Chapter 2

LITERATURE SURVEY

The existing literature regarding intrusion detection systems as well as machine learning algorithms are discussed below. Additionally the theoretical details of Hidden Markov Models and Naive Bayes Classifier are explained.

2.1 Intrusion Detection Systems

An Intrusion can be termed as any activity which is aimed at disrupting or denying a service by gaining unauthorized access. Various attacks like DDoS, Identity spoofing, Ping of Death, Buffer Overflow etc. are used to intrude and disrupt a network. Intrusion Detection Systems are used to prevent such intrusions by detecting them at an earlier stage.

All Intrusion Detection Systems use one of two detection techniques:

i. Signature Based Detection
ii. Statistical Anomaly Based Detection

2.1.1 Signature Based Detection

Signature-based IDS monitors packets in the network, and compares them with pre-configured and pre-determined attack patterns, known as signatures.[6]

2.1.2 Statistical Anomaly Based Detection

Anomaly based detection techniques model a “normal” scenario and any deviation from that is considered an anomaly. Anomaly based detection is preferred over Signature based detection due to their ability to detect novelty attacks.[6]

2.2 Machine Learning

Machine learning is a branch of artificial intelligence, concerned with the construction and study of systems that can learn from data. Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.[7]

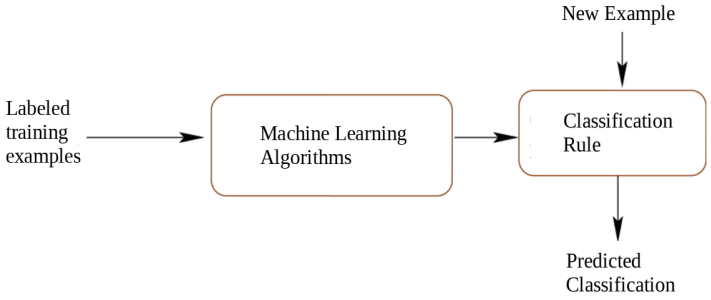

Figure 2.1 Machine Learning System

Machine learning algorithms are classified into 3 types-

i. Supervised Learning
ii. Unsupervised Learning
iii. Reinforcement Learning

2.2.1 Supervised Learning

In supervised learning, the model defines the effect one set of observations, called inputs, has on another set of observations, called outputs.[7] As shown in Figure 2.2, the inputs are assumed to be at the beginning and outputs at the end of the causal chain. The models can include mediating variables between the inputs and outputs. A supervised learning algorithm makes it possible, given a large data set of input/output pairs, to predict the output value for some new input value that you may not have seen before. In supervised learning, one is trying to find the connection between two sets of observations.

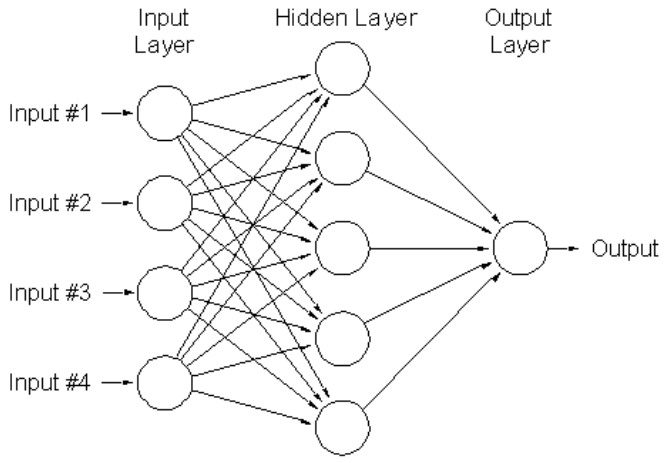

Figure 2.2 Supervised Learning Model

2.2.2 Unsupervised Learning

In unsupervised learning, all the observations are assumed to be caused by latent variables, that is, the observations are assumed to be at the end of the causal chain.[7] With unsupervised learning it is possible to learn larger and more complex models than with supervised learning. The learning can proceed hierarchically from the observations into ever more abstract levels of representation. Each additional hierarchy needs to learn only one step and therefore the learning time increases (approximately) linearly in the number of levels in the model hierarchy unlike supervised learning where the learning task increases exponentially. Unsupervised learning can be used for bridging the causal gap between input and output observations. The latent variables in the higher levels of abstraction are the causes for both sets of observations and mediate the dependence between inputs and outputs.

