RESEARCH ARTICLE OPEN ACCESS

A Comprehensive Study of Recent Advancements in Methods and Tools for Network Anomaly Detection

Koustubha Balakrishnan M [1]
Research Scholar, Gondwana University, Maharashtra, India
Dr. Vilas Maruti Ghodki [2]
Associate Prof., Dept. of Computer Science, J.B. College, Wardha, Maharashtra, India
Kuruvikulam Chandrasekaran [3]
Academics, Asia Pacific University of Technology and Innovation, Kuala Lumpur, Malaysia

ABSTRACT
The recent advancements in Internet technologies along with the exponential and incessant rise in the number of network attacks has made network intrusion detection a significant and a dynamic research issue. Though many remarkable network intrusion detection methods and systems (NIDS) have been proposed in the past literature there are still many potential opportunities to enhance the state-of-the-art for intrusion detection. In this paper, we provide a systematic and comprehensive classification of various anomaly detection methods, systems, tools and classification of attacks and their characteristics. The paper also includes an overview of crucial feature extraction techniques, performance metrics, in addition to a discussion of the datasets used for evaluation of any IDS.

Keywords:- Anomaly detection, NIDS, attack, dataset, intrusion detection, classifier, tools

I. INTRODUCTION

Security in computer networks is an extremely active and broad area of research, as networks of all sizes are targeted daily by attackers seeking to disrupt or disable network traffic. A successful denial-of-service (DoS) attack degrades network performance, resulting in losses of several millions of dollars [14]. Hence, development of methods to counter these and other threats is of high interest. Recent countermeasures under development focus on detection of anomalies and intrusions, their prevention, or a combination of both. An intrusion attempt or a threat is an intentional and unauthorized attempt to do any of the following actions : (i) access information, (ii) manipulate information, or (iii) render a system unreliable or unusable. As an example, (a) Denial of Service (DoS) attack is a type of network intrusion attack which attempts to starve a host of its resources, which are needed to function correctly during processing; (b) Worms and viruses exploit other hosts through the network; and (c) Compromises obtain privileged access to a host by taking advantages of known vulnerabilities. A list of various network attacks is given in Table 1.

The term anomaly-based intrusion detection in networks refers to the problem of finding exceptional patterns in network traffic that do not conform to the expected normal behavior. These nonconforming patterns are often referred to as anomalies, outliers, exceptions, aberrations, surprises, peculiarities or discordant observations in various application domains [3], [4]. Out of these, anomalies and outliers are two of the most commonly used terms in the context of anomaly-based intrusion detection in networks. The applications of anomaly detection are pretty extensive in areas such as fraud detection for credit cards, intrusion detection for cyber security,and military surveillance for enemy activities and also extends to medical field too. For example, an anomalous traffic pattern in a computer network may mean that a hacked computer is sending out sensitive data to an unauthorized host. Intrusion can be attempted by an inside or outside agent to gain unauthorized entry and control of the security mechanism. To protect infrastructure of network systems, intrusion detection systems (IDSs) provide well-established mechanisms, which gather and analyze information from various areas within a host or a network to identify possible security breaches. Intrusion detection functions span from (i) monitoring and analyzing user, system, and network activities, (ii) configuring systems for generation of reports of possible vulnerabilities, (iii) assessing system and file integrity (iv) recognizing patterns of typical attacks (v) analyzing abnormal activity, to (vi) tracking user policy violations. Intrusion

detection works on the assumption that intrusion activities are noticeably different from normal system activities and thus detectable. Basically, IDs can be categorized on their deployment in real-time. ***(a) Host-based IDS (HIDS)***: A HIDS monitors and analyzes the internals of a computing system rather than its external interfaces [12]. A HIDS might detect internal activity such as which program accesses what resources and attempts illegitimate access. ***(b) Network-based IDS (NIDS)***: An NIDS deals with detecting intrusions in network data. Intrusions typically occur as anomalous patterns though certain techniques model the data in a sequential fashion and detect anomalous patterns.

Attack name	Characteristics	
Virus	A self replicating program that infects the system without any knowledge or permission from the user.	Trivial.88.D, Polyboot.B, Tuareg
Worm	A self replicating program that propagates through network services on computer systems without user intervention	SQL Slammer,Mydoom,CodeRed, NImda
Trojan	A malicious program that cannot replicate itself but can cause serious security problems in the computer system	Example-Mail Bomb, Phishing attack
Denial of Service attack (DoS)	Attempts to block access to system or network resources. The loss of service is the inability of a particular network or a host service, such as e-mail to function.	Buffer Overflow, Ping od Death,TCP SYN, smurf, teardrop
Network Attack	Any process used to maliciously attempt to compromise the security of the network ranging from the data link layer to the application layer by various means such as manipulation of network protocols.	Packet Injection, SYN flood
Physical Attack	An attempt to damage the physical components of networks or computers.	Cold boot, evil maid
Password Attack	Aims to gain a password within a short period of time, and is usually indicated by a series of login failures.	Dictionary attack.SQL injection attack
Information Gathering Attack	Gathers information or finds known vulnerabilities by scanning or probing computers or networks.	SYS scan, Fin Scan, Xmas Scan
User to Root (U2R) attack	It is able to exploit vulnerabilities to gain privileges of super user of the system while starting as a normal user on the system.	RootKit, Loadmodule, perl
Remote to Local attack(R2L)	Ability to send packets to a remote system over a network without having any account on that system, gain access either as a user or as a root to the system and do harmful operations.	Warezclient,warezmaster,imap,ftp_write,multihop,phf,spy
Probe	Scans the networks to identify valid IP addresses and to collect information about host (e.g., what services they offer, operating system used).	IPsweep, portsweep

Table 1: Classification of network attacks

The primary reason for these anomalies is attacks launched by outside attackers who want to gain unauthorized access to the network to steal information or to disrupt the network. Intrusion detection techniques can be classified into three types based on the detection mechanism [1], [3]. This includes (i) *misuse-based*, (ii) *anomaly-based*, and (iii) *hybrid* . Today, researchers mostly concentrate on anomaly-based network intrusion detection because it can detect known as well as un-known attacks.

A. The Problem of Anomaly Detection

We need to first understand the concept of normality to provide an appropriate solution in network anomaly detection. Normality is defined by a formal model that expresses relations among the fundamental variables involved in the system dynamics. Consequently, an event or an object is detected as anomalous if its degree of deviation with respect to the profile or behavior of the system, specified by the normality model, is high enough. For example, let us take an anomaly detection system S that uses a supervised approach. It can be thought of as a pair S = (M, D), where M is the

model of normal behavior of the system and D is a proximity measure that allows one to compute, given an activity record, the degree of deviation that such activities have with regard to the model M. Thus, each system has mainly two modules: (i) a modeling module and (ii) a detection module. The modeling module trains the systems to get the normality model M. This obtained model is subsequently used by the detection module to evaluate new events or objects or traffic as anomalous or outliers. It is the measurement of deviation that allows classification of events or objects as anomalous or outliers. In this section, we present a Generic Architecture of ANIDS in Figure No.1. The architecture contains an anomaly detection engine which is the heart of any network intrusion detection system. It attempts to detect occurrence of any intrusion either online or offline. However, before sending any network traffic to the detection engine, it needs preprocessing. If the attacks are known, they can be detected using the misuse detection approach. On the other hand, unknown attacks can be detected using the anomaly-based approach based on an appropriate matching mechanism.

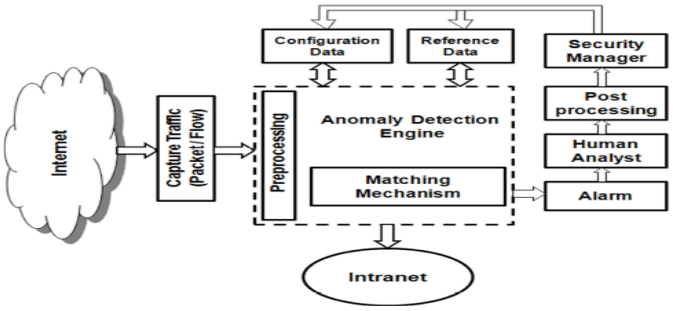


Figure 1 : A generic architecture of ANIDS

The matching mechanism looks for a particular pattern or profile in network traffic that can be built by continuous monitoring of network behavior including known exploits or vulnerabilities. The reference data stores information about known intrusion signatures or profiles of normal behavior. Reference data needs to be stored in an efficient manner. Possible types of reference data used in the generic architecture of NIDS are: profile, signature and rule. In case of ANIDS, it is mostly profiles. The processing elements update the profiles as new knowledge about the observed behavior becomes available. These updates are performed in regular intervals in a batch oriented fashion. Configuration data corresponds to intermediate results, e.g., partially created intrusion signatures. The space needed to store such information can be quite large. Intermediate results need to be integrated with existing knowledge to produce consistent, up-to-date results. The alarm component of the architecture is responsible for generation of alarm based on the indication received from the detection engine.

A human analyst is responsible for analysis, interpretation and takes necessary steps to diagnose the alarm information as a post-processing activity to support reference or profile updation with the help of security manager. The post-processing module is an important module in a NIDS for post-processing of the generated alarms for diagnosis of actual attacks. Traffic capturing is an important module in a NIDS. The raw traffic data is captured at both packet and flow levels. Packet level traffic can be captured using a common tool, e.g., Wireshark and then preprocessed before sending into the detection engine. Flow level data in high speed networks, is comprised of information summarized from one or more packets. Some common tools to capture flow level network traffic include Nfdump, NfSen, and Cisco Netflow V.9. Stored intrusion signatures are updated by the Security Manager (SM) as and when new intrusions become known. The analysis of novel intrusions is a highly complex task.

II. KEY ASPECTS OF NETWORK ANOMALY DETECTION

In this section, we present some important aspects of anomaly-based network intrusion detection. Basically, the network intrusion detection problem is a classification or clustering problem formulated with the following components [3]: (a) types of input data, (b) appropriateness of proximity measures, (c) labelling of data, (d) classification of methods based on the use of labelled data, (e) relevant feature identification and (f) reporting anomalies. We discuss each of these topics in brief.

1) ***Types of input data***: A key aspect of any anomaly-based network intrusion detection technique is the nature of the input data used for analysis. Input is generally a collection of data instances (also referred to as objects, records, points, vectors, patterns, events, cases, samples, observations, entities). Each data instance may consist of only one attribute (univariate) or multiple attributes (multi-variate). In the case of multivariate data instances, all attributes may be of the same type or may be a mixture of data types. The nature of attributes determines the applicability of anomaly detection techniques.

2) ***Appropriateness of proximity measures***: Proximity (similarity or dissimilarity) measures are necessary to solve many pattern recognition problems in classification and clustering. Distance is a quantitative degree of how far apart two objects are. Distance measures that satisfy metric properties [27] are simply called metric while other non-metric distance measures are occasionally called divergence. The choice of a proximity measure depends on the measurement type or representation of objects. Generally,

proximity measures are functions that take arguments as object pairs and return numerical values that become higher as the objects become more alike. A proximity measure is usually defined as follows.

Name	Measure ($S_i(x_i, y_i)$)
Euclidean	$\sqrt{\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2}$
Squared Euclidean	$\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2$
Minkowski	$\sqrt[p]{\sum_{i=1}^{d} \lvert x_i - y_i \rvert^2}$
Jaccard	$\frac{\sum_{i=1}^{d} x_i y_i}{\sum_{i=1}^{d} x_i^2 + \sum_{i=1}^{d} y_i^2 - \sum_{i=1}^{d} x_i y}$
Mahalanobis	$\sqrt{(x-y)^t \sum^{-1} (x-y)}$
Pearson	$\sum_{i=1}^{d} (x_i - y_i)^2$
Chebyshev	$\lvert x_i - y_i \rvert$
Cosine	$\frac{\sum_{i=1}^{d} x_i y_i}{\sum_{i=1}^{d} x_i^2 \sum_{i=1}^{d} y_i^2}$

Table 1: Proximity measures

Definition 3.1: *A proximity measure S is a function X →X ! R that has the following properties [47].*
– Positivity: $8_{x,y}$ 2 X, S(x, y) ≥ 0
– Symmetry: $8_{x,y}$ 2 X, S(x, y) = S(y, x)
– Maximality: $8_{x,y}$ 2 X, S(x, x) ≥ S(x, y)
where X is the data space (also called the universe) and x, y are the pair of k-dimensional objects. The most common proximity measures for numeric, categorical and mixed type data are listed in the Table 2.

3) ***Labelling of data:*** The label associated with a data instance denotes if that instance is normal or anomalous. It should be noted that obtaining accurate labeled data of both normal or anomalous types is often prohibitively expensive. Labeling is often done manually by human experts and hence substantial effort is required to obtain the labeled training dataset [3]. Moreover, anomalous behavior is often dynamic in nature, e.g., new types of anomalies may arise, for which there is no labeled training data.

4) ***Classification of methods based on use of labeled data***: Based on the extent to which labels are available, anomaly detection techniques can operate in three modes: *supervised, semi-supervised and unsupervised*. In supervised mode, one assumes the availability of a training dataset which has labeled instances for the normal as well as the anomaly class. The typical approach in such cases is to build a predictive model for normal vs. anomaly classes. Any unseen data instance is compared against the model to determine which class it belongs to. There are two major issues that arise in supervised anomaly detection. First, anomalous instances are far fewer compared to normal instances in the training data. Issues that arise due to imbalanced class distributions have been addressed in data mining and machine learning literature. Second, obtaining accurate and representative labels, especially for the anomaly class, is usually challenging. A number of techniques inject artificial anomalies in a normal dataset to obtain a labeled training dataset. Semi-supervised techniques assume that the training data has labeled instances for only the normal class. Since they do not require labels for the anomaly class, they can be more readily used compared to supervised techniques. Finally, unsupervised techniques do not require training data, and thus are potentially most widely applicable. The techniques in this category make the implicit assumption that normal instances are far more frequent than anomalies in the test data . When this assumption is not true, such techniques suffer from high false alarm rates. Many semi-supervised techniques can be adapted to operate in an unsupervised mode by using a sample of the unlabeled dataset as training data. Such adaptation assumes that the test data contains very few anomalies and the model learnt during training is robust to these few anomalies.

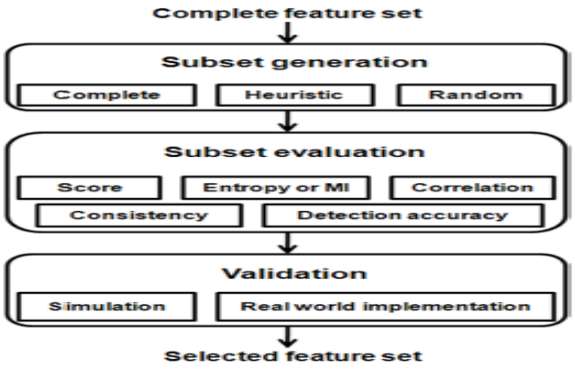


Figure 2: A framework for feature extraction

5) ***Relevant feature identification***: Feature selection plays an important role in detecting network anomalies.

Feature selection methods are used in the intrusion detection domain for eliminating unimportant or irrelevant features. Feature selection reduces computational complexity, removes information redundancy, increases the accuracy of the detection algorithm, facilitates data understanding and improves generalization. The feature selection process includes three major steps: (a) sub-set generation, (b) subset evaluation and (c) validation. Three different approaches for subset generation are: *complete, heuristic and random*. Evaluation functions are categorized into five distinct categories: *score-based, entropy or mutual based, correlation-based, consistency-based* and *detection accuracy-based*. Simulation and real world implementation are the two ways to validate the evaluated subset. A conceptual framework of the feature selection process is shown in Figure 3. Feature selection algorithms have been classified into three types: *wrapper, filter* and *hybrid* methods. While wrapper methods try to optimize some predefined criteria with respect to the feature set as part of the selection process, filter methods rely on the general characteristics of the training data to select features that are independent of each other and are highly dependent on the output. The hybrid feature selection method attempts to exploit the salient features of both wrapper and filter.

6) Reporting anomalies: An important aspect of any anomaly detection technique is the manner in which anomalies are reported [3]. Typically, the outputs produced by anomaly detection techniques are of two types: **(a) a score**, which is a value that combine (i) distance or deviation with reference to a set of profiles or signatures,(ii) influence of the majority in its neighborhood, and (iii) distinct dominance of the relevant subspace. **(b) a label**, which is a value (normal or anomalous) given to each test instance.

III. METHODS FOR NETWORK ANOMALY DETECTION

This section discusses about the classification of various network anomaly detection methods and systems. These methods are categorized based on the nature of the underlying algorithms. The classification of classification scheme for network anomaly detection methods and systems is not pretty straightforward as there is substantial overlap among the methods. However, we have decided on six distinct classes of methods and systems as shown in Figure 3.

1. ***Statistical Methods:*** Normally, statistical methods fit a statistical model (usually for normal behavior) to the given data and then apply a statistical inference test to determine if an unseen instance belongs to this model. Instances that have a low probability to be generated from the learnt model based on the applied test statistic are declared anomalies.

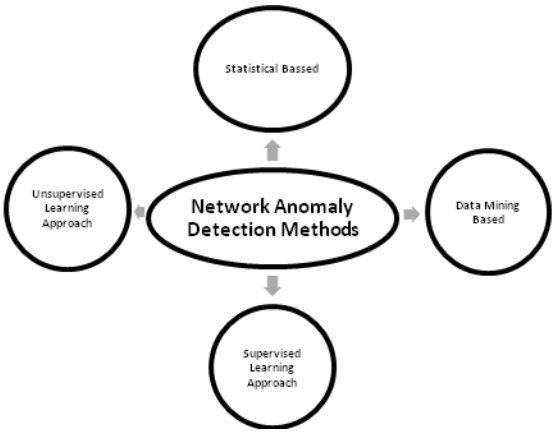


Figure 3 : A taxonomy of Network Anomaly Detection Methods

Both parametric and non-parametric techniques have been applied to design statistical models for anomaly detection. While parametric techniques assume knowledge of the underlying distribution and estimate the parameters from the given data , non-parametric techniques do not generally assume knowledge of the underlying distribution . As an example of a statistical IDS is HIDE. HIDE is an anomaly-based network intrusion detection system, that uses statistical models and neural network classifiers to detect intrusions. HIDE is a distributed system, which consists of several tiers with each tier containing several Intrusion Detection Agents (IDAs). IDAs are IDS components that monitor the activities of a host or a network. The probe layer collects network traffic at a host or in a network, abstracts the traffic into a set of statistical variables to reflect network status, and periodically generates reports to the event preprocessor. The event preprocessor layer receives reports from both the probe and IDAs of lower tiers, and converts the information into the format required by the statistical model. The statistical processor maintains a reference model of typical network activities, compares reports from the event preprocessor with the reference models, and forms a stimulus vector to feed into the neural network classifier. The neural network classifier analyzes the stimulus vector from the statistical model to decide whether the network traffic is normal. The post-processor generates reports for the agents at higher tiers. One of the major advantage of statistical approaches is that they do not have prior knowledge of normal activities of the target system.

2. ***Data Mining Based Methods:*** Data mining techniques construct rules describing normal network behaviors. The rules include association rules that describe frequency associations between any two fields of the network record database and also frequent episodes that describe the frequency with which a field takes a certain value after two other fields have particular values in a definite time interval. Deviations from these rules (outliers) indicate an attack on the network. One such method is classification. Classification is the problem of identifying which of a set of categories a new observation belongs to, on the basis of a training set of data containing observations whose category membership is known. Assuming we have two classes whose instances are shown as + and −, and each object can be defined in terms of two attributes or features x_1 and x_2 , linear classification tries to find a line between the classes as shown in Figure 5(a). The classification boundary may be non-linear as in Figure 5(b).

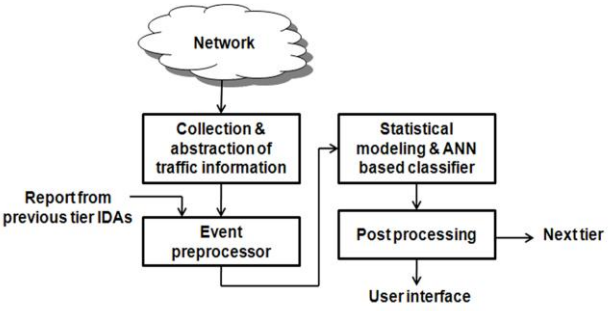


Figure 4 : Architecture of HIDE IDS

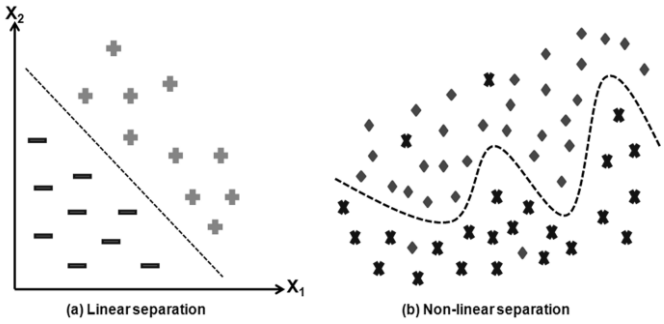


Figure 5: Linear and non-linear classification in 2-D

Several classification-based techniques such as k-nearest neighbor, support vector machines (SVM), and decision trees have been applied to anomaly detection in network traffic data.

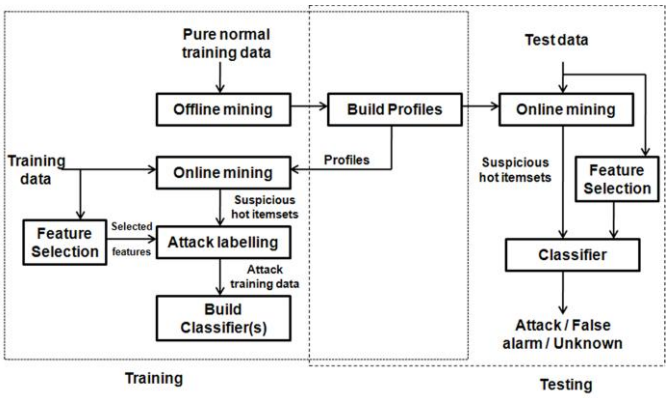


Figure 6: Architecture of ADAM IDS

An example of classification-based IDS is Automated Data Analysis and Mining (ADAM) [30] that provides a testbed for detecting anomalous instances. An architecture diagram of ADAM is shown in Figure 7. ADAM exploits a combination of classification techniques and association rule mining to discover attacks in a tcpdump audit trail. First, ADAM builds a repository of "normal" frequent itemsets from attack-free periods. Second, ADAM runs a sliding-window based online algorithm that finds frequent itemsets in the connections and compares them with those stored in the normal itemset repository, discarding those that are deemed normal. ADAM uses a classifier which has been trained to classify suspicious connections as either a known type of attack or an unknown type or a false alarm.

4. Supervised Learning-Based Approaches: Recently, methods from machine learning and pattern recognition have been utilized to detect intrusions. Supervised learning and unsupervised learning are both used. For supervised learning for intrusion detection, there are mainly supervised neural network (NN)-based approaches [17], [24], and support vector machine (SVM)-based approaches [7], [25].

a) ANN-based approaches: Artificial Neural Networks (ANN) are motivated by the recognition that the human brain computes in an entirely different way from the conventional digital computer. The brain organizes its constituents, known as neurons, so as to perform certain computations (e.g., pattern recognition, perception, and motor control) many times faster than the fastest digital computer. To achieve good performance, real neural networks employ massive interconnections of neurons. Neural networks acquire

knowledge of the environment through a process of learning, which systematically changes the interconnection strengths, or synaptic weights of the network to attain a desired design objective.

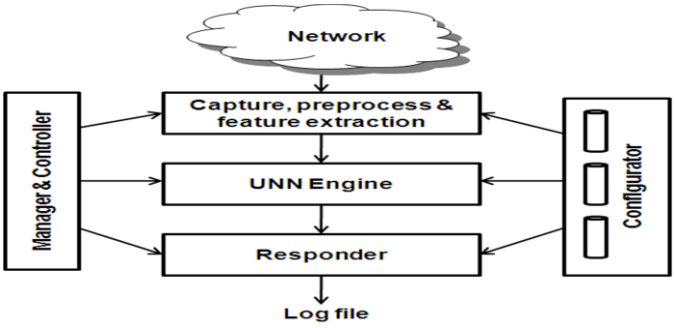


Figure 7: Architecture of RT-UNNID

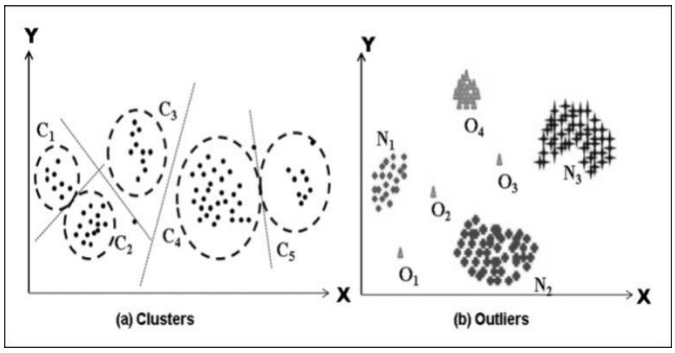
An example of ANN-based IDS is RT-UNNID. This system is capable of intelligent real time intrusion detection using unsupervised neural networks (UNN). The architecture of RT-UNNID is given in Figure 8. The first module captures and preprocesses the real time network traffic data for the protocols: TCP, UDP and ICMP. It also extracts the numeric features and converts them into binary or normalized form. The converted data is sent to the UNN-based detection engine that uses Adaptive Resonance Theory (ART) and Self-Organizing Map (SOM) neural networks. Finally, the output of the detection engine is sent to the responder for recording in the user's system log file and to generate alarm when detecting attacks. RT-UNNID can work in real time to detect known and unknown attacks in network traffic with high detection rate.

b) SVM-based approaches: SVMs are used to distinguish between normal network behaviors and intrusions and further identify important features for intrusion detection. Zhang and Shen [25] propose an approach for online training of SVMs for real-time intrusion detection based on an improved text categorization model. Aside from the aforementioned unsupervised-learning-based approaches for intrusion detection, decision tree and discriminant analysis are applied to detect intrusions.

IV. UNSUPERVISED LEARNING-BASED APPROACHES

Supervised learning methods for intrusion detection can only detect known intrusions. Unsupervised learning methods can detect the intrusions that have not been previously learned. Examples of unsupervised learning for intrusion detection include *K*-means-based approaches and self-organizing feature map (SOM)-based approaches [3], [9].

a) K-means-based approaches: Guan *et al.* propose a *K*-means-based clustering algorithm, which is named *Y* -means, for intrusion detection. Xian *et al.* [16] combine the fuzzy *K*-means method and a clonal selection algorithm to detect intrusions. Jiang *et al.* [17] use the incremental clustering algorithm that is an extension of the *K*-means algorithm to detect intrusions. Clustering is the task of assigning a set of objects into groups called clusters so that the objects in the same cluster are more similar in some sense to each other than to those in other clusters. Clustering is used in explorative data mining. For example, if we have a set of unlabeled objects in two dimensions, we may be able to cluster them into 5 clusters by drawing circles or ellipses around them, as in Figure 8(a). Outliers are those points in a dataset that are highly unlikely to occur given a model of the data, as in Figure 8(b). Clustering and outlier finding are examples of unsupervised machine learning.



Clustering can be performed in network anomaly detection in an offline environment. For example, MINDS (Minnesota Intrusion Detection System) is a data mining-based system for detecting network intrusions. The architecture of MINDS is given in Figure 10. It accepts NetFlow data collected through flow tools as input. Flow tools only capture packet header information and build one way sessions of flows. The analyst uses MINDS to analyze these data files in batch mode. The reason for running the system in batch mode is not due to the time it takes to analyze these files, but because it is convenient for the analyst to do so. Before data is fed into the anomaly detection module, a data filtering step is executed to remove network traffic in which the analyst is not interested.

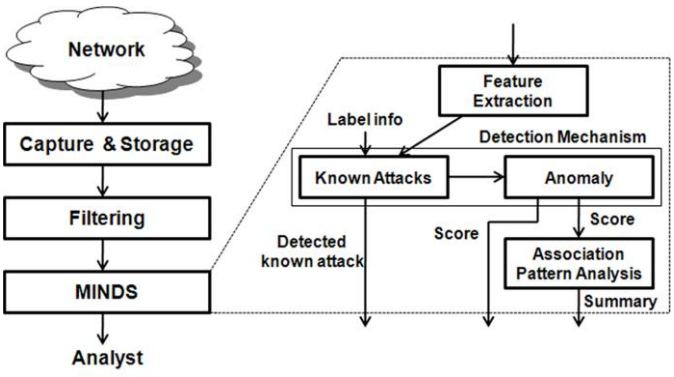


Figure 9: Architecture of MNDS IDS

The first step of MINDS is to extract important features that are used. Then, it summarizes the features based on time windows. After the feature construction step, the known attack detection module is used to detect network connections that correspond to attacks for which signatures are available, and to remove them from further analysis. Next, an outlier technique is activated to assign an anomaly score to each network connection. A human analyst then looks at only the most anomalous connections to determine if they are actual attacks or represent other interesting behavior. The analyst provides feedback after analyzing the summaries created and decides whether these summaries are helpful in creating new rules that may be used in known attack detection. Clustering techniques are frequently used in anomaly detection. Some examples include single-link clustering algorithms, k-means (squared error clustering), and hierarchical clustering algorithms.

Tool Name	Purpose	Source
Wireshark	Packet Capture	http://www.wireshark.org/
Gulp	Lossless gigabit remote packet capturing	http://staff.washington.edu/corey/gulp/
tcptrace	TCP-based feature extraction	http://jarok.cs.ohiou.edu/ software/tcptrace/
nfdump	netflow data collection	http://nfdump.sourceforge.net/
nmap	Scanning port	http://nmap.org/
rnmap	Coordinated scanning	http://rnmap.sourceforge. net/
Targa	Attack simulation	http://www10.0rg/cdrom/papers/409

b) SOM-based approaches: Hoglund *et al.* extract features that describe network behaviors from audit data, and they use the SOM to detect intrusions. Kayacik *et al.* [9] propose a hierarchical SOM approach for intrusion detection. Specific attention is given to the hierarchical development of abstractions, which is sufficient to permit direct labeling of SOM nodes with connection type. The classification capability of the SOM on selected dimensions of the data set to detect anomalies can be efficiently utilized. Their results are among the best known for intrusion detection.

V. PERFORMANCE EVALUATION CRITERIA

To evaluate performance, it is important that the system identifies the attack and normal data correctly. In this section, we describe the several datasets and evaluation measures available for evaluating network anomaly detection methods and systems.

A. Tools for Capturing Network traffic

Capturing and preprocessing high speed network traffic is essential prior to detection of network anomalies. Different tools are used for capture and analysis of network traffic data. We list a few commonly used tools and their features in Table 3.

B. Datasets

1. Synthetic datasets: Synthetic datasets are generated to meet specific needs or conditions or tests that real data satisfy. This can be useful when designing any type of system for theoretical analysis so that the design can be refined. This allows for finding a basic solution or remedy, if the results prove to be satisfactory. Synthetic data is used in testing and creating many different types of test scenarios. It enables designers to build realistic behavior profiles for normal users and attackers based on the generated dataset to test a proposed system.

2. Benchmark datasets: In this subsection, we presenting simulated environments that include a number of networks and by executing different attack scenarios. **(a) KDDcup99 dataset**: Since 1999, the KDDcup99 dataset [25] has been the most widely used dataset for the evaluation of network-based anomaly detection methods and systems. The KDD training

dataset consists of approximately 4, 900, 000 single connection vectors, each of which contains 41 features and is labeled as either normal or attack with a specific attack type. The test dataset contains about 300, 000 samples with 24 training attack types, with an additional 14 attack types in the test set only. The names and descriptions of the attack types are available .

(b) NSL-KDD dataset: Analysis of the KDD dataset showed that there were two important issues in the dataset, which highly affect the performance of evaluated systems resulting in poor evaluation of anomaly detection methods. To solve these issues, a new dataset known as NSL-KDD[23], consisting of selected records of the complete KDD dataset was introduced. **(c) DARPA 2000 dataset:** A DARPA evaluation project targeted the detection of complex attacks that contain multiple steps. Two attack scenarios were simulated in the 2000 evaluation contest, namely, LL-DOS (Lincoln Laboratory scenario DDoS) 1.0 and LL-DOS 2.0. To achieve the necessary variations, these two attack scenarios were carried out over several network and audit scenarios. These sessions were grouped into four attack phases: (a) probing, (b) breaking into the system by exploiting vulnerability, (c) installing DDoS software for the compromised system and (d) launching DDoS attack against another target. LLDOS 2.0 is different from LLDOS 1.0 in the sense that attacks are more stealthy and thus harder to detect. Since this dataset contains multistage attack scenarios, it is also commonly used for evaluation of alert correlation methods. **(d) DEFCON dataset:** The DEFCON dataset is an-other commonly used dataset for evaluation of IDSs . It contains network traffic captured during the hacker competition called Capture The Flag (CTF), in which competing college teams are divided into two groups: attackers and defenders. The traffic produced during CTF is very different from real world network traffic since it contains only intrusive traffic without any normal background traffic. Due to this limitation, the DEFCON dataset has been found useful in evaluating alert correlation techniques.**(e) CAIDA dataset**: CAIDA collects many different types of data and makes it available to the research community. Most CAIDA datasets are very specific to particular events or attacks (e.g., CAIDA DDoS attack 2007 dataset). All backbone traces are anonymized and do not have payload information. **(f) LBNL dataset** : LBNL's (Lawrence Berkeley National Laboratory) internal enterprise traces are full header network traces , without payload. This dataset has undergone heavy anonymization to the extent that scanning traffic was extracted and separately anonymized to remove any information which could identify individual IPs. The packet traces were obtained at the two central routers of the LBNL network and they contain more than one hundred hours of traffic generated from several thousand internal hosts.

3. Real life datasets: In this subsection, we present three real life datasets created by collecting network traffic on several days, which include both normal as well as attack instances in appropriate proportions in the authors' respective campus networks.(a) **UNIBS dataset**: The UNIBS packet traces were collected on the edge router of the campus network of the University of Brescia, Italy, on three consecutive working days. It includes traffic captured or collected and stored through 20 workstations running the GT client daemon. The authors collected the traffic by running tcp-dump on the faculty router, which was a dual Xeon Linux box that connected their network to the Internet through a dedicated 100Mb/s uplink. The traces were captured and stored on a dedicated disk of a workstation connected to the router through a dedicated ATA controller.**(b) ISCX-UNB dataset**: Real packet traces were analyzed to create profiles for agents that generate real traffic for HTTP, SMTP, SSH, IMAP, POP3 and FTP protocols. Various multi-stage attack scenarios were explored for generating malicious traffic.

B. Evaluation Measures

An evaluation of a method or a system in terms of accuracy or quality is a snapshot in time. As time passes, new vulnerabilities may evolve, and current evaluations may become irrelevant. In this section, we discuss various measures used to evaluate network intrusion detection methods and systems.

1) Accuracy: Accuracy is a metric that measures how correctly an IDS works, measuring the percentage of detection and failure as well as the number of false alarms that the system produces. If a system has 80% accuracy, it means that it correctly classifies 80 instances out of 100 to their actual classes. The following are the some accuracy measures.**(a) Sensitivity and Specificity:** These two measures attempt to measure the accuracy of classification for a 2-class problem. When an IDS classifies data, its decision can be either right or wrong. It assumes true for right and false for wrong, respectively. If S is a detector and D_t is the set of test instances, there are four possible outcomes described using the confusion matrix given in Figure 10.

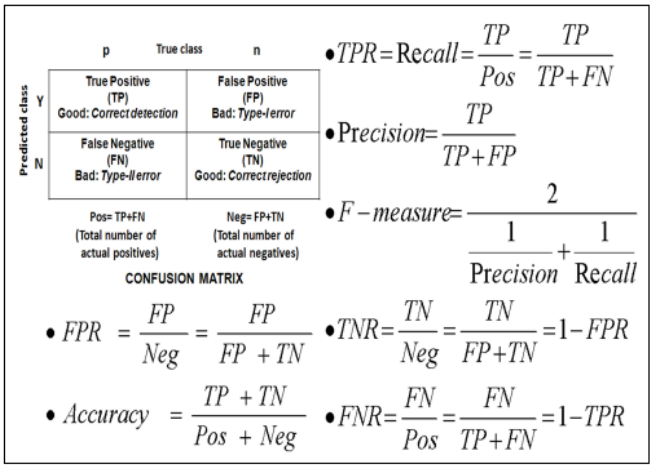


Figure 10: Confusion matrix and their related evaluation measures

When an anomalous test instance (p) is predicted as anomalous (Y) by the detector S, it is counted as true positive (TP); if it is predicted as normal (N), it is counted as false negative (FN). On the other hand, if a normal (n) test instance is predicted as normal (N) it is known as true negative (TN), while it is a false positive (FP) if it is predicted as

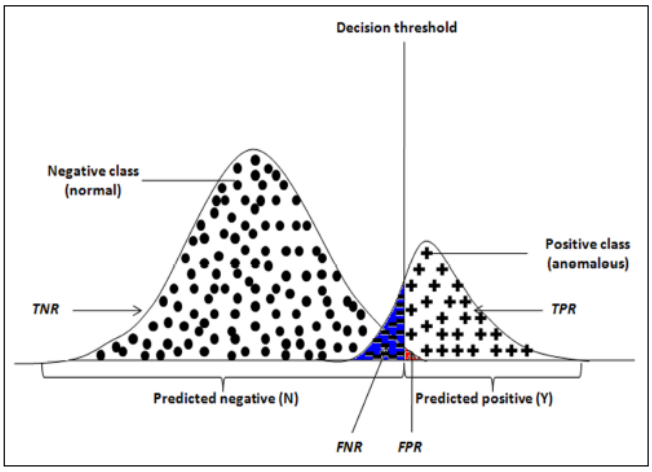


Figure 11: Illustration of confusion matrix in terms of their related evaluation measures

classified correctly over the total number of anomalous instances present in the test data. TPR is also known as sensitivity. The false positive rate (FPR) is the proportion of normal instances incorrectly classified as anomalous over the total number of normal instances contained in the test data. The true negative rate (TNR) is also called specificity. TPR, FPR, TNR, and the false negative rate (FNR) can be defined for the normal class. We illustrate all measures related to the confusion matrix in Figure 11. Sensitivity is also known as the hit rate. Between sensitivity and specificity, sensitivity is set at high priority when the system is to be protected at all cost, and specificity gets more priority when efficiency is of major concern [27]. Consequently, the aim of an IDS is to produce as many TPs and TNs as possible while trying to reduce number of both FPs and FNs. The majority of evaluation criteria use these variables and the relations among them to model the accuracy of the IDSs. **(b) ROC Curves:** The Receiver Operating Characteristics (ROC) analysis originates from signal processing theory. Its applicability is not limited only to intrusion detection, but extends to a large number of practical fields such as medical diagnosis, radiology, bioinformatics as well as in artificial intelligence and data mining. In intrusion detection, ROC curves are used on the one hand to visualize the relation between TP and FP rates of a classifier while tuning it and also to compare the accuracy with two or more classifiers. The ROC space uses the orthogonal coordinate system to visualize the classifier accuracy. Figure 12 illustrates the ROC approach normally used for network anomaly detection methods and systems evaluation. **(c) Misclassification rate:** This measure attempts to estimate the probability of disagreement between the true and predicted cases by dividing the sum of FN and FP by the total number of pairs observed, i.e., (TP+FP+FN+TN). In other words, misclassification rate is defined as (FN+FP)/(TP+FP+FN+TN).**(d) Confusion Matrix:** The confusion matrix is a ranking method that can be applied to any kind of classification problem. The size of this matrix depends on the number of distinct classes to be detected. The aim is to compare the actual class labels against the predicted ones as shown in Figure 12. The diagonal represents correct classification. The confusion matrix for intrusion detection is defined as a 2-by-2 matrix, since there are only two classes known as intrusion and normal. Thus, the TNs and TPs that represent the correctly predicted cases lie on the matrix diagonal while the FNs and FPs are on the right and left sides. As a side effect of creating the confusion matrix, all four values are displayed in a way that the relation between them can be easily understood.

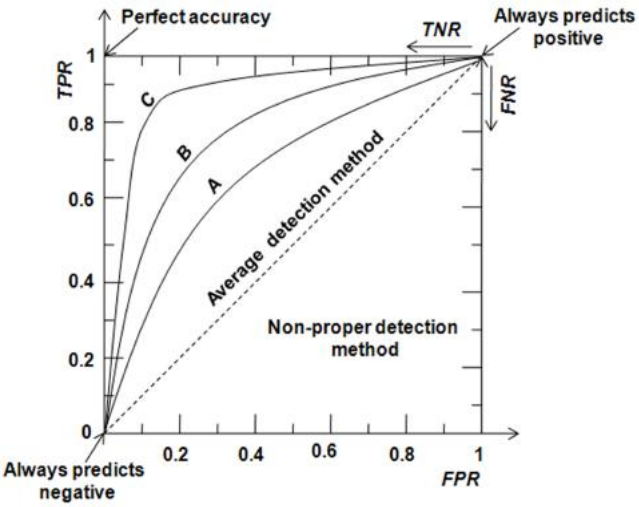


Figure 12: Illustration of ROC measure where A,B,C represents the accuracy of a detection method or a system in ascending order.

(e) Precision, Recall and F-measure: Precision is a measure of how a system identifies attacks or normals. A flagging is accurate if the identified instance indeed comes from a malicious user, which is referred to as true positive. The final quantity of interest is recall, a measure of how many instances are identified correctly (see Figure 12). Precision and recall are often inversely proportional to each other and there is normally a trade-off between these two ratios. An algorithm that produces low precision and low recall is most likely defective with conceptual errors in the underlying theory. The types of attacks that are not identified can indicate which areas of the algorithm need more attention. Exposing these flaws and establishing the causes assist future improvement. The F-measure mixes the properties of the previous two measures as the harmonic mean of precision and recall . If we want to use only one accuracy metric as an evaluation criterion, F-measure is the most preferable. Note that when precision and recall both reach 100%, the F-measure is the maximum, i.e., 1 meaning that the classifier has 0% false alarms and detects 100% of the attacks. Thus, a good classifier is expected to obtain F-measure as high as possible.

