CHAPTER -INTRODUCTION

All of the country's daily needs can now be found on the internet, and the amount of time spent there is rapidly increasing. More people than ever are using the Internet. Cyberattacks and cybercrime are consequently growing more common. Diverse machine learning methodologies will be employed to identify network intrusions and counteract cyber security hazards. Creating intrusion detection systems can enhance cybersecurity and help find unusual activity on a server. Using machine learning techniques, an effective intrusion detection and prevention system will be developed. This project's main objective is to develop a probabilistic network intrusion detection system (IDS) that uses neural networks and probabilistic techniques to identify malicious network attacks.

**What is the problem?**

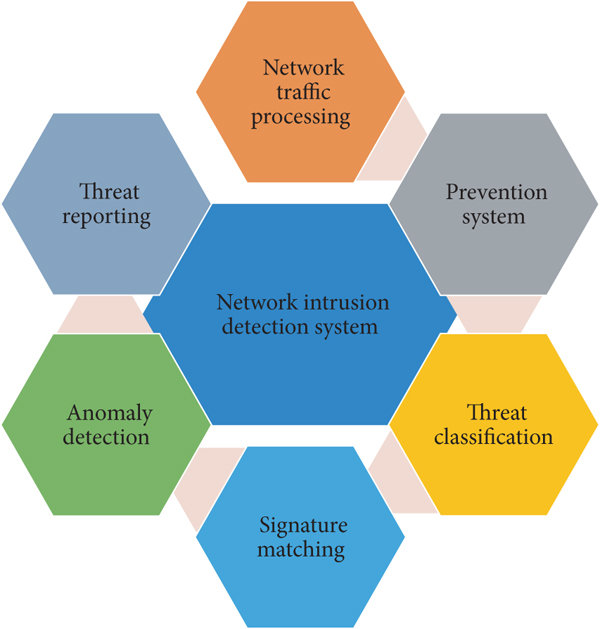
The field of network security is constantly changing, and traditional intrusion detection systems (IDS) are unable to keep up with the growing threats. Conventional systems frequently use signature-based detection, which compares incoming network traffic to known attack patterns. However, depending solely on signature-based techniques is no longer adequate due to the increasing cyber threats (such as advanced persistent threats and zero-day attacks).

The main problems with traditional approaches are:

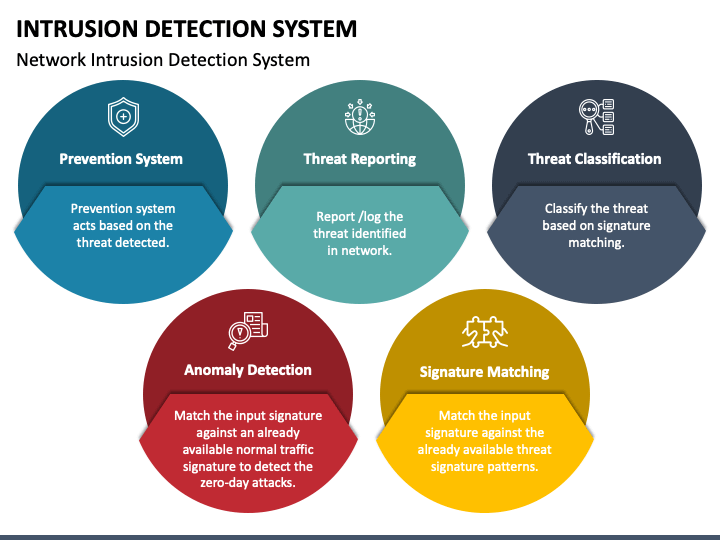
1. **A high rate of false positives**   
   Conventional intrusion detection systems frequently result in a notable quantity of false positives, which identify ordinary activities as suspicious. Security teams are overburdened by this, which makes it challenging to distinguish between genuine threats and benign activity.
2. **Restricted Identification of Unknown Dangers**  
   Traditional IDS often depend on signature-based detection, meaning they are successful against known attack patterns but struggle to identify novel or zero-day assaults that don’t fit pre-defined signatures.
3. **Failure to Recognize Insider Threats**  
   Conventional intrusion detection systems (IDS) tend to concentrate more on external threats and may be less successful in identifying malicious activity coming from inside the company, such as insider threats or improper use of privileges.
4. **Problems with Scalability**it difficult to scale as networks get bigger and more complicated. In large or distributed networks, they might not be able to handle the volume of traffic efficiently, which could result in missed detections
5. **Difficulty in Real-Time Detection**

Traditional IDS are not always effective in providing real-time detection, especially in high-speed networks. They may fail to detect an attack until after it has already occurred

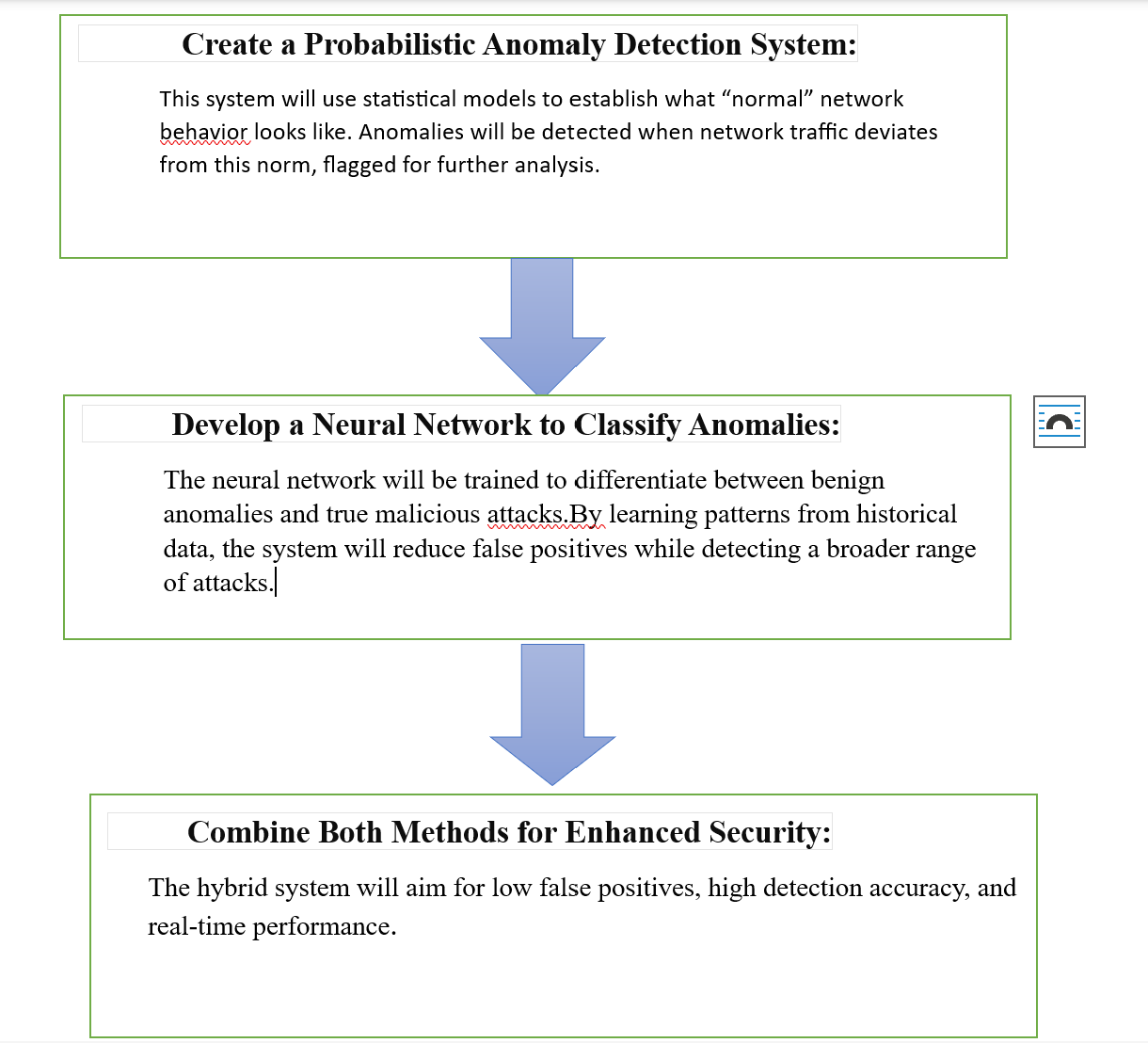
**What is the solution?**

****

Our goal is to develop a real-time system that can detect network intrusions using neural networks and probabilistic models. Over time, by examining patterns and data, the probabilistic model will come to understand what typical network behavior looks like. The system marks activity as anomalous when it finds deviations from this typical behavior. Anomalies, however, are not always dangerous; some might just be odd but unusual occurrences. We'll incorporate a neural network to assess these detected anomalies and identify whether they are actually malicious (like a cyberattack) or benign (like an odd but safe activity) in order to increase accuracy. The capacity of the neural network to recognize and understand complicated patterns will help lower the quantity of false alarms.

****

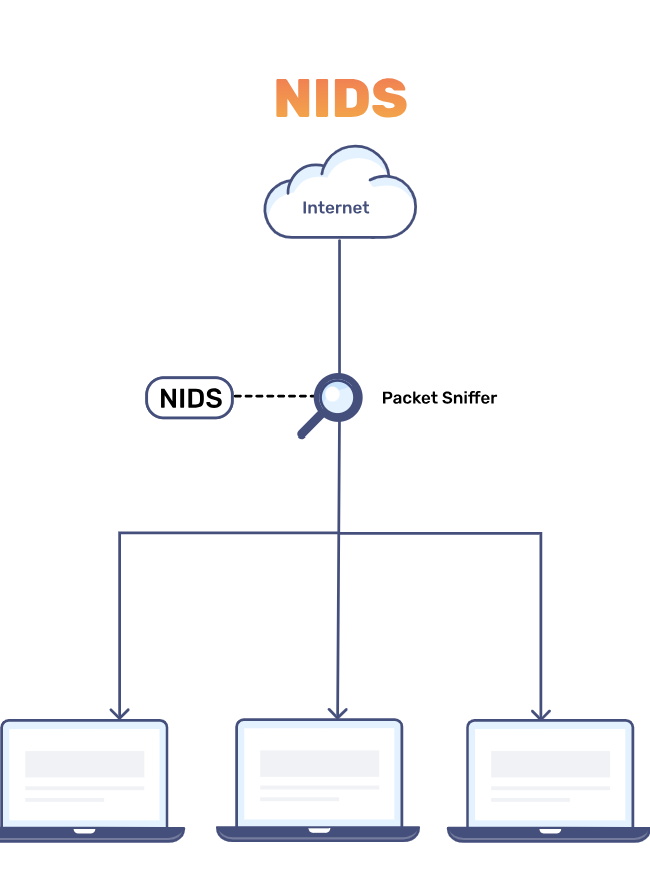
**Project Objectives:**



**Background Research:**

**network intrusion detection system**

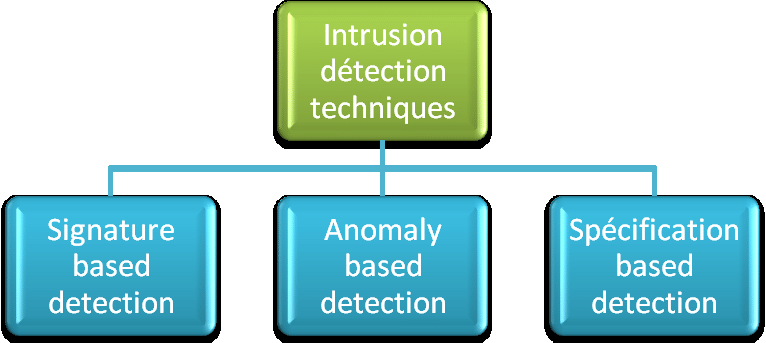
Network Intrusion Detection Systems monitor the incoming and outgoing network traffic at a strategic point in the network. They can analyze the traffic of the entire sub-network and report intrusions without using any resources of the monitored systems. They provide real-time detection and response because they generally analyze small amounts of data (packets or flows). However, they are unable to detect attacks that happen locally on the devices or on the parts of the network from which traffic does not pass the monitoring point and are unable to analyze the encrypted packet payloads**.**

****

**Key Concepts of NIDS:**

**Monitoring Network Traffic:**

* In order to identify unusual activity in real time, NIDS continually records and examines all network traffic.
* Typically, it involves strategically installing a sensor or monitoring device within a network, such as next to firewalls or gateways.

****

**Detection Techniques:**

* **Signature-based Detection:**
* Network traffic is compared by the system to a database of recognized attack signatures, or patterns.
* While fresh or unknown threats (zero-day attacks) cannot be detected by this technology, it is successful in detecting known attacks.
* **Anomaly-based Detection:**
* NIDS creates a baseline of typical network activity. Traffic is tagged as suspicious if it drastically departs from this baseline.
* Although this approach is more flexible and capable of identifying new threats, it could result in false positives.
* **Hybrid Detection:**
* Combines both signature-based and anomaly-based approaches to improve detection accuracy and reduce false positives.

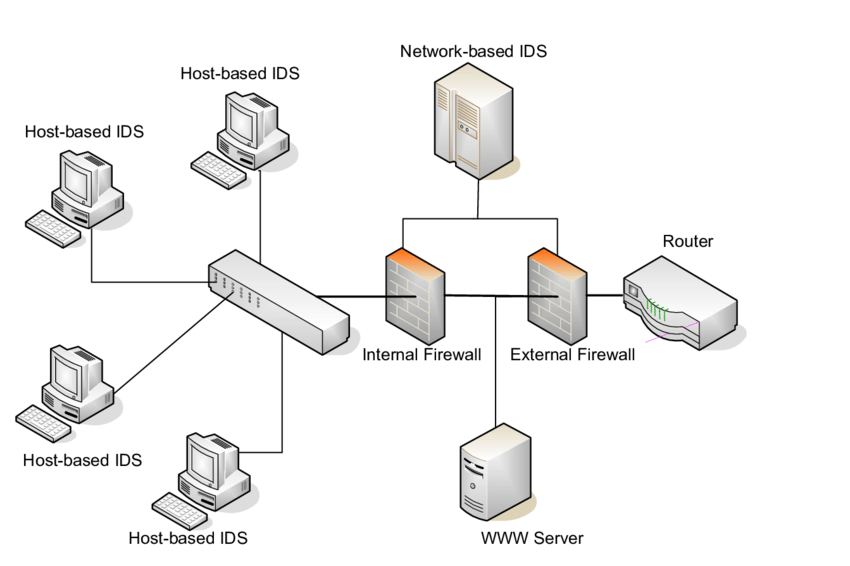
**Response Mechanisms:**

When NIDS detects an intrusion, it can take various actions:

* Alerting: Sending notifications to administrators about potential intrusions.
* Logging: Recording the suspicious traffic for further analysis.
* Active Response: Some systems can be configured to block the malicious traffic in real time or alert firewalls to deny access.

**Placement of NIDS:**

* **Network-Based NIDS (the most common type):**
  + Placed at critical points within a network to monitor all incoming and outgoing traffic.
* **Host-Based NIDS (HIDS):**
  + Monitors traffic to and from a specific host (e.g., an important server) and analyzes network activity on that machine.

