Ensemble Network Intrusion Detection Model Based on Classification & Clustering for Dynamic Environment

Musyimi Samuel Muthama,

Prof. Waweru Mwangi
School of Computing & IT
Jomo Kenyatta University of Agriculture and Technology
(JKUAT)

Dr. Otieno Calvin.
School of Computing & IT
Jomo Kenyatta University of Agriculture and Technology
(JKUAT)

***Abstract* - Anomaly detection is a critical issue in Network Intrusion Detection Systems (NIDSs). Most anomaly based NIDSs employ supervised algorithms, whose performances highly depend on attack-free training data. However, this kind of training data is difficult to obtain in real world network environment. Moreover, with changing network environment or services, patterns of normal traffic will be changed. This leads to high false positive rate of supervised NIDSs. Unsupervised outlier detection can overcome the drawbacks of supervised anomaly detection. Therefore, study apply one of the efficient data mining algorithms called ensemble network intrusion detection model based on classification & clustering. Without attack-free training data, ensemble clustering algorithm can detect outliers in datasets of network traffic. In this paper, study discuss model of anomaly-based network intrusion detection. In machine learning, a combination of classifiers, known as an ensemble classifier, often outperforms individual ones. While many ensemble approaches exist, it remains, however, a difficult task to find a suitable ensemble configuration for a particular dataset. This paper proposed method includes an ensemble feature selecting classifier, data mining classifier. The former consists of four classifiers using different sets of features and each of them employs a machine learning algorithm named -bagging-randomization -boosting and -stacking. The latter applies data mining technique to automatically extract computer users' normal behavior from training network traffic data. The outputs of ensemble feature selecting classifier and data mining classifier are then fused together to get the final decision. The study proposes an ensemble-based that analysis of algorithm performance for intrusion detection. The method combines the output of four clustering methods to achieve an optimum selection. study then perform an extensive experimental evaluation of our proposed method using intrusion detection benchmark dataset, NSL-KDD.**

Keywords: - Artificial intelligence, Ensemble machine learning, Intrusion detection system, Intrusion Network security, Bagging, randomization, stacking, boosting.

I. BACK GROUND

With the tremendous growth of network-based services and sensitive information on networks, the number and the severity of network-based computer attacks have significantly increased. Although a wide range of security technologies such as information encryption, access control, and intrusion prevention can protect network-based systems, there are still many undetected intrusions. Thus, Intrusion Detection Systems (IDSs) play a vital role in network security. Network Intrusion Detection Systems (NIDSs) detect attacks by observing various network activities, while Host-based Intrusion Detection Systems (HIDSs) detect intrusions in an individual host.[1]

There are two major intrusion detection techniques: misuse detection and anomaly detection. Misuse detection discovers attacks based on the patterns extracted from known intrusions. Anomaly detection identifies attacks based on the deviations from the established profiles of normal activities. Activities that exceed thresholds of the deviations are detected as attacks. Misuse detection has low false positive rate, but cannot detect new types of attacks. Anomaly detection can detect unknown attacks, under a basic assumption that attacks deviate from normal behavior[2].

Due to extensive usages of internet, electronic assaults on network and information system of the financial organizations, military and energy sectors are increasing. Large web sites of any organization are attacked by various intruders and hackers [3]. Cyber security is the set of technologies and processes

designed to protect computers, networks, programs, and data from attack, unauthorized access, change, or destruction.[4].

Existing intrusion detection system approaches have high detection rate, whereas they suffer from high false-alarms. The task of reducing false positives is extremely necessary for intrusion detection system. Various Machine learning approaches have been used to implement intrusion detection system because it has the advantage of discovering useful knowledge from dataset. These approaches have ability to reduce the false positives. Bayes principle, Bayesian Belief Network, Hidden Markov Model, Artificial Neural Network, Genetic Algorithm, and Association of rules and clustering methods of machine learning are widely used to implement intrusion detection system. The combination of different base Machine learning algorithms is called as ensemble method. In literature survey, it is found that an Ensemble method of Machine learning helps to reduce false positive rates. There are four main methods to combine basic Machine learning classifiers and clustering. Bagging, Boosting, randomization and Stacking. In this paper, the Ensemble method of machine learning is proposed to implement intrusion detection system [3].

A. Statement of the Problem

Traditionally, network security has largely focused on identifying and preventing attacks, e.g., through attack signature generation or anomaly detection. However, the scale, complexity and diversity of large campus and enterprise networks render such an approach alone less efficient, scalable and manageable. Security incidents and evolving threats are on the rise and are increasing exponentially. Therefore, Intrusion detection is an important component of a modern information technology protection from unauthorized users**.** It detects and treat anomalies efficiently, because they affect the quality of services provided, resulting in degradation of network, performance and even in operations' interruption [5].

The task of uncovering new attacks in enterprise class networks quickly becomes unmanageable. Recently data mining methods have gained importance in addressing network security issues, including network intrusion detection a challenging task in network security. Classification-based data mining models for intrusion detection are often ineffective in dealing with dynamic changes in intrusion patterns and characteristics. The adversary methods are ever changing day night, the complexity and sophistication of attacks and vulnerability methods continues to rise yearly, and the potential impact to bottom line is significant organization information systems. And as Internet devices and applications continue to grow, it becomes increasingly important to understand network behavior for efficient network management and security monitoring.

Attacks have increased in frequency, size, variety, and complexity in recent years. The scope of threats has also changed into more complex schemes, including service and application-targeted attacks. Such attacks can cause far more serious disruptions than traditional brute force attempts and also require a more in-depth insight into IP services and applications for their detection. Through executing attack scenarios and applying abnormal behavior, the aim of this objective is to perform a diverse set of multistage attacks; each carefully crafted and aimed towards recent trends in security threats [6]. As the Internet usage is increasing significantly, security becomes more challenging problem. A network is secured only when it is provided by a software/hardware protection system with a strong monitoring, analyze and defense mechanisms. A class of these types of systems is named as Network Intrusion Detection Systems (NIDS). It is to monitor the dynamic behavior of intrusion from time to time and implement the defiance mechanisms within in a short span of time [7]

However, the existing detection methods still suffer from low True Negative Rate (TNR), accuracy, and precision. And their methods or models are homogeneous, so the robustness, stability, and universality are difficult to be guaranteed. To address the above-mentioned problems, this paper, we propose the Ensemble attack detection method based on hybrid heterogeneous metaclassifier ensemble learning [3].

The primary contributions of this paper:

i. To uncover percentage capability of existing intrusion detection systems and uncover new attacker pattern which compromise with Intranet performance.
ii. Visualizing the network traffic behavior normal or anomaly in dynamic environment.
iii. Analysis of Ensemble machine learning algorithm performance for intrusion detection and vote the appropriate, accurate predictive performance a single comprehensible structure

II. RELATED WORKS OF THE STUDY

Several authors have studied the ensemble classification and other classifications as a machine learning technique applied to, image processing pattern recognition and NIDS are summarized. Intrusion detection (ID) is the core element for network security. The main objective of ID is to identify abnormal behaviors and attempts caused by intruders in the network and computer system, and it is a big challenge to design and implement an intrusion detection system (IDS) meeting the objective Generally speaking, clustering techniques can be divided into two categories pairwise clustering and central clustering. The former, also called similarity-based clustering, groups similar data instances together based on a data-pairwise proximity measure [8].

Machine learning techniques deals with the construction and study of algorithms that can generalize (i.e. learn) from limited sets of data. Such algorithms operate by building models based on input and using those models to make predictions or decisions, rather than following only explicitly programmed instructions. Having such characteristics makes them ideal candidates for intrusion detection tasks [2]. Alternatively, it can be said that system based upon machine learning have ability to manipulate execution strategy based upon new inputs. Having such characteristics makes them ideal candidates for intrusion detection tasks Dadhich and Yadav [9], [2]. The machine learning has been successfully implemented in intrusion detection. Major machine learning techniques include the following:

Some IDS designers utilize ANN (Artificial Neural Network) as a Here, the NN learns to predict the behavior of the various users and daemons in the system. If properly designed and implemented, NN have the potential to address many of the problems encountered by rule-based approaches. The main advantage of NN is their tolerance to imprecise data and uncertain information and their ability to infer solutions from data without having prior knowledge of the regularities in the data. In order to apply this approach to ID, study would have to introduce data representing attacks and non-attacks to the NN to adjust automatically coefficients of this Network during the training phase. During training, the neural network parameters are optimized to associate outputs (each output represents a class of computer network connections, like normal and attack) with corresponding input patterns (every input pattern is represented by a feature vector extracted from the characteristics of the network connection record). When the neural network is used, it identifies the input pattern and tries to output the corresponding class. ANNs often suffer from local minima and thus long runtimes during learning [4].

Because the advanced versions of ANNs require even more processing power, they are implemented commonly on graphics processing units [10].

Classification and regression trees (CART) are machine-learning methods for constructing prediction models from data. These models are obtained through recursively partitioning the data and fitting a prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree. Classification trees are designed for variables that are dependent and that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values. The baseline will identify what is "normal" for that subject and alert when anomalous behavior is detected, or significantly different than the baseline. Main issue is the higher false positive rate [2].

Support vector machine Support Vector Machine: Support Vector Machine (SVM) is a supervised machine learning algorithm. It can be used for both classification and regression analysis. This algorithm plots each data item as a point in n-dimensional space (where n is number of features available) with the value of each feature being the value of a particular coordinate. Then, classification is performed by finding the hyper-plane that differentiates the two classes clearly. Main significance of the Support Vector Machines is that it is less susceptible for over fitting of the feature input from the input items, this is because SVM is independent of feature space. Here classification accuracy with SVM is quite impressive or high. SVM is fast accurate while training as well as during testing constructs decision trees from a set of available training data using the concept of information entropy. At each node of the tree, the algorithm selects the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurs on the smaller sub lists [4]

K-Medoids is clustering by partitioning algorithm as like as K-means algorithm. The most centrally situated instance in a cluster is considered as centroid in place of taking mean value of the objects in K-Means clustering. This centrally located object is called reference point. It minimizes the distance between centroid and data points which means minimizing the squared error. K-Medoids algorithm performs better than K-means algorithm when the number of data points increases. It is robust in presence of noise and outlier because medoid is less influenced by outliers, but processing is more expensive [11].

K-Nearest Neighbor (KNN): It is one of the simplest classification techniques. It calculates the distance between different data points on the input vectors and assigns the unlabeled data point to its nearest neighbor class. K is an important parameter. If k is equal to 1, then the data point is assigned to the class of its nearest neighbor. When value of K is large, then it takes large time for prediction and influence the accuracy by reduces the effect of noise

The k-Means algorithm groups 'n' instances into k disjoint clusters, where k is a predefined parameter. Each instance is assigned to its nearest cluster. For instance, assignment, measure the distance between centroid and each instance using Euclidean distance and according to minimum distance assign each and every data points into cluster. K –Means algorithm takes less execution time, when it is applied on small dataset. When the data point increases then it takes more execution time.

METHODOLOGIES

Human threats and attackers were classified. However, they need to be detected to prevent them. There are many approaches which use data mining algorithms to detect intrusions. Network based detection is one of the mechanism to accurately distinguish insider behavior from the normal behavior. Anomaly detection has become up-to-date topic because of the weakness of signature based IDSs in detecting novel or unknown attacks. Ensemble methodology are learning algorithms that construct a. set of classifiers and then classify new data points by taking a (weighted) vote of their predictions. The original ensemble method is Bayesian averaging.

But more recent algorithms include error-correcting output coding, Bagging, and boosting. Ensemble learning helps improve machine learning results by combining several models.

Ensemble methods are meta-algorithms that combine several machine learning techniques into one predictive model in order to decrease variance (bagging), bias (boosting), or improve predictions (stacking).

Unsupervised learning is known as descriptive or undirected classification. If there are data without the desired output, itis called unsupervised. The well-known unsupervised learning algorithms are clustering. Clustering can be categorized as an unsupervised learning approach, since we try to interpret and discover hidden structure in unlabeled data. On the other hand, the problem of classification is to predict the correct label for some input data. The classifier is learned using a set of training data containing feature vectors and their labels [12]

A. Ensemble Algorithm

Currently, Machine learning algorithms are widely used to implement intrusion detection system. Machine learning algorithm has the advantage of discovering useful knowledge from dataset. In literature survey, it is found that the Bagging Ensemble method of machine learning provides the high classification accuracy and low false positive rates. Aiming at constructing an intrusion detection approach with high classification accuracy, low false positives and low model building time, in this correspondence, study apply Bagging algorithm to intrusion detection system. In our Bagging based algorithm for intrusion detection, REPTrees are used as weak classifiers.

B. K-nearest neighbor design (KNN):

In pattern recognition, the k-Nearest Neighbors algorithm (or k-NN for short) is a non-parametric method used for Clustering and regression. In both cases, the input consists of the k closest training examples in the feature space. ... In k-NN Clustering, the output is a class membership[13].

The Euclidean distance Document Clustering

$$dist = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

Distance measured by Euclidean distance

$$sim(X, D_j) = \frac{\sum_{t_i \in (X \cap D_j)} x_i \times d_{ij}}{\|X\|_2 \times \|D_j\|_2}$$

X – Test document D_j – jth training document

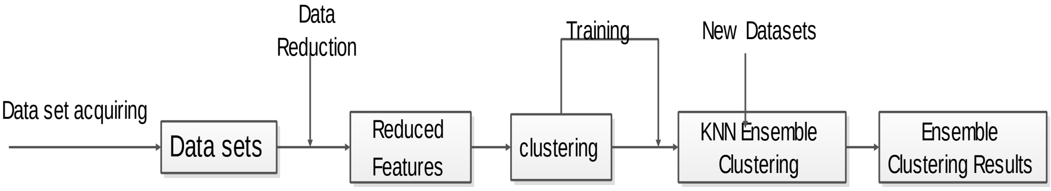
t_i – Word shared by X and D_j x_i – Weight of word t_i in X D_{ij} – Weight of word t_i in D_j.

C. The approach consists of three main phases (Training, testing and updating)

i. Training phase, the k-NN algorithm is used in order to establish a normal profile

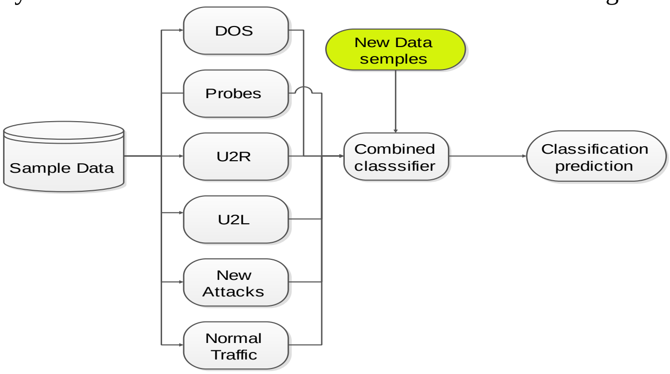
ii. Testing phase, check whether the current traffic of the node is normal or anomalous. If it is normal then update the normal profile otherwise isolate the malicious node and ignore that node from the network.

iii. To update the normal profile periodically, weighted coefficients and a forgetting equation is used[13],[14].



D. *Computer Attack classification*

Categorizes the attacks into five major types based on the goals and actions of the attacker. Proposed solution based on dynamic model Intelligent



Clustering of Test instances using Euclidean Network.

1 Input: Data Set, Class number K, CP. Output: Cluster Result.

Step1: K different objects are randomly selected as initial cluster centers.

Step2: Compute the similarity between object and class center by formula (6).

Step3: Divide each data object according to the nearest neighbor principle.

Step4: update the class center through the formula5

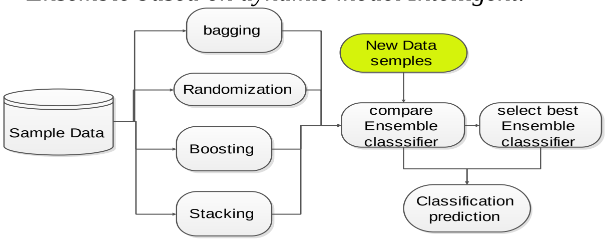
Step5: Repeat Step2, 3,and4, until the target function is no longer changed

Step6: evaluate the cluster precision, if meet the requirements, turn Step 8, otherwise, