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 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.

2.1.1 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 2.1:

	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

2.1.2 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 in figure 2.1 as:

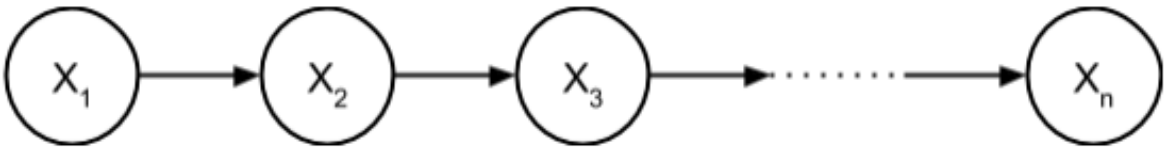

Figure 2.1 Markov Chain Illustration

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.

2.1.3 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 in figure 2.2:

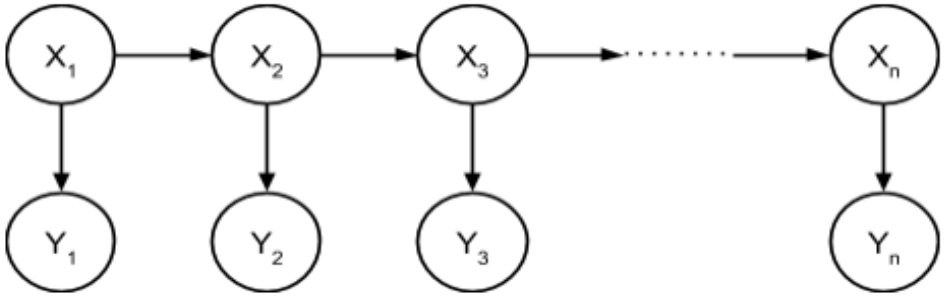

Figure 2.2 Trellis Graph Representation of a HMM

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**.

2.1.4 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.

2.2 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]

2.2.1 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.

2.2.2 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, V_{MAP}, 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.

2.2.3 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.

2.2.4 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.

2.2.5 Recent Research in Network Anomaly Detection in NBC

- **A System Approach to Network Modeling 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 2.3 They report improvements in accuracy with increase in window size.

  The concept of windowing, formation of bands and grouping of packets based on TCP flags was incorporated into the proposed system. For the grouping of likely events, the Jump Clustering Algorithm was used as suggested by the author.

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 2.3 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 Bayesian Networks and therefore, concluded that the BNs performed equivalently well in network attacks.

  The classification of attacks was well portrayed in this paper. Various attacks like DoS, Probe, R2L and U2R were used in the experiment. Also, the different methods of detecting these attacks was articulated. The proposed system will concentrate on the DDoS attacks using a Naive Bayes Classifier.

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.

- **An empirical study of the Naive Bayes Classifier by Irina Rish :** In this paper, Monte Carlo simulations were used to conduct a systematic study of classification accuracy for several classes of randomly generated problems. It also analyzes the impact of the distribution entropy on the classification error, showing that low-entropy feature distributions yield good performance of Naïve Bayes. It also shows that the Naive Bayes works with both independent features and functionally dependent features.

  The most important part that the paper discusses is the characteristics of the input data and its effect on the Naive Bayes classifiers performance. The author presents a challenge that the accuracy of the performance depending on the features in a NBC can only be proved my practical implementation of the NBC and proving it with specific inputs. With this, the proposed system has safely assumed these assertions and will be using the Naive Bayes Classifier without a distinction between independent assumptions and functionally dependent assumptions.

- **Adaptive Network Intrusion Detection System using a Hybrid Approach by R Rangadurai Karthick, Vipul P. Hattiwale, Balaraman Ravindran:** In this paper, the authors discuss Intrusion Detection Systems in detail and their role in the detection of DDoS attacks. They assert that two key criteria should be met by an IDS for it to be effective: (i) ability to detect unknown attack types, (ii) having very less miss classification rate. They also describe an adaptive network intrusion detection system, that uses a two stage architecture. 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.

  Considering the first phase proposed in this paper, the proposed system implements only the online classification phase. This is achieved only by implementing a Naive Bayes Classifier for the incoming network traffic. The classifier will classify the incoming traffic into either attack or normal based on the training data.

2.3 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.