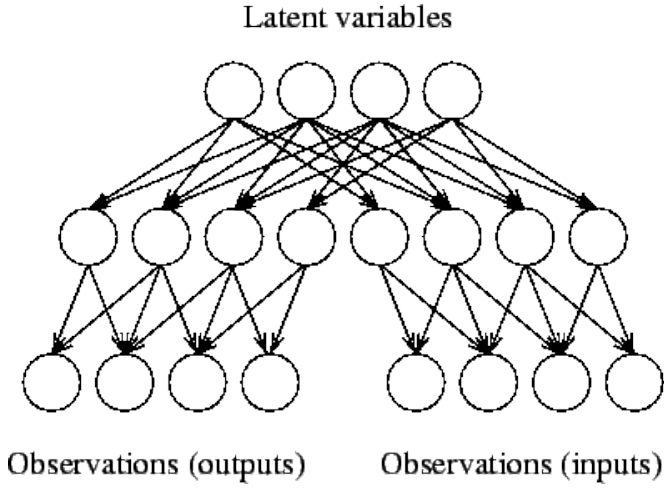

Figure 2.3 Unsupervised Learning Model

2.2.3 Reinforcement Learning

Reinforcement learning is learning what to do - how to map situations to actions - as to maximize a numerical reward signal. A Reinforcement learning agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions on basis of its experiences (exploitation) and also by new choices (exploration), which is essentially trial and error learning.

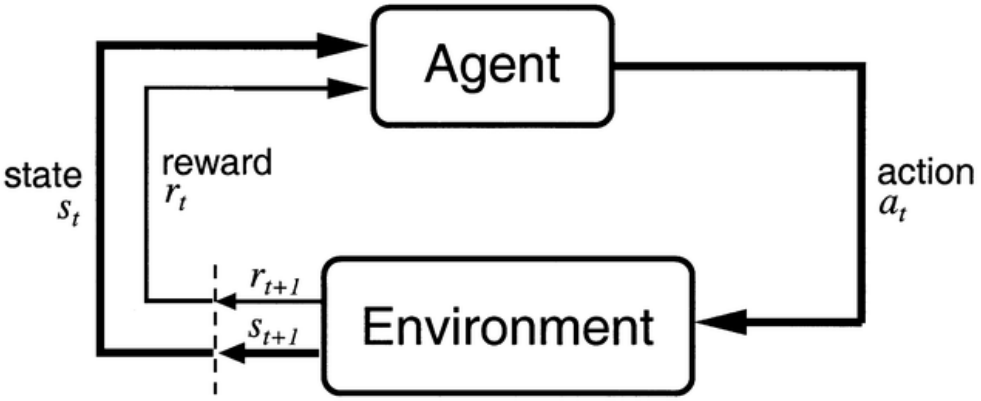

Figure 2.4 Reinforcement Learning Model

2.3 Types of Anomaly Based Detection

We now review the different techniques of Anomaly based IDS. The most important ones are-

i. Statistical Anomaly Detection
ii. Data Mining Based Detection
iii. Knowledge Based Detection

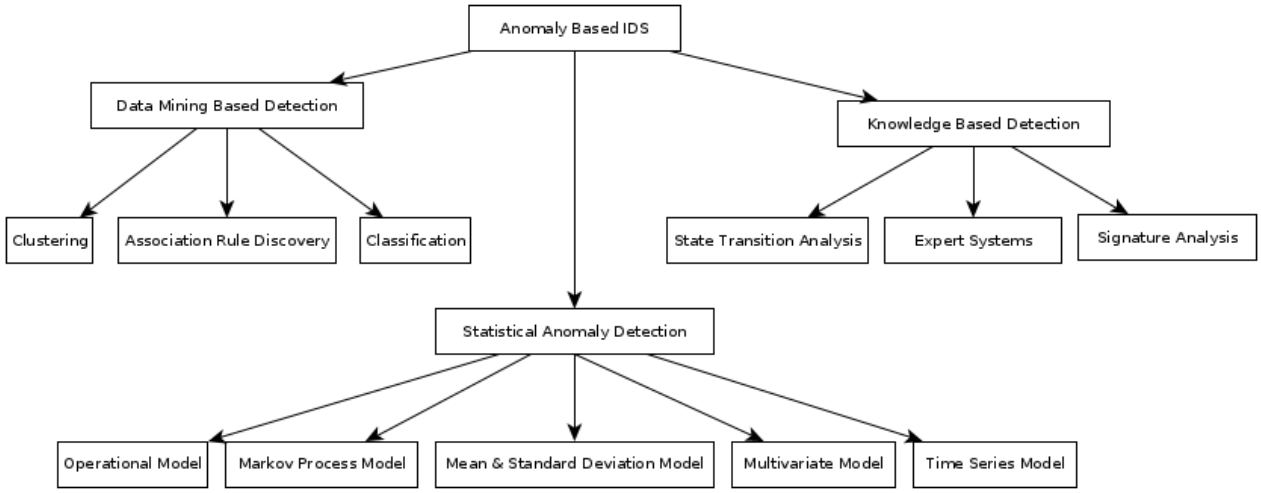

Figure 2.5 Taxonomy of Anomaly Based IDS

2.3.1 Statistical Anomaly Detection

Statistical based anomaly detection techniques use statistical properties and statistical tests to determine whether "Observed behavior" deviate significantly from the "expected

behavior". Statistical based anomaly detection techniques use statistical properties (e.g., mean and variance) of normal activities to build a statistical based normal profile and employ statistical tests to determine whether observed activities deviate significantly from the normal profile. The IDS goes on assigning a score to an anomalous activity. As soon as this score becomes greater certain threshold, it will generate an alarm.[8] Statistical Anomaly Based IDS can further be classified into the following categories-

- **Operational Model:** This model is based on the operational assumption that an anomaly can be identified through a comparison of an observation with a predefined limit. Based on the cardinality of observation that happens over a time an alarm is raised. The operational model is most applicable to metrics where experience has shown that certain values are frequently linked with intrusions. For example, an event counter for the number of password failures during a brief period, where more than say 10, suggest a failed log-in.
- **Markov Process Model:** The Markovian Model is used with the event counter metric to determine the normalcy of a particular event, based on the events which preceded it.[8] The model characterizes each observation as a specific state and utilizes a state transition matrix to determine if the probability of the event is high (normal) based on the preceding events. This model is basically used in two main approaches: Markov chains and Hidden Markov models. A Markov chain keeps track of an intrusion by examining the system at fixed intervals and maintains the record of its state. If the state change takes place it computes the probability for that state at a given time interval. If this probability is low at that time interval then that event is considered as an anomalous.
- **Mean and Standard Deviation Model:** The Mean and Standard Deviation Model is based on the traditional statistical determination of the normalcy of an observation based on its position relative to a specified confidence range.[8] In this sub-model, event that falls outside the set interval will be declared as an anomalous.
- **Multivariate Model:** This model can be applied to intrusion detection for monitoring and detecting anomalies of a process in an information system.[8] This model is

similar to the mean and standard deviation model except that it is based on correlations among two or more features. This model would be useful in the situation where two or more features are related. This model permits the identification of potential anomalies where the complexity of the situation requires the comparison of multiple parameters.

- **Time Series Model:** The Time Series Model, attempts to identify anomalies by reviewing the order and time interval of activities on the network.[8] If the probability of the occurrence of an observation is low, then the event is labeled as abnormal. This model provides the ability to evolve over time based on the activities of the users. Anomalies in time series data are data points that significantly deviate from the normal pattern of the data sequence.

2.3.2 Data Mining Based Approach

As IDS can only detect known attacks, but it cannot detect insider attacks, the better solution for an IDS can be Data Mining at its core and is defined as "the process of extracting useful and previously unnoticed models or patterns from large data stores".[8] The data-mining process tends to reduce the amount of data that must be retained for historical comparisons of network activity, creating data that is more meaningful to anomaly detection.

Data Mining based approach is classified into following techniques:

- **Clustering:** Clustering[8] is an unsupervised technique for finding patterns in unlabeled data with many dimensions (number of attributes). Mostly k-means clustering is used to find natural groupings of similar instances. Records that are far from any of these clusters indicate unusual activity that may be part of a new attack.
- **Association Rule Discovery:** Association rules mining finds correlation between the attributes. This technique was initially applied to the so-called market basket analysis, which aims at finding regularities in shopping behavior of customers of supermarkets. This method is usually very slow and its being replaced by other powerful techniques like clustering and classification.[8]

- **Classification:** Classification[8] is one of the main techniques used in data mining. Its main goal is to learn from class-labeled training instances for predicting classes of new or previously unseen data. Instances in the training set have labels. New data is classified based on the training set. Basically a classification tree (also called as decision tree) is built to predict the category to which a particular instance belongs.

2.3.3 Knowledge Based Detection Techniques

Knowledge based detection Technique[8] can be used for both signature based IDS as well as anomaly based IDS. It accumulates the knowledge about specific attacks and system vulnerabilities. It uses this knowledge to exploit the attacks and vulnerabilities to generate the alarm. Any other event that is not recognized as an attack is accepted. Therefore the accuracy of knowledge based intrusion detection systems is considered good. Knowledge based detection Technique can further be classified as:

i. **State Transition Analysis:** This technique describes the attacks with a set of goals and transitions, and represents them as state transition diagrams.[8] State transition diagram is a graphical representation of the actions performed by an intruder to archive a system compromise. In state transition analysis, an intrusion is viewed as a sequence of actions performed by an intruder that leads from some initial state on a computer system to a target compromised state. State transition analysis diagrams identify the requirements and the compromise of the penetration. They also list the key actions that have to occur for the successful completion of an intrusion.

ii. **Expert Systems:** The expert system[8] contains a set of rules that describe attacks. Audit events are then translated into facts carrying their semantic significance in the expert system, and the inference engine draws conclusions using these rules and facts. This method increases the abstraction level of the audit data by attaching semantic to it. It also encodes knowledge about past intrusions, known system vulnerabilities and the security policy. As information is gathered, the expert system determines whether any rules have been satisfied.

iii. **Signature Analysis:** Signature analysis follows exactly the same knowledge-acquisition approach as expert systems, but the knowledge acquired is exploited in a different way.[8] The semantic description of the attacks is transformed into information that can be found in the audit trail in a straightforward way. For example, attack scenarios might be translated into the sequences of audit events they generate, or into patterns of data that can be sought in the audit trail generated by the system.

2.4 Statistical Modeling

The formalization of relationships between variables in the form of mathematical equations results in a **Statistical Model**. It describes how random variables are related to each other. The model is statistical as the variables are not deterministically but rather **stochastically** related.

2.4.1 Markov Models

In order to understand the definition of a Markov Model, a clear definition of the mathematical principle known as Markov Property must be established. A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, rather than on the sequence of events that preceded it.