****

**NIDS's advantages**  
1) NIDS notifies security personnel of questionable activity before an attack has a chance to do serious harm.  
 2) It facilitates prompt response by identifying active attacks.  
3) It offers insightful information about traffic trends and possible weak places in the network

**Probability and Anomaly Detection:**

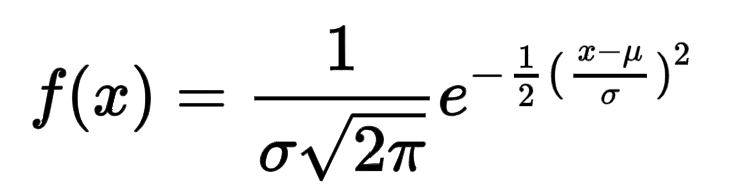
**How Probability is Used in IDS?**

In a Network Intrusion Detection System (NIDS), probability is used to model normal network behavior and identify anomalies based on statistical deviations. By analyzing historical network traffic data, probabilistic models like Gaussian distributions or Bayesian networks are employed to calculate the likelihood of certain events, such as specific traffic patterns, occurring under normal circumstances. These models estimate the probability of incoming traffic being part of the regular network behavior. If the probability of a certain activity is significantly lower than expected, it is flagged as an anomaly, suggesting a potential intrusion. This probabilistic approach allows the system to detect subtle deviations from the norm, making it effective in identifying previously unknown attacks. By continuously updating and adjusting these probabilities, the NIDS adapts to evolving network behaviors, offering a flexible and dynamic method of intrusion detection.

Probabilistic methods help in modeling normal network behavior by calculating the likelihood of certain network events. By assuming that network traffic follows a distribution (e.g., Gaussian/Normal Distribution), we can detect deviations from the norm.

* **Modeling Normal Behavior (Probabilistic Model)**

A probabilistic model, such as a Gaussian distribution, can be used to represent normal network traffic. Let x be a feature of network traffic (e.g., packet size, bandwidth usage). The probability density function (PDF) of the feature assuming a normal distribution is:



where:

µ is the mean (expected value) of the feature based on normal behavior.

σ is the standard deviation, representing the spread of the data.

If f(x) is significantly small, the feature value x is likely an anomaly, suggesting a possible intrusion.

* **Anomaly Detection (Thresholding)**

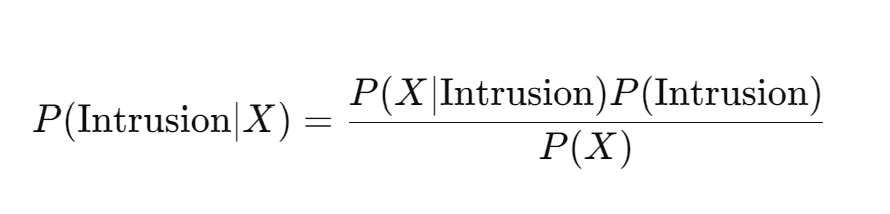
Once a model for normal behavior is established, we can flag events that deviate from this model as anomalies. A common method is setting a **threshold** T for anomaly detection. If the probability of an event x is lower than a defined threshold, it is considered an anomaly:

Anomaly if P(x) < T

The threshold T can be chosen based on the desired balance between **false positives** and **false negatives**.

* **Bayesian Networks**

In more advanced probabilistic approaches, **Bayesian networks** can be used to model the dependencies between different features of network traffic. Given a set of observations X={x1, x2,...,xn} the probability of a network event being normal or an intrusion is computed using **Bayes’ Theorem**:



where:

* P(X/Intrusion) is the likelihood of observing the features X given an intrusion.
* P(Intrusion) is the prior probability of an intrusion.
* P(X) is the marginal probability of observing X across all possible events.

If P(Intrusion/X) exceeds a certain threshold, the system flags the event as a potential attack.

Thus ,we will use probabilistic models to define what "normal" traffic is for a given network. When traffic deviates significantly from this model (according to a calculated probability threshold), it will be flagged as an anomaly.

The accuracy of this probabilistic approach depends on how well the model represents normal behavior and how much variability exists in legitimate traffic. If the model is too rigid, it may generate false positives (flagging normal behavior as a threat). If it’s too loose, it might miss genuine threats. Probabilistic systems are effective in detecting previously unknown or novel attacks (zero-day threats) but must be fine-tuned to balance detection rates and false positives for optimal accuracy.

Imagine we have:

* P(Intrusion) =0.05 (5% chance of an attack),
* P(X/Intrusion)= 0.9 (90% chance of anomaly if there is an attack),
* P(X) = 0.1 (10% chance of observing the anomaly overall).

Using Bayes' theorem:

P(Intrusion/X) = ( 0.05\*0.9)/0.1 = 0.45

This means there is a 45% probability that an observed anomaly indicates a real attack, which can inform security analysts about the need for further investigation.

**Neural Networks for Intrusion detection system**

Probabilistic models are excellent at finding anomalies, but they sometimes have trouble categorizing them. Not all anomalies are malicious occasionally, typical network activity can appear strange. Neural networks are used in this situation.

Neural networks are excellent at:

1. Pattern Recognition - Neural networks excel at recognizing complex patterns within large datasets .They can be trained to distinguish between benign and malicious traffic more finely than traditional rule-based systems could.
2. Adaptive Learning - Neural networks can adapt to changing network conditions.They can continuously update their understanding of what constitutes normal and abnormal behavior as they are trained on new data.
3. Reducing False Positives -By learning from historical data, neural networks can improve classification accuracy. This leads to a reduction in false positives
4. High Dimensionality - Neural networks are well-suited for handling high-dimensional data, allowing them to process and learn from multiple features simultaneously.

1. Integration with Other Techniques -Neural networks can be effectively combined with probabilistic models or other machine learning techniques to create hybrid systems. Thus integration enhances overall detection capabilities

1. Scalability - Neural networks can handle large volumes of data, making them suitable for modern

networks that generate significant amounts of traffic.

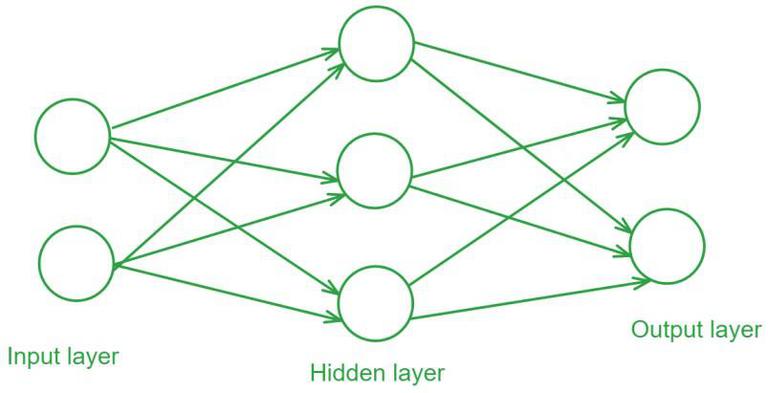
**Types of Neural Networks:**

1) **Feedforward Neural Network**

A Feedforward Neural Network (FNN) is a type of artificial neural network where connections between the nodes do not form cycles. This characteristic differentiates it from recurrent neural networks (RNNs). The network consists of an input layer, one or more hidden layers, and an output layer. Information flows in one direction—from input to output—hence the name “feedforward.”

**Structure of a Feedforward Neural Network**

1. Input Layer: The input layer consists of neurons that receive the input data. Each neuron in the input layer represents a feature of the input data.
2. Hidden Layers: One or more hidden layers are placed between the input and output layers. These layers are responsible for learning the complex patterns in the data. Each neuron in a hidden layer applies a weighted sum of inputs followed by a non-linear activation function.
3. Output Layer: The output layer provides the final output of the network. The number of neurons in this layer corresponds to the number of classes in a classification problem or the number of outputs in a regression problem**.**



2) **Recurrent Neural Network (RNN)**

Recurrent Neural Network(RNN) is a type of neural network where the output from the previous step is fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other. Still, in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is its **Hidden state**, which remembers some information about Sequence. The state is also referred to as *Memory State*since it remembers the previous input to the network.

