the honeypot, though, to fully achieve that potential.

II. Honeynet (A Network of Honeypots) Defence Mechanism

A fake system or honeynet is a collection of honeypots. It is maintained on just a few servers, each of which represents a different environment, even though it looks like a real network

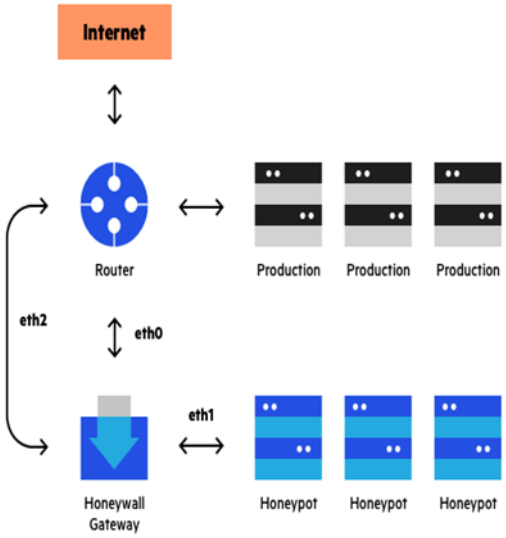


Fig 3. Honeynets a Network of Honeypots

and contains many different systems. A "Honeywell" monitors network traffic entering and exiting the device and directs it to honeypot implementations. Security holes might be introduced to a honeynet to make it easier for a hacker to get to the trap. A point of entry for attackers might be any system on the honeynet. The honeynet collects information on the intruders and propagates them from the real network. A honeynet has the benefit of feeling more natural and having a wider coverage area than a basic honeypot. Since it provides attackers with a different internal network that may be an attractive alternative to the actual one, a honeynet is a preferable alternative for large, complicated networks.

II. Moving Target Defence: Moving Target Defence is a highly promising information security technique for APT mitigation approach which breaks the unauthorized access privileges for protracted periods. To make it more complicated and difficult for attackers, to reduce the disclosure of weaknesses and attack chances, and to boost system reliability, moving target defense [12] provides us to design, analyze, test, and deploy a variety of techniques and strategies. Moving Target Defense (MTD) is the act of moving system parts repeatedly and independently. As a result, the system becomes more difficult (or expensive) for the attacker to efficiently attack the system. There are two ways to classify MTD i.e. security modeling and stack protocol implementation. For further, on the fundamentals of security modelling, MTD categories into shuffle, diversity, and redundancy. Protocol stack MTD is described as network level, host level, and application level.

2. Machine Learning for APT Detection

Techniques like anomaly detection, behavioural analysis, predictive analytics, threat intelligence, and automation can be used to apply machine learning to APT detection. Machine learning algorithms can identify prospective APT attacks and assist organizations in responding to them in a rapid and efficient manner by analysing immense quantities of data and learning from prior trends. As a result, organizations may be able to better safeguard their systems and data from complex cyber threats and boost APT detection capabilities [5].

2.1. SVM for Advanced Persistent Threat Detection

In the field of cyber security, support vector machines (SVMs) can be an effective method for identifying advanced persistent threats (APTs). APTs are a kind of sophisticated cyber-attack created particularly to bypass conventional security measures and go unnoticed for extended periods of time. Through the detection of unusual patterns in network traffic or other data sources, SVMs can aid in the detection of these assaults. A decision border and a margin are the two primary parts of an SVM model. In the context of APT detection, the decision boundary divides the data into two classifications that typically correspond to legitimate and malicious behaviour. The SVM algorithm searches for the margin, also known as the decision border that maximizes the distance between the nearest data points of each class during training. This method of classification is known as maximal margin classification, and it is predicated on the idea that the wider the margin, the less likely it is that the model will over fit or incorrectly categorise incoming data. To achieve maximum classification, the SVM algorithm uses a kernel function to move the input information towards a space with more dimensions where it is simpler to distinguish between the data. SVMs frequently use the kernel functions linear, polynomial, radial basis function (RBF), and sigmoid. Based on the placement of fresh, unused data points in relation to the decision boundary and margin, the SVM model can be used to forecast the class of previously unknown data points after training. If a data point falls within the margin or on the wrong side of the decision boundary, it is classified as malicious and flagged as a potential APT.

Here is a mathematical model of SVM for APT detection:

Given a training set of labelled data points, $\{(x_1,y_1), (x_2,y_2), ..., (x_n, y_n)\}$, where x_i is the feature vector of the i-th data point and y_i is the corresponding label (1 for APT and -1 for normal), the goal of SVM is to find the hyper plane that maximally separates the two classes. The hyper plane is defined as:

$$w^T . x_i + b = 0\text{-------(i)}$$

where w is the weight vector of size m x 1, b is the bias term, and x_i is the ith data point.

The distance between a data point x_i and the hyperplane

can be computed as:

$$d(x_i) = |w.\ x_i + b| / ||w||\text{----------(ii)}$$

where ||w|| is the Euclidean norm of the weight vector.

The distance between the hyperplane and the nearest data point is referred to as the margin of the hyperplane. The hyperplane that maximizes the margin is the perfect hyperplane. The following sentence can be employed for defining the SVM optimisation problem:

$$\text{minimize } 1/2\ ||w||^2\text{----------(iii)}$$

$$\text{subject to } y_i(w.\ x_i + b) >= 1 \text{ for } i = 1, 2, ..., n\text{-----(iv)}$$

where the constraint guarantees that all data points are accurately identified with a margin of at least one, and ||w|| is the weight vector's Euclidean norm. Quadratic programming can be used to address this problem. New data points can be categorised by determining which side of the ideal hyperplane they fall on after the hyperplane has been identified:

$$f(x) = \text{sign}(w*x + b)\text{----------(v)}$$

where the sign is the sign function, which, depending on the argument's sign, returns +1 or -1. The features taken from network traffic, such as packet length, protocol type, and other characteristics, would be the input vectors x in the context of APT detection. Based on these characteristics, the SVM would then learn to categorise the traffic as either benign or malicious.

The following rule can be applied to the SVM to predict the class of new data points:

$$\text{if } w^T x_i + b > 0, \text{ then } y_i = 1 \text{ (APT)--------(vi)}$$

$$\text{if } w^T x_i + b < 0, \text{ then } y_i = -1 \text{ (normal)--------(vii)}$$

2.2. KNN for Advanced Persistent Threat Detection

Advanced persistent threat (APT) detection can be done using the machine Learning K-nearest neighbour (KNN). In APT detection, where there may be a limited number of known threats, KNN's capacity to be trained using a small number of labelled instances is an essential advantage. By manually updating the system on newly acquired information, KNN may also be rapidly modified to identify new threats. KNN may be computationally expensive for huge databases and may handle high-dimensional or noisy data less well than other algorithms, like SVM or deep learning.