In other words, a Markov Model is used mainly to analyze temporal data, and it assumes that the future is independent of the past states, and is dependent only on the data contained in the present state. Given the following set $D = \{X_1 ... X_n\}$, we can say X_t depends on $X_{t-1} ... X_{t-m}$ with the simplest case being m=1. We further simplify our model by assuming discrete time and space of the sets.[9]

The most common Markov models and their relationships are summarized in the following table:

	Fully Observable System States	**Partially Observable System States**
Autonomous System	Markov Chain	Hidden Markov Model
Controlled System	Markov Decision Process	Partially Observable MDP

Table 2.1 Markov Models and their relationships

2.4.2 Markov Chains

Markov chains are used to model autonomous systems whose states are fully observable. Discrete random variables $X_1 ... X_n$ form a Markov Chain, if they respect

$$P(X_t|X_1 ... X_{t-1}) = P(X_t|X_{t-1})$$

The above equation is illustrated as:

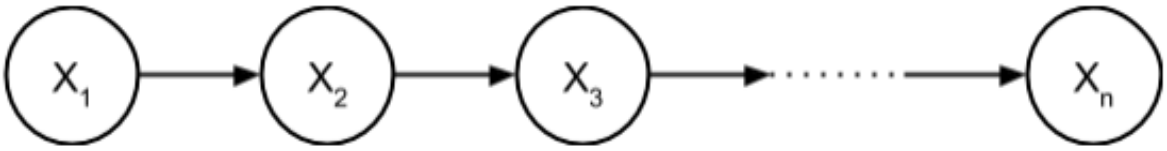

Figure 2.6 Markov Chain Illustration

The biggest drawback of Markov Chains is that it is assumed that the whole system is visible. This assumption is invalid in several real-life scenarios as more often there are two different types of states/variable namely observable variables and hidden variable. Hidden variables are also known as latent variables.

2.4.3 Hidden Markov Model

To overcome the disadvantage of Markov Chains, the concept of Hidden Markov Models (HMM) is used. Hidden Markov Models are useful to define an autonomous stochastic model whose states are partially observable. The concept of HMM is as illustrated in the **Trellis Graph** shown below:

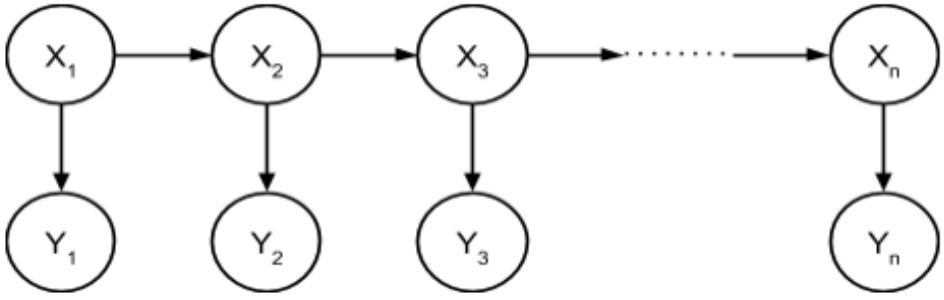

Figure 2.7 Trellis Graph Representation of a HMM

In the above graph, there are two types of states. The hidden/latent variables are represented as,

$$X_1 ... X_n \in \{1...m\}$$

The visible states, which are observable, are represented by

$$Y_1 ... Y_n \in \chi \text{(discrete/finite variables)}$$

There are three parameters under consideration under Hidden Markov Models.

i. State Transition Probability Matrix. The probability of the HMM transitioning from an existing state X_k to a new state X_{k+1}. It is formally defined as
$A_{ij} = P(X_{k+1} = j | X_k = i) \forall (i, j\{1...m\})$

ii. Emission Probability. The probability of a given observable variable Y_k occurring given the probability of a latent variable X_k. It is a **probability mass function**, as we are considering only discrete random variables. It is formally defined as
$B_i(y) = P(Y_k = y | X_k = i) \forall (i \in \{1...m\}, x \in \chi)$

iii. Initial Distribution Probability. The probability that a given HMM is in an initial state X_k. It is formally defined as
$\pi_i = P(X_k = i) \forall (i \in \{1...m\})$