VI. OPEN ISSUES AND CHALLENGES

Although, many methods and systems have been developed by the research community, there are still a number of open research issues and challenges. The suit-ability of performance metrics is a commonly identified drawback in intrusion detection. In evaluating IDSs, the three most important qualities that need to be measured are completeness, correctness, and performance. The current state-of-the-art in intrusion detection restricts evaluation of new systems to tests over incomplete datasets and micro-benchmarks that test narrowly defined components of the system. A number of anomaly-based systems have been tested using contrived datasets. Such evaluation is limited by the quality of the dataset that the system is evaluated against. Construction of a dataset which is unbiased, realistic and comprehensive is an extremely difficult task. After a study of existing NIDSs, we find that it is still extremely difficult to design a new NIDS to ensure robustness, scalability and high performance. In particular, practitioners find it difficult to decide where to place the NIDS and how to best configure it for use within an environment with multiple stakeholders. We sort out some of the important issues as challenges and enumerate them below. Inability to capture and inspect every single packet in real-time.

a) Dependence on the set-up environment.
b) Continuously varying nature of attacks evade existing intrusion detection solutions.
c) Avoiding a high rate of false alarms.
d) Dynamic updation of profiles in anomaly-based NIDSs without conflict and without compromising performance is an important task.
e) Preparing an unbiased network intrusion dataset with all normal variations in profiles is another challenging task. The number of normal instances is usually large and their proportion with attack instances is very skewed in the existing publicly available intrusion datasets. Only a few intrusion datasets with sufficient amount of attack information are available publicly. Thus, there is an overarching need for benchmark intrusion datasets for evaluating NIDSs and detection methods.
f) Developing an appropriate and fast feature selection method for each attack class is yet another challenge.

VII. CONCLUSION

In this paper, we have discussed the various sources, causes and aspects of network anomalies. We attempt to provide a classification of various anomaly detection methods, systems and tools till date. In addition, we have also emphasized two well-known criteria to classify and evaluate the NIDSs: detection strategy and evaluation datasets. Also, we have discussed several evaluation criteria for testing the performance of a detection method or system. A brief description of the different existing datasets and its taxonomy is also provided.

REFERENCES

[1] A. Sundaram, "An introduction to intrusion detection," Cross-roads, vol. 2, no. 4, pp. 3–7, April 1996.

[2] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," James P Anderson Co, Fort Washington, Penn-sylvania, Tech. Rep., April 1980.

[3] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection : A Survey," ACM Computing Surveys, vol. 41, no. 3, pp. 15:1– 15:58, September 2009.

[4] N. K. Ampah, C. M. Akujuobi, M. N. O. Sadiku, and S. Alam, "An intrusion detection technique based on continuous binary communication channels," International Journal of Security and Networks, vol. 6, no. 2/3, pp. 174–180, November 2011.

[5] F. Y. Edgeworth, "On discordant observations," Philosophy Magazine, vol. 23, no. 5, pp. 364–375, 1887.

[6] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Computer Networks, vol. 51, no. 12, pp. 3448–3470, 2007.

[7] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based network intrusion detection : Techniques, systems and challenges," Computers & Security, vol. 28, no. 1-2, pp. 18–28, 2009.

[8] V. Hodge and J. Austin, "A survey of outlier detection method-ologies," Artificial Intelligence Review, vol. 22, no. 2, pp. 85– 126, 2004.

[9] T. Nguyen and G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," IEEE Communications Surveys & Tutorials, vol. 10, no. 4, pp. 56– 76, 2008.

[10] M. Agyemang, K. Barker, and R. Alhajj, "A comprehensive survey of numeric and symbolic outlier mining techniques," Intelligence Data Analysis, vol. 10, no. 6, pp. 521–538, 2006.

[11] J. Ma and S. Perkings, "Online novelty detection on temporal sequences," in Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003, pp. 613–618.

[12] D. Snyder, "Online intrusion detection using sequences of sys-tem calls," Master's thesis, Department of Computer Science, Florida State University, 2001.

[13] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outlier Detection. John Wiley & Sons, 1987.

[14] V. Barnett and T. Lewis, Outliers in Statistical Data. John Wiley & Sons, 1994.

[15] D. Hawkins, Identification of Outliers. New York: Chapman and Hall, 1980.

[16] R. J. Beckman and R. D. Cook, "Outliers," Technometrics, vol. 25, no. 2, pp. 119–149, 1983.

[17] Z. Bakar, R. Mohemad, A. Ahmad, and M. Andderis, "A comparative study for outlier detection techniques in data mining," in Proc. of the IEEE Conference on Cybernetics and Intelligent Systems, 2006, pp. 1–6.

[18] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita, "A Survey of Outlier Detection Methods in Network Anomaly Identification," Computer Journal, vol. 54, no. 4, pp. 570–588, April 2011.

[19] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A Survey on Internet Traffic Identification," IEEE Communications Surveys & Tutorials, vol. 11, no. 3, pp. 37–52, 2009.

[20] W. Zhang, Q. Yang, and Y. Geng, "A Survey of Anomaly Detection Methods in Networks," in Proc. of the International Symposium on Computer Network and Multimedia Technol-ogy, January 2009, pp. 1–3.

[21] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An Overview of IP Flow-Based Intrusion De-tection," IEEE Communications Surveys & Tutorials, vol. 12, no. 3, pp. 343–356, quarter 2010.

[22] B. Sun, F. Yu, K. Wu, Y. Xiao, and V. C. M. Leung, "Enhancing security using mobility-based anomaly detection in cellular mobile networks," IEEE Transactions on Vehicular Technology, vol. 55, no. 4, pp. 1385 –1396, July 2006.

[23] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," IEEE Wireless Communication Magazine, vol. 14, no. 5, pp. 56–63, October 2007.

[24] B. Sun, Y. Xiao, and R. Wang, "Detection of Fraudulent Usage in Wireless Networks," IEEE Transactions on Vehicular Technology, vol. 56, no. 6, pp. 3912–3923, November 2007.

[25] B. Sun, K. Wu, Y. Xiao, and R. Wang, "Integration of mobility and intrusion detection for wireless ad hoc networks," Inter-national Journal of Communication Systems, vol. 20, no. 6, pp. 695–721, June 2007.

[26] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," ACM Computing Surveys, vol. 39, no. 1, pp. 1–42, April 2007.

[27] M. Al-Kuwaiti, N. Kyriakopoulos, and S. Hussein, "A com-parative analysis of network dependability, fault-tolerance, reliability, security, and survivability," IEEE Communications Surveys & Tutorials, vol. 11, no. 2, pp. 106–124, April 2009.

[28] B. Donnet, B. Gueye, and M. A. Kaafar, "A Survey on Network Coordinates Systems, Design, and Security," IEEE Communication Surveys & Tutorials, vol. 12, no. 4, pp. 488– 503, October 2010.

[29] S. X. Wu and W. Banzhaf, "The use of computational intelli-gence in intrusion detection systems: A review," Applied Soft Computing, vol. 10, no. 1, pp. 1–35, January 2010.

[30] Y. Dong, S. Hsu, S. Rajput, and B. Wu, "Experimental Anal-ysis of Application Level Intrusion Detection Algorithms,"