The idea of distance between data points serves as the foundation for the mathematical formula used in KNN for advanced persistent threat (APT) identification. In order to identify the K nearest neighbors of a new threat and estimate its label based on those neighbours' labels, the technique known as KNN uses the distances among data points.

The KNN model is stated in the following way:

The KNN algorithm determines the name of a new threat x by performing the following: Given a dataset D of n data points represented as (x1, y1), (x2, y2),..., (xn, yn), where xi indicates the attributes of the i-th data point and yi denotes its label. Use a distance metric, such as the Manhattan distance or Euclidean distance, to determine the separation between each data point in D and x. Let the distances between x and each data point be d1, d2,…d_n. According to their distances, select x's K adjacent neighbours. Let N be the collection of x's K closest neighbours. Determine the likelihood of each label in N. Let pi represent the total number of data points with label i in N.

Pi/K is the chances that x fits label i. Give x the label with the greatest likelihood.

The formula for utilizing the Euclidean distance to determine the separation between two data points, x and y, is as follows:

$$d(x, y) = \text{sqrt}(\text{sum}((x_i - y_i)^2))\text{----------(viii)}$$

where xi and yi, respectively, represent the features of x and y.

The following equation can be used to find out the distance between the two points of data, x and y, using the Manhattan distance:

$$d(x, y) = \text{sum}(|x_i - y_i|)\text{----------(ix)}$$

where x_i and y_i are the features of x and y respectively.

2.3. Deep Belief Network for Advanced Persistent Threat Detection

DBNs are very useful for detecting APTs because they can spot patterns in huge, complicated datasets, irrespective of whether the data is noisy or lacking. Each layer of the network's hidden units learns to represent more complex aspects of the input data. The network is made up of several layers in total. The network can recognize patterns in the data without labelled examples because it was trained using unsupervised learning techniques.

The following is an illustration of the mathematical model for a DBN:

Let y be a binary output variable and X be a d-dimensional input vector. L binary Restricted Boltzmann Machines (RBM) and a final logistic regression layer can be used to represent a DBN with L layers.

Each layer k has a bias vector bk, a weight matrix Wk, and nk binary units. An RBM with weights Wk and biases bk is said to have the following energy:

$$E(X, h) = -\sum_{i,j} Wk_{ij} X_i h_j - \sum_i b_i X_i - \sum_j b_j h_j \text{ -------(x)}$$

where Wk_{ij} is the weight between input unit i and hidden unit j and Xi and hj are the binary states of the input units and hidden units, respectively. The input and hidden units' interactions are shown in the energy function's first term, while their biases are shown in the second and third terms, respectively. The following is a definition of the combined probability distribution of the visible and hidden units:

$$P(X, y) = 1/Z \exp(-E(X,y)) \text{ ----------(xi)}$$

Where Z is the partition function, which guarantees normalisation of the distribution. The probabilities of the output layer are then calculated using the hidden layer's activation and are provided by:

$$p(y \mid x) = \text{SoftMax}(W(L+1) * h(L) + b(L+1)) \text{ ---------- (xii)}$$

where SoftMax is a normalisation function that makes sure the probabilities add up to 1, and $p(y \mid x)$ is the probability of output y given input x. Unsupervised pre-training, a method for training the DBN, involves learning the weights and biases of each layer separately using a restricted Boltzmann machine (RBM). Following the completion of the pre-training, the DBN can be improved using supervised learning strategies such back propagation. A greedy layer-wise pretraining phase in which each layer is trained as an RBM and a fine-tuning phase in which the entire network is adjusted using backpropagation are the two stages that make up the training of a DBN. To increase the likelihood of the training data, the weights and biases of each layer are learned during pretraining. The network is trained to minimize a cost function, such as cross-entropy loss, during fine-tuning.

The DBN can be used for a variety of tasks after training, including classification, regression, and clustering. The DBN can be used to detect anomalies in incoming network traffic for enhanced persistent threat detection after being trained on a dataset of typical network traffic. Any incoming communication that deviates noticeably from these qualities can be flagged as possibly malicious by the DBN, which can learn a set of features that are typical of normal network traffic.

2.4. Decision Tree for Advanced Persistent Threat Detection

Advanced persistent threats (APTs) can be found using decision trees as a machine learning technique. The algorithm is trained on a labelled dataset of known APTs and non-APTs in order to employ decision trees for APT identification. Based on the traits and traits of the APTs and non-APTs in the training dataset, the decision tree then develops a set of rules. These rules are used by the algorithm to determine if a new sample is an APT or not during the testing phase when it is given to the decision tree. In order to reach this conclusion, the decision tree takes into account a number of characteristics of the sample, including its network traffic patterns, system log files, and behavioural patterns. Because they can manage vast volumes of data and may spot patterns in the data that may be challenging for humans to notice, decision trees are particularly helpful for APT detection. Decision trees are a helpful tool for continuing APT detection and prevention since they may be changed over time when new APTs are found.

One possible mathematical equation for building a decision tree for APT detection is:

IF [Feature 1] <= [Threshold 1] AND [Feature 2] <= [Threshold 2] AND ... AND [Feature n] <= [Threshold n] THEN [Classify as APT]

ELSE [Classify as Not APT]

In this equation, [Feature 1], [Feature 2], ..., [Feature n] represent the key features or conditions that we use to identify an APT. Each feature is associated with a threshold value ([Threshold 1], [Threshold 2], ..., [Threshold n]) that determines whether the feature is present or absent.

2.5. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) is a technique for deep learning that can be applied to cyber security to detect advanced persistent threats (APTs). Although CNNs were created primarily for image processing tasks, they can also be utilized for other sorts of data, such as traffic on networks sequences or system call patterns. Convolutional, pooling, and fully linked layers of neurons make up the various levels of the CNN algorithm. The pooling layers are used to lower the size of the feature maps, while the convolutional layers are in charge of identifying features in the input data. The input data is categorized using the learned characteristics using the fully linked layers.