Given the above three parameters, it is possible to formally define HMM's as follows: A Hidden Markov Model (λ) is a five tuple as given: $(\lambda) = [X, Y, A, B, \pi]$, with the joint probability distribution of X and Y defined as,

$$P(Y_1 ... Y_n, X_1 ... X_n) = P(X_1) P(Y_1 | X_1) \prod_{k=2}^{n} P(X_k | X_{k-1}) P(Y_k | X_k)$$

There are three problems associated with Hidden Markov Models. These are commonly used to analyse and obtain useful information from the HMM. The three problems are as described.

i. **Evaluation** of the probability that a given model generate a given sequence of observations. This is solved by the **Forward-Backward Algorithm**.
ii. **Decoding** the sequence of the hidden states that most likely generated a given sequence of observations. The **Viterbi Algorithm** solves this problem.
iii. **Learning** the model which most probably underlies the given sample of observations. In other words, learning the parameters of the HMM. This problem is solved using the **Baum-Welch Algorithm**.

2.4.4 HMM Based Related Work

i. **Adaptive Network Intrusion Detection System Using a Hybrid Approach:** In this approach, an adaptive network intrusion detection system, that uses a two stage architecture is described. In the first stage a probabilistic classifier is used to detect potential anomalies in the traffic. In the second stage a HMM based traffic model is used to narrow down the potential attack IP addresses. Due to the difficulties that arise when implementing HMM in real time, HMM model along with NB model was incorporated into a hybrid model for intrusion detection[10]. The proposed hybrid model performed well in detecting intrusions. HMM with more than five states had 100% accuracy classifying all IPs from CAIDA[11] as attack and IPs from DARPA[12] as clean. When tried with HMM with less than five states, few of the attacking streams were classified as clean traffic. This is due to the inability of HMM with very few states to model the data accurately. Using HMM with larger states gave exact results and the number of states to be chosen for a server can be computed empirically.
ii. **Sensing attacks in Computers Networks with Hidden Markov Models:** In this approach,[13] an Intrusion Detection model for computer networks based on Hidden Markov Models is proposed. The proposed model aims at detecting intrusions by analysing the sequences of commands that flow between hosts in a network for a particular service (e.g., an ftp session). For each command sequence, a probability value is assigned and a decision is taken according to some predefined decision threshold. First the system must be trained in order to learn the typical sequences of

commands related to innocuous connections. Then, intrusion detection is performed by identifying anomalous sequences. Different HMM models were investigated in terms of the dictionary of symbols, number of hidden states, and different training sets. It was found that good performances can be attained by using dictionary of symbols made up of all symbols in the training set, and adding a NaS (not-a-symbol) symbol in account of symbols in the test set that are not represented in the training set. Performances can be further improved by combining different HMM. As the size of training sets in an intrusion detection application is typically large, it was proposed to split the training set in a number of parts, training different HMM and then combining the output probabilities by three well known combination.

2.5 Naive Bayes Classifier

A Naïve Bayes Classifier is a simple probabilistic classifier which uses the Bayes' theorem with an assumption that the occurrence of an event is totally unrelated to the occurrence of another.

Despite their naive design and apparently oversimplified assumptions, Naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers.[14]

2.5.1 Bayes Theorem

Bayes theorem is used by the Naïve Bayes Classifier. It is particularly suited when the dimensionality of the inputs is high. The formula for the Bayes theorem is represented by the figure below.

$$P(W|L) = \frac{P(L|W)P(W)}{P(L)} = \frac{P(L|W)P(W)}{P(L|W)P(W) + P(L|M)P(M)}$$

where,

$P(W|L)$ →Probability of outcome W given L or "posterior probability"

$P(L|W)$ →Probability of outcome L given W or "likelihood"

$P(W)$ →Independent probability of W

$P(L)$ →Independent probability of L or "prior probability"

To explain with an example, let W represent the event that the conversation was held with a Woman and L denote the event that the conversation was held with a long-haired person. Assuming that women constitute half the population, the probability that W occurs is P (W) = 0.5.