The CNN algorithm can be trained on a dataset comprising both benign and malicious data in the context of APT detection, with the features of the dataset serving as the CNN's inputs. Through the processing of the input data through the convolutional and pooling layers, the CNN learns to recognize the patterns and features that are indicative of benign and harmful material. The data is subsequently categorized as either benign or harmful by the fully connected layers. For APT detection, the CNN algorithm

has a number of benefits. It is capable of handling high-dimensional data and is proficient at locating features important for APT detection. Additionally, because there aren't many known APTs, CNNs can be trained on vast datasets, which is particularly helpful in the context of APT detection.

A mathematical model of a CNN for APT detection could involve the following components:

1. Input layer: This layer takes in the input data, which could be network traffic or system logs. The data is usually represented as a matrix or tensor.

2. Convolutional layer: This layer uses filters to gather features from the input data. The filters that are used are typically tiny matrices that move through the input data and multiply and add the elements one by one.

3. Activation layer: The convolutional layer's output is applied to the activation layer's non-linear activation function in this layer. To provide non-linearity to the model and enable the network to learn more complicated features, the activation function is used.

4. Pooling layer: By down sampling the feature maps, the pooling layer lowers the dimensionality of the output from the activation layer. This aids in lowering the model's computing expense and enhancing its resistance to noise and changes in the input data.

5. Fully connected layer: This layer translates a collection of output classes to the output from the preceding layer. A SoftMax activation function that normalizes the output scores into a probability distribution over the output classes often follows the fully connected layer.

6. Output layer: As many neurons as there are APT classes to categories would be present in the output layer, which would employ a cross-entropy loss function to determine the error between anticipated and real APT class labels.

7. Loss function: This function is used to calculate the discrepancy between the model's anticipated output and the actual output. The model's objective during training is to reduce this loss function.

8. Training: In order to minimize the loss function, the model would be trained using backpropagation and gradient descent. Batches of labelled data would be fed through the network during training, and the model's weights and biases would be changed to reduce loss.

9. Evaluation: The trained model would be evaluated on a separate test set to measure its performance. The performance metrics used could include accuracy, precision, recall, and F1 score.

10. Optimization algorithm: The model's parameters are updated using this approach during training in order to reduce the loss function. The stochastic gradient descent algorithm and its variations are well-known optimization algorithms.

A model could analyses network traffic data and find trends that might point to the presence of an APT by combining these elements. It should be understood, nevertheless, that the model's efficacy would depend on the caliber and variety of the training data as well as the specific APT threats being targeted. Convolutional neural networks (CNNs), a subset of deep learning algorithms, have been applied to cyber security's identification of advanced persistent threats (APTs). APTs are a type of cyber assault created particularly to obtain sensitive data while going unnoticed for an extended period of time.

The mathematical equation of a CNN can be broken down into the following steps:

Convolution operation: A set of learnable filters (also known as kernels) are convolved with the input image to create a set of feature maps. The mathematical representation of the convolution operation is:

The mathematical equation for a basic CNN can be written as follows:

$$y = f(W * x + b) \text{ ----------(xiii)}$$

Where W is a collection of learnable weights (also known as kernels or filters), y is the CNN's output, f is an activation function (such as ReLU or sigmoid), and f is an activation function that is applied to the input data. A single image or a series of images may be used as the input data for the convolution operation, which involves swiping the kernel over the input data and computing dot products between the kernel and local patches of the input. If the input data is network traffic data, x may represent packets or flows.

Convolution operation

$$F(i,j) = \sum x \sum y\ I(i+x-1, j+y-1) * w(x,y) \text{ ---------(xiv)}$$

where F(i,j) represents the output of the convolution operation at location (i,j) in the feature map, I(i+x-1, j+y-1) represents the pixel value at location (i+x-1, j+y-1) in the input image, and w(x,y) represents the weight (or kernel coefficient) at location (x,y) in the filter.

Activation function: An activation function is applied element-wise to the output of the convolution operation to introduce non-linearity into the model. The most commonly used activation function is the Rectified Linear Unit (ReLU), which can be mathematically represented as:

$$\text{ReLU activation function } f(x) = \max(0, x) \text{ --------(xv)}$$

where f(x) represents the output of the activation function for input x.

Pooling operation: The feature maps are down-sampled using a pooling operation to reduce the spatial

dimensionality of the data and make the model more computationally efficient. The most commonly used pooling operation is the max pooling operation, which selects the maximum value within a sliding window. The max pooling operation can be mathematically represented as:

The notation P (i, j) represents the output of the max pooling operation at location (i, j) in the down-sampled feature map. The location (i, j) corresponds to the top-left corner of the rectangle that the max pooling operation is currently considering.

The size of the rectangular tly region is determined by the size of the max pooling filter. For example, if the max pooling filter has size (x, y), then the rectangular region being considered by the max pooling operation at location (i,j) is the submatrix of the input feature map defined by the indices (i+x-1, j+y-1), (i+x-1, j), (i, j+y-1), and (i, j). The output value P(i,j) is simply the maximum value within this rectangular region. Specifically,

P(i, j) is given by:

$$P(i, j) = \max\{F(i+x-1, j+y-1),$$
$$F(i+x-1, j), F(i, j+y-1), \quad F(i,j)\} \text{------(xvi)}$$

where F(i,j) represents the value of the input feature map at location (i,j).

3. Performance Analysis

The purpose of this study is to examine the performance of the various machine learning techniques to detect the advanced persistent threat detection. In this study we have used Support vector machine learning, deep belief network, KNN, Decision tree and Convolution neural networks for analysis. The dataset considered for the performance of machine learning model is GiuseppeLaurenza/I_F_Identifier dataset [27].It has been used to evaluate the performance of various machine-learning algorithms and techniques for detecting and classifying network attacks. All the standard data mining process such as data cleaning, pre-processing, normalization, visualization and classification were implemented in the python. The training data set were normalized from 0 and 1.

4. Result and Discussion

4.1. Accuracy

The accuracy ratings for various machine learning algorithms are shown in fig. 4, which provide helpful information about how well they perform on a categorization challenge. With the highest accuracy of 0.894, the Convolutional Neural Network (CNN) takes the lead and demonstrates its extraordinary capacity for accurate prediction. With an accuracy of 0.887, the Artificial Neural Network (ANN) comes in second place, proving its dependability. With accuracy values of 0.875 and 0.883, respectively, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) both perform wonderfully, highlighting their usefulness in categorization. With an accuracy of 0.874, the Decision Tree (DT) method is not far behind. With an accuracy of 0.868, Bayesian Learning (BL), on the other hand, trails somewhat and offers room for growth. In summary, CNN and ANN excel in accuracy, followed by SVM, KNN, and DT, while BL shows room for enhancement in making precise classifications.

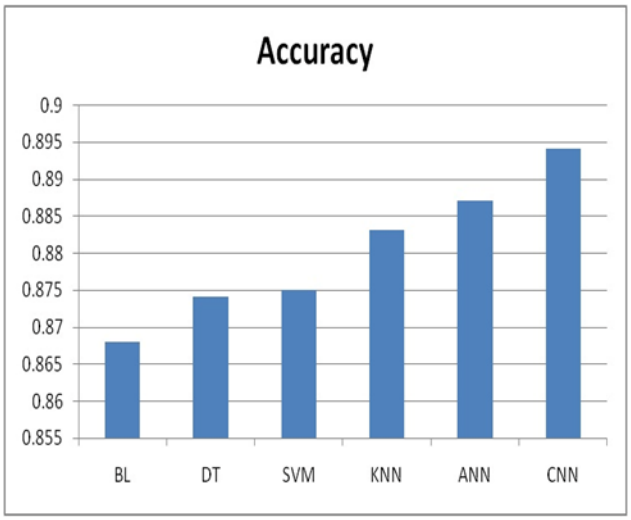


Fig 4 Accuracy of various ML Algorithms

4.2. True Positive Rate

In fig. 5, The True Positive Rate (TRP) rankings show how well machine learning systems can recognize positive examples. In this case, the TRP scores displayed by each algorithm vary. The Artificial Neural Network (ANN), which has a TRP of 0.84, is closely behind the Convolutional Neural Network (CNN), which has the highest TRP of 0.843. With TRP scores of 0.825, 0.818, and 0.838, respectively, Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) likewise perform admirably. With CNN and ANN excelling in this area, these results show the efficiency of all algorithms in identifying positive events.

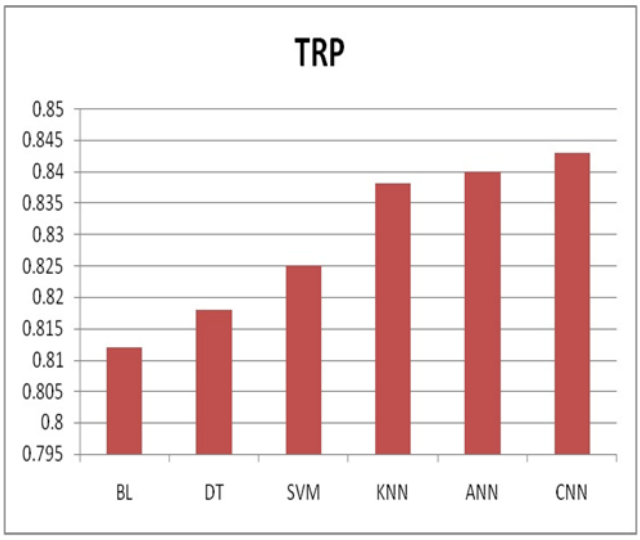


Figure 5 True Positive Rate of various ML Algorithms

4.3. True Negative Rate

In the fig. 6, In a classification job, the True Negative Rate (TNR) scores show how well machine learning systems accurately detect negative cases. All algorithms present different TNR scores in this circumstance. The Artificial Neural Network (ANN) is closely behind the Convolutional Neural Network (CNN), which has the greatest TNR at 0.866. K-Nearest Neighbors (KNN), Decision Tree (DT), and Support Vector Machine (SVM) all perform admirably, with TNR scores of 0.848, 0.844, and 0.861, respectively. These findings demonstrate the potency of all algorithms in identifying bad events, with CNN and ANN excelling in this area.

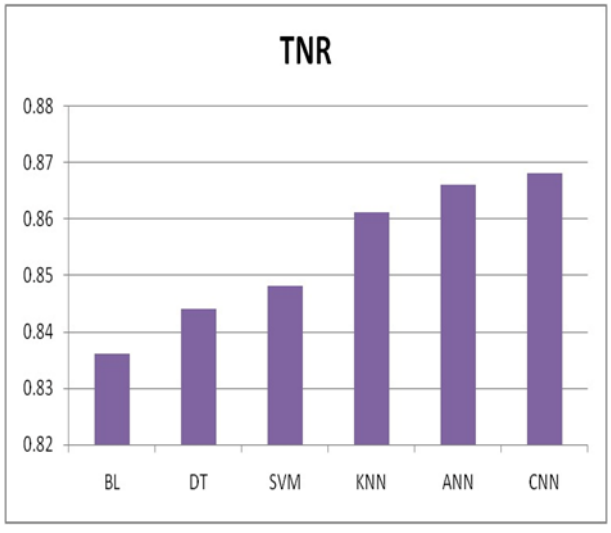


Fig 6 True Negative Rate of various ML Algorithms

4.4. False Positive Rate

The False Positive Rate (FPR) ratings demonstrate in fig. 7, the effectiveness of machine learning algorithms in preventing false positive mistakes in classification tasks. All algorithms exhibit different FPR ratings in this situation. With the lowest FPR of 0.132, which highlights its capacity to reduce false positive classifications, the Convolutional Neural Network (CNN) stands out. The Artificial Neural Network (ANN), which has an FPR of 0.134 and is highlighted for its dependability in handling false positives, comes in second place. With FPR scores of 0.152, 0.156, and 0.139, respectively, The performance of the Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) is likewise excellent, demonstrating their capability to lower false positive errors. These results demonstrate that all algorithms are effective at reducing false positives, with CNN and ANN standing out in this regard.

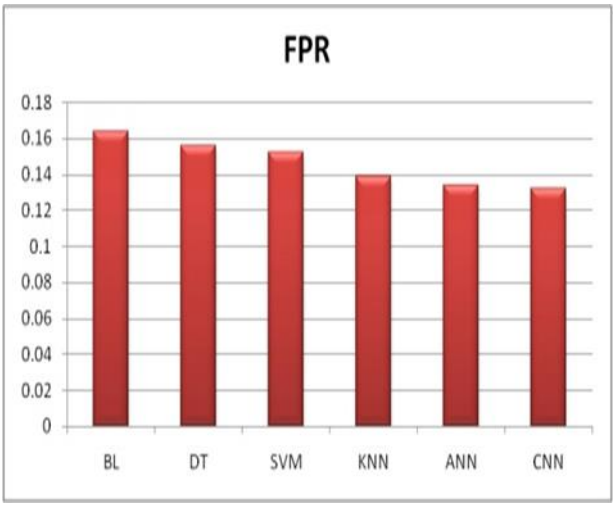


Fig 7 False Positive Rate of various ML Algorithms

4.5. False Negative Rate

In fig. 8, The False Negative Rate (FNR) ratings give information about how well an algorithm can detect positive examples in a classification test. In this situation, the FNR ratings of all methods differ. With a FNR of 0.157, the Convolutional Neural Network (CNN) shows the lowest FNR, demonstrating its accuracy in identifying positive cases. With a FNR of 0.16, the Artificial Neural Network (ANN), which follows it closely, emphasizes its dependability in this area. With FNR scores of 0.175, 0.182, and 0.166, respectively, Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) similarly perform well in terms of recognizing good situations. With CNN and ANN excelling in this area, these results show the efficiency of all algorithms in identifying positive events.

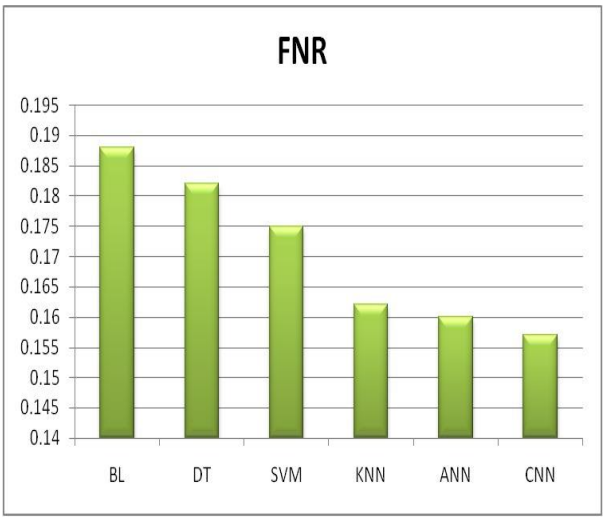


Fig 8 False Negative Rate of various ML Algorithms

4.6. Recall

The algorithms' abilities to accurately identify positive cases in a classification challenge are revealed by the Recall scores which is shown in fig. 9. Recall ratings differ among all algorithms. With a Recall of 0.843, the Convolutional Neural Network (CNN) takes the lead and demonstrates its superior ability to recognize positive situations. The Artificial Neural Network (ANN), which follows it closely and has a Recall of 0.84, stands out for its dependability. With Recall scores of 0.825, 0.818, and 0.838, respectively, Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) likewise exhibit good performances, highlighting their proficiency in detecting favorable cases. These findings suggest that all algorithms successfully identify positive examples, with CNN and ANN standing out in this regard.

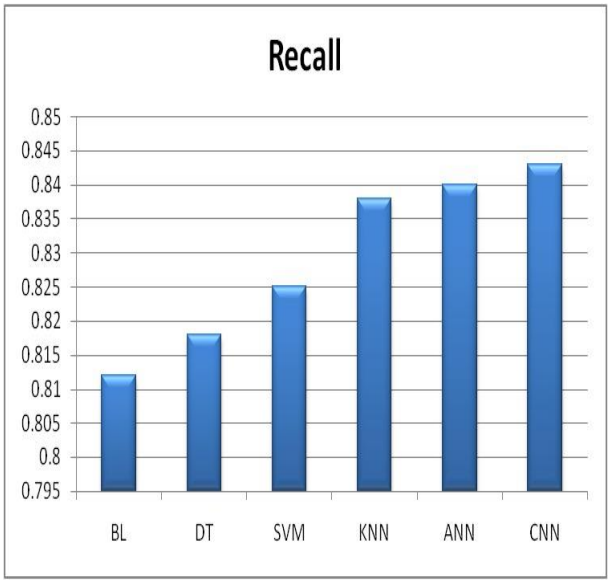


Fig 9 Recall of various ML Algorithms

4.7. Precision

The algorithms' capacity to make precise positive predictions in a classification test is revealed by precision scores shown in fig. 10. All algorithms exhibit different precision ratings in this situation. With a precision score of 0.864, the Convolutional Neural Network (CNN) takes the lead and demonstrates its ability to generate accurate positive class classifications. The Artificial Neural Network (ANN), which has a precision of 0.862 and emphasizes its dependability, follows it closely. With precision scores of 0.844, 0.839, and 0.857, respectively, Support Vector Machine (SVM), Decision Tree (DT), and K-Nearest Neighbors (KNN) also perform well, highlighting their prowess in producing precise positive predictions. These findings show that all algorithms are capable of making accurate positive class predictions, CNN and ANN standing out in this regard.

4.8. F-1 Score

In fig. 11, F1 scores show how well precision and recall are balanced in machine learning systems. With F1 ratings of roughly 0.853 and 0.851, respectively, the Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) shine in this analysis, demonstrating their prowess in balancing accurate positive predictions and proper identification of positives. With F1 scores at 0.835, 0.847, and 0.828, respectively, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and Decision Tree (DT) also perform well, highlighting their proficiency in reaching this balance. With an F1 score of roughly 0.822, Bayesian Learning (BL) performs brilliantly. These findings show how these algorithms balance precision and recall in a variety of interesting yet practical ways.

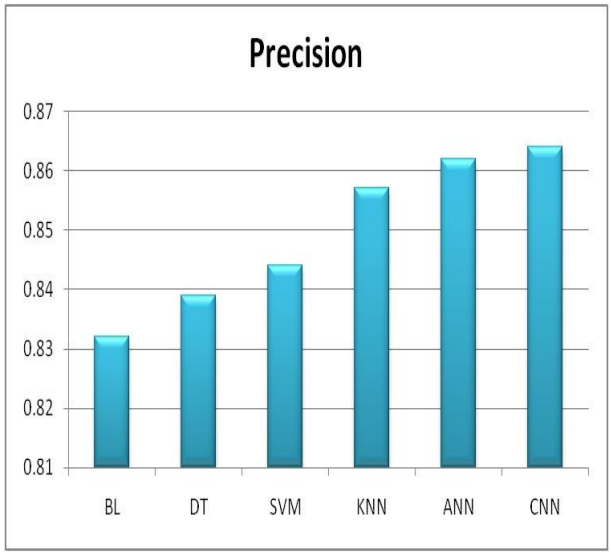


Fig 10 Precision of various ML Algorithms

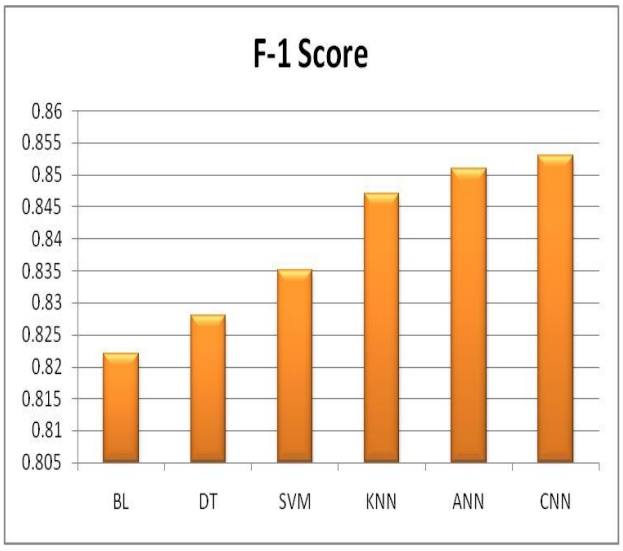


Fig 11 F-1 Score of various ML Algorithms

5. Conclusion

In the relentless battle against Advanced Persistent Threats (APTs), the development and evaluation of effective detection techniques are paramount. This paper has undertaken a comprehensive analysis of APT detection methods, with a specific focus on machine learning approaches. Through a thorough exploration of APTs, their attack models, and various detection methodologies, this study has shed light on the complexities and challenges posed by these persistent and sophisticated cyber threats. The assessment of machine learning techniques for APT detection, including Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Deep Belief Networks (DBN), Decision Trees, and Convolutional Neural Networks (CNN), has provided valuable insights into their efficacy within the context of cyber security. The evaluation, conducted on the NSL-KDD dataset, encompassing diverse network traffic scenarios, has enabled a robust comparison across key performance metrics such as precision, recall, F1-score, accuracy, true positive rate, and true negative rate. The results have unequivocally demonstrated that Convolutional Neural Networks (CNN) stand out as the most proficient method for APT detection among the examined techniques. CNNs have exhibited superior performance, achieving high values across all the assessed metrics. This finding underscores the potential of CNNs as a powerful tool in fortifying cybersecurity systems against APT threats. While this study has yielded compelling results, it is essential to acknowledge the dynamic nature of the cybersecurity landscape. As APTs evolve and adapt, continuous research and innovation will be crucial to staying ahead of these threats. Future work could explore the integration of multiple detection methods or the incorporation of real-time data streams to enhance the accuracy and robustness of APT detection systems.

In conclusion, this research underscores the importance of machine learning, specifically the effectiveness of Convolutional Neural Networks, in mitigating the risks posed by APTs. By deepening our understanding of these detection techniques, we pave the way for more resilient and adaptive cybersecurity measures, providing a stronger defense against the ever-evolving world of cyber threats.

References

[1] A. Asharani, S. Myneni, A. Chowdhary and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851-1877, 2019.

[2] Chu, Wen-Lin, Chih-Jer Lin, and Ke-Neng Chang. "Detection and Classification of Advanced Persistent Threats and Attacks Using the Support Vector Machine," *Applied Sciences*, Vol. 9, no. 21, 2019.

[3] Yi-xi Xie, Li-xin Ji, Ling-shu Li, Zehua Guo, Thar Baker, "An adaptive defense mechanism to prevent advanced persistent threats," Connection Science, Vol. 33, No. 2, pp. 359-379, 2020.

[4] Zitong Li, Xiang Cheng, Lixiao Sun, Ji Zhang, Bing Chen, "A Hierarchical Approach for Advanced Persistent Threat Detection with Attention-Based Graph Neural Networks," Security and Communication Networks, vol. 2021, 1-14.

[5] Do Xuan, Cho, Dao, Mai Hoang, Nguyen and Hoa Dinh, "APT Attack Detection Based on Flow Network Analysis Techniques Using Deep Learning," Journal of Intelligent & Fuzzy Systems, vol. 39, no. 3, pp. 4785-4801, 2020.

[6] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," Computer Networks, vol. 109, pp. 127–141, 2016.

[7] Wang, Guozhu, Yiwen Cui, Jie Wang, Lihua Wu, and Guanyu Hu, "A Novel Method for Detecting Advanced Persistent Threat Attack Based on Belief Rule Base," *Applied Sciences*, Vol. 11, No. 21, 2021.

[8] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel, "Advanced persistent threats: Behind the scenes," in Information Science and Systems (CISS), 2016 ual Conference on. IEEE, 2016, pp. 181–186.

[9] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in IFIP International Conference on Communications and Multimedia Security. Springer, 2014, pp. 63–72.

[10] A. K. Sood and R. J. Enbody, "Targeted cyber-attacks: a superset of advanced Persistent threats,"

IEEE security & privacy, vol. 11, no. 1,pp. 54–61, 2013.

[11] P. Mell, K. Scarfone, and S. Romanosky, “Common vulnerability scoring system,” IEEE Security & Privacy, vol. 4, no. 6, 2006.

[12] M. Lee and D. Lewis, “Clustering disparate attacks: mapping the activities of the advanced persistent threat,” Last accessed June, vol. 26,2013.

[13] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar,and A. Rashid, “Data exfiltration: A review of external attack vectorsand countermeasures,” Journal of Network and Computer Applications,2018.

[14] X.Wang, K. Zheng, X. Niu, B.Wu, and C.Wu, “Detection of commandand control in advanced persistent threat based on independent access,”in Communications (ICC), 2016 IEEE International Conference on. IEEE, 2016, pp. 1–6.

[15] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, “Security evaluation ofthe cyber networks under advanced persistent threats,” IEEE Access,2017.

[16] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, “Panorama:capturing system-wide information flow for malware detection andanalysis,” in Proceedings of the 14th ACM conference on Computerand communications security. ACM, 2007, pp. 116–127.

[17] N. Virvilis and D. Gritzalis, “The big four-what we did wrong inadvanced persistent threat detection?” in Availability, Reliability andSecurity (ARES), 2013 Eighth International Conference on. IEEE,2013, pp. 248–254.

[18] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, “Malware detection using machine learning based analysis of virtual memory access patterns,” in2017 Design, Automation & Test in Europe Conference & Exhibition(DATE). IEEE, 2017, pp. 169–174.

[19] C. Vaas and J. Happa, “Detecting disguised processes using applicationbehaviorprofiling,” in Technologies for Homeland Security (HST), 2017 IEEE International Symposium on. IEEE, 2017, pp. 1–6.

[20] A. Bohara, U. Thakore, and W. H. Sanders, “Intrusion detection inenterprise systems by combining and clustering diverse monitor data,”in Proceedings of the Symposium and Bootcamp on the Science ofSecurity. ACM, 2016, pp. 7–16.

[21] A. Shalaginov, K. Franke, and X. Huang, “Malware beaconing detectionby mining large-scale dns logs for targeted attack identification,”in 18th International Conference on Computational Intelligence inSecurity Information Systems. WASET, 2016.

[22] A.M. Lajevardi, M. Amini, “Big knowledge-based semantic correlation for detecting slow and low-level advanced persistent threats,” Journal of Big Data Vol. 8, Issue 148, 2021.

[23] Longkang Shang, Dong Guo, Yuede Ji, Qiang Li, “Discovering unknown advanced persistent threat using shared features mined by neural networks,” Computer Networks, Volume 189, 2021,

[24] Hernandez Guillen, J.D. Martin del Rey, A., Casado-Vara, R, “Propagation of the Malware Used in APTs Based on Dynamic Bayesian Networks,” Mathematics, Vol. 9, 2021.

[25] Chaoxian Wei, Qiang Li, Dong Guo, Xiangyi Meng, “Toward Identifying APT Malware through API System Calls,” Hindawi Security and Communication Networks, Volume 2021, pp. 1-14.

[26] The GiuseppeLaurenza/I_F_Identifier dataset is taken from, https://github.com/GiuseppeLaurenza/I_F_Identifier, accessed on September 2022.

Authors Bibliography:

Anil Kumar received his M.Tech. in Computer Science and Engineering from the Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar (Haryana), India and pursuing Ph.D. in the same. Currently he is working as Assistant Professor of Computer Science at Indira Gandhi Govt. College Tohana (Fatehabad), Haryana, India. He has published more than 6 research papers in repute journals. His research interest in Cyber Security, Artificial Intelligence, Intrusion Detection and Machine Learning.

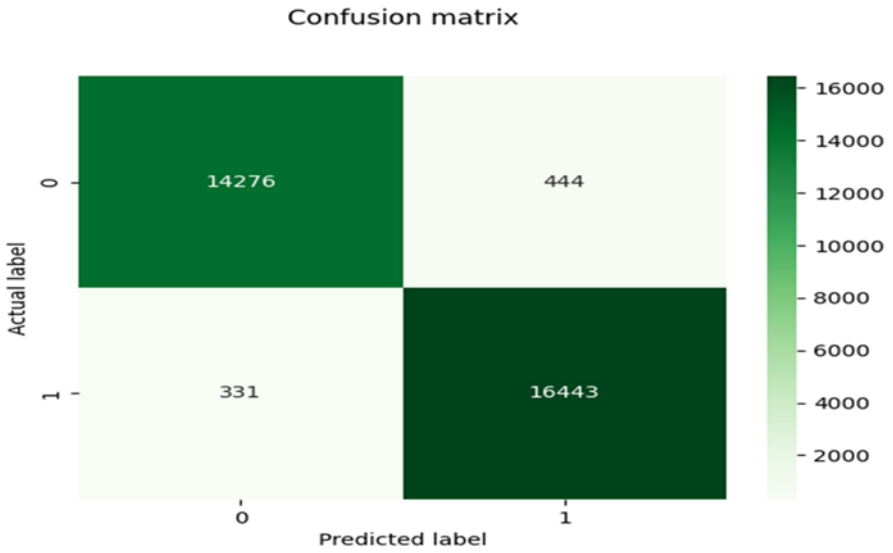
Amandeep Noliya received his B.Tech, M.Tech, and Ph. D Degrees in Computer Science and Engineering from the Department of Computer Science & Engineering, Guru Jambheshwar University of Science and Technology Hisar (Haryana), India. He is currently working as Assistant Professor in the Department of Artificial Intelligence and Data Science in the same Organization. He has published more than 25 papers in repute journals. His research interest is in Intrusion Detection, Cyber Security, Data Science, 5G, and Machine Learning.

Ritu Makani received his B.E. (Electronics), M.Tech (CSE), and Ph.D. (Network Security and Open Source Intrusion Detection System). Currently she is working as Associate Professor in the Department of Computer Science and Engineering at Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India. She has published more than 25 international journals and conferences. Her area of research interest is network security.

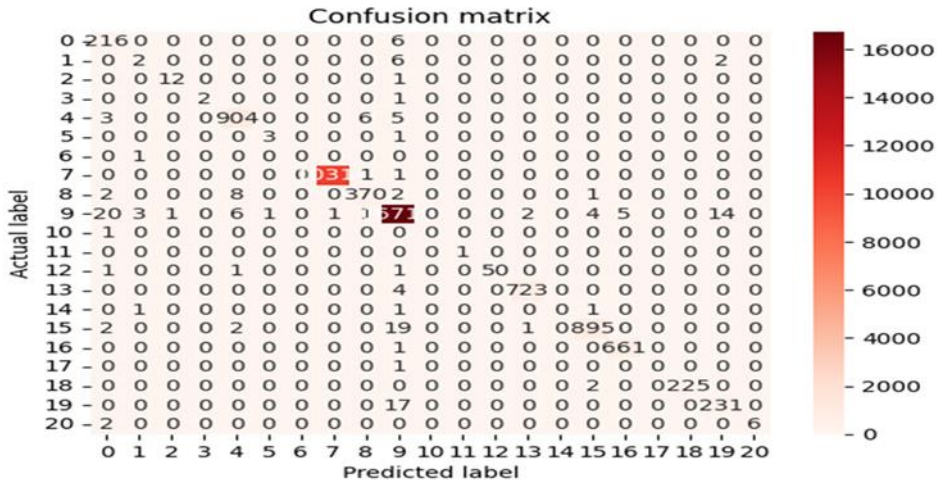
Pardeep Kumar received his MCA and Ph.D from Department of Computer Science and Engineering, Chaudhary Devi Lal University Sirsa, Haryana India. Currently he is working as Assistant Professor of Computer Science, Indira Gandhi Govt. College Tohana, Haryana, India. His research area is Cloud Computing, IoT and Machine Learning.

Jagsir Singh received his Ph.D. in Computer Engineering from the Department of Computer Science and Engineering, Punjabi University, Patiala, India in 2021. Currently he is working as Assistant Professor of Computer Science, Indira Gandhi Govt. College Tohana, Haryana, India. His research areas are Machine Learning, Computer and Cyber Security.

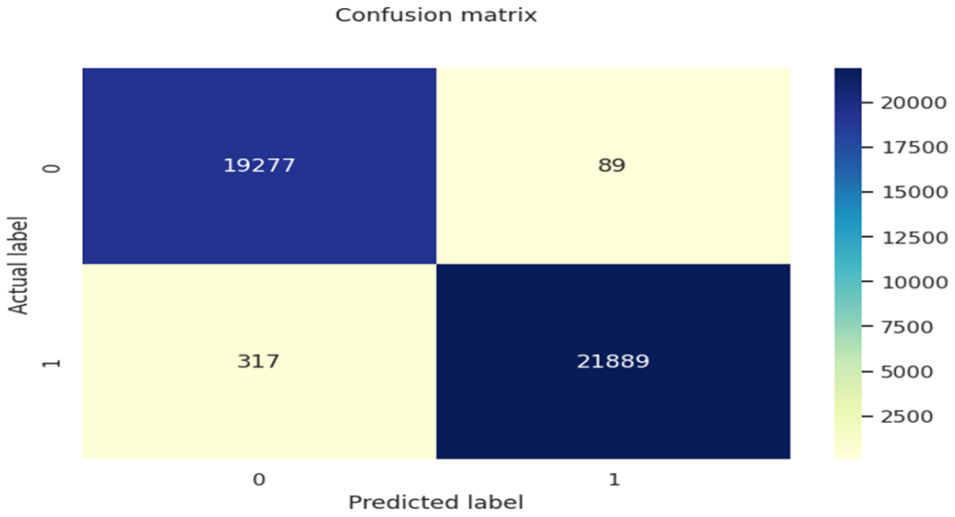
Appendix



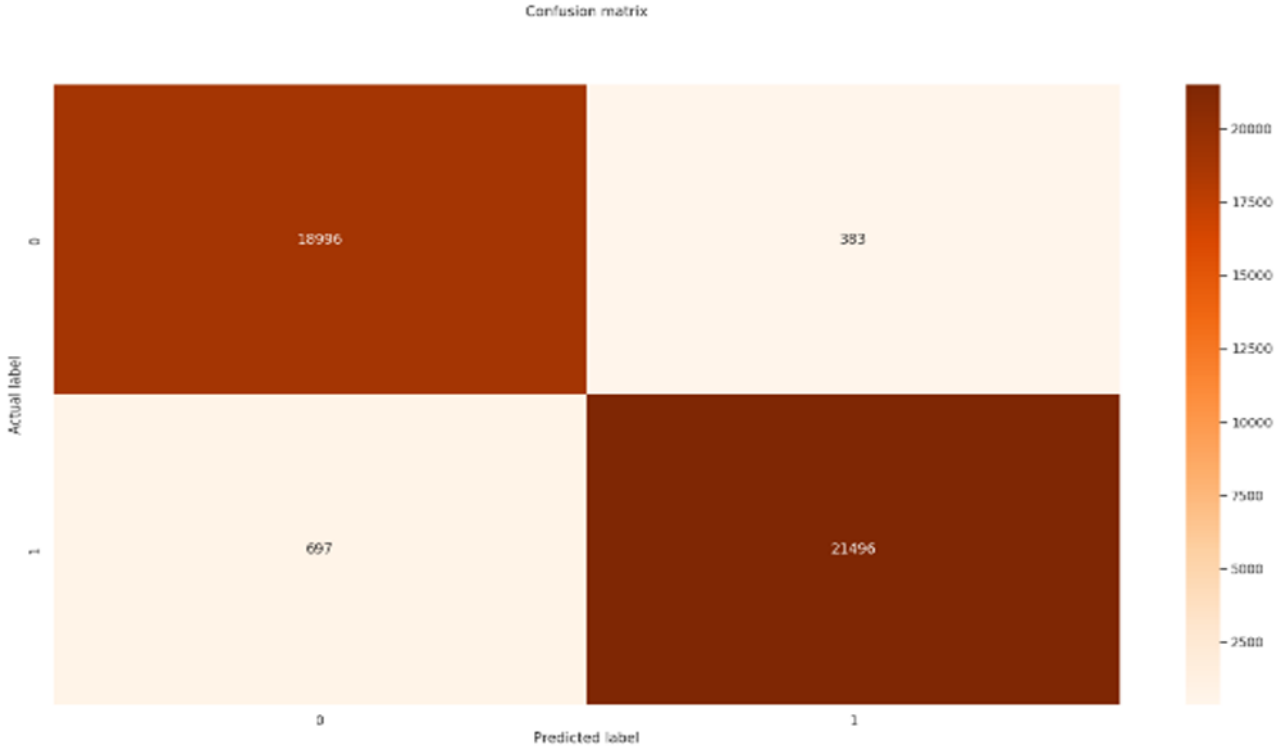
Confusion Matrix Support Vector Machine



Confusion Matrix K- Nearest Neighbours



Confusion Matrix Artificial Neural Networks



Confusion Matrix Decision Tree

Metric	Formula	Description
Precision	$\text{Precision} = \frac{TP}{TP+FP}$	*A small percentage of insider threat data entries are marked as malicious insiders, indicating a truly malevolent insider.*
Accuracy	$\text{Accuracy} = \frac{TN+TP}{TN+TP+FN+FP}$	*It represents the percentage of all positive and negative items that are correctly categorised. It can be regarded as a classifier's total effectiveness.*
F-score	$\text{F-score} = 2.\frac{\text{Precision.Recall}}{\text{Precision+Recall}}$	*It is also referred to as the F-measure and is the harmonic mean of sensitivity and precision.*
True Negative Rate (TNR)	$TNR = \frac{TN}{TN+FP}$	*TNR, often referred to as specificity, refers to the detector's capacity to accurately recognise data devoid of harmful elements.*
True Positive Rate (TPR)	$TPR = \frac{TP}{TP+FN}$	*TPR, often referred to as sensitivity or recall, is the percentage of insider threat information on dangerous entries that is accurately classified.*
False Positive Rate (FPR)	$FPR = \frac{FP}{FP+TN}$	*It is determined by dividing the total number of benign insiders by the proportion of benign insiders that are mistakenly labelled as harmful insiders.*
False Negative Rate (FPR)	$FNR = \frac{FN}{FP+TP}$	*False negative refers to the percentage of malicious insiders that were missed and misclassified as benign insiders by the model.*

Performance Evaluation Matrix