Suppose it is also known that 75% of women have long hair, which we denote as $P(L|W) = 0.75$. Likewise, suppose it is known that 15% of men have long hair, or $P(L|M) = 0.15$, where M is the complementary event of W.

Calculation of the probability that the conversation was held with a woman, given the fact that the person had long hair, or, in our notation, $P(W|L)$ can be calculated using Bayes formula shown in the figure above.

2.5.2 Formal Definition and Background

The naive Bayes classifier applies to learning tasks where each instance x is described by a conjunction of attribute values and where the target function $f(x)$ can take on any value from some finite set V. A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values $< a_1, a_2...a_n >$. The learner is asked to predict the target value, or classification, for this new instance.[15]

Assumption:

- Training set consists of instances described as conjunctions of attribute values
- Target classification is based on a finite set of classes 'V'
- Independence assumption: features are independent in a given class.

Task: Predict the correct class for a new instance $< a_1, a_2...a_n >$

Key Idea: The Bayesian approach to classifying the new instance is to assign the most probable target value, VMAP, given the attribute values $< a_1, a_2...a_n >$ that describe the instance.

$$V_{MAP} = argmax\ P(v_j|a_1, a_2, a_3...a_n)\ \forall\ v_j \in V$$

Rewriting the above equation using Bayes Theorem we get,

$$V_{MAP} = argmax\ \frac{P(a_1, a_2, a_3...a_n|v_j)}{P(a_1, a_2, a_3...a_n)}\ \forall\ v_j \in V$$

$$V_{MAP} = argmax\ P(a_1, a_2, a_3...a_n|v_j)P(v_j)\ \forall\ v_j \in V$$

The naive Bayes classifier is based on the simplifying assumption that the attribute values are conditionally independent given the target value. The probability of observing the conjunction <a1, a2… an> is just the product of the probabilities for the individual attributes. That is,

$$V_{MAP} = argmax\ P(a_1, a_2, a_3...a_n|v_j) = \prod_{i}^{n} P(a_i|v_j)$$

Hence we get the following classifier:

$$v_{NB} = argmax\ P(v_j) \sum_{i}^{n} P(a_i|v_j)$$

where,

$v_{NB} \rightarrow$ target value output by the NBC.

2.5.3 Properties of Naive Bayes

The properties of Naive Bayes Classifier are:

- **Incremental approach:** With each training example, the prior and the likelihood can be updated dynamically. This results in a flexible and a robust model.
- **Combines prior knowledge and observed data:** The prior probability of a hypothesis multiplied with a probability of the hypothesis given the training data leads to more accuracy in correct classification.
- **Probabilistic hypothesis:** The output obtained is not just the classification but also a probability distribution over all classes.
- **Meta-Classification:** The outputs of several classifiers can be combined by multiplying the probabilities that all classifiers predict for a given class.

2.5.4 Advantages and Limitations

In this section we discuss the various advantages and disadvantages of the Naïve Bayes classifier.

Advantages are:

- Naïve Bayes Classification is computationally fast and space efficient
- It is faster to train the classifier as it requires a single scan
- It is not sensitive to irrelevant data
- It handles streaming data well

Disadvantages are:

- Requires several records to obtain good results
- When a predictor category is not present in the training data, naive Bayes assumes that a record with that category of the predictor has zero probability. This can be a problem if this rare predictor value is important.

2.5.5 Recent Research in Network Anomaly Detection in NBC

- **A System Approach to Network Modelling for DDoS Detection using a Naive Bayesian Classifier by R Vijayasarathy, S V Raghavan and Balaraman Ravindran:** In this paper, the authors discuss a practically realized system to detect DoS attacks using a Naive Bayesian Classifier. They have modeled the network for two protocols – TCP and UDP. The concept of windowing is used to have a reasonable estimate and control over the reaction time of the system for attacks and better modeling from larger datasets.[16] Three datasets were used in the experiment, namely, DARPA dataset, SETS (TCP) dataset and SETS (UDP) dataset. To determine four parameters i.e. false positives, false negatives, true positives, true negatives, they performed two tests: False Positive Test (FPT) and False Negative Test (FNT). They have calculated critical system parameters namely, Accuracy, False Alarm Rate and Miss Rate. The results of these tests are represented in figure 2.8 They report improvements in accuracy with increase in window size.

WS	DARPA at t =1%			SETS at t =5%		
	Acc	FAR	MR	Acc	FAR	MR
50	98.3%	2.3%	0	97%	7%	0
100	98.6%	2%	0	97.4%	6.2%	0.06%
200	98.7%	1.8%	0	97.1%	5%	1.1%

Figure 2.8 Experimental Results for TCP

- **From Feature Selection to Building of Bayesian Classifiers: A Network Intrusion Detection Perspective by Kok-Chin Khor, Choo-Yee Ting and Somnuk-Phon Amnuaisuk:** In this paper, the authors proposed a novel approach to select important features by utilizing two selected feature selection algorithms utilizing filter approach. Extra features were added into the final proposed feature set. They then constructed three types of BN namely, Naive Bayes Classifiers (NBC), Learned BN and Expert-elicited BN by utilizing a standard network intrusion dataset. The performance of each classifier was recorded. They found that there was no difference in overall performance of the BNs and therefore, concluded that the BNs performed equivalently well in network attacks. They conclude that Nave Bayes Classifier should be considered if the dataset involved is huge and time for building network and testing is the main factor to consider. Experimental results are represented in the figure 2.9. [17]

Category	NBC	BN	EE	Wenke lee
Normal	96.7	99.8	97.5	N/A
DoS	99.3	99.9	99.2	79.9
Probe	93.5	89.4	93.0	97
R2L	92.8	91.5	86.8	75
U2R	55.8	69.2	69.2	60

Figure 2.9 Comparison of Classification Results of various BN's on full dataset and Wenke Lee

- **Combining Naïve Bayes and decision tree for adaptive Intrusion Detection by Dewan Md. Farid, Nouria Harbi, and Mohammad Zahidur Rahman:** In this paper, the authors propose a new learning algorithm for adaptive network intrusion detection using naive Bayesian classifier and decision tree, which performs balance detection and keeps false positives at acceptable level for different types of network attacks, and eliminates redundant attributes as well as contradictory examples from

training data that make the detection model complex. The proposed algorithm also addresses some difficulties of data mining such as handling continuous attribute, dealing with missing attribute values, and reducing noise in training data. The authors tested the performance of their proposed algorithm with existing learning algorithms by employing on the KDD99 benchmark intrusion detection dataset. Their experimental results prove that the proposed algorithm achieved high detection rates (DR) and significant reduce false positives (FP) for different types of network intrusions using limited computational resources. The results of the experiment can be found below. [18]

Method	Normal	Probe	DOS	U2R	R2L
Proposed Algorithm (DR %)	99.72	99.25	99.75	99.20	99.26
Proposed Algorithm (FP %)	0.06	0.39	0.04	0.11	6.81
Naïve Bayesian (DR %)	99.27	99.11	99.69	64.00	99.11
Naïve Bayesian (FP %)	0.08	0.45	0.04	0.14	8.02
ID3 (DR %)	99.63	97.85	99.51	49.21	92.75
ID3 (FP %)	0.10	0.55	0.04	0.14	10.03

Figure 2.10 Comparison of results using 41 attributes

2.6 Proposed System

Researchers are trying to build a fool-proof system which can be used to detect anomalies. The current systems haven't achieved a satisfactory detection rate. Constantly working to counter challenges encountered in the field of network anomaly detection is imperative. Through our proposed system, we plan to increase the existing accuracy and to contribute to the field of Network Security. The proposed system also introduces new challenges in the field of anomaly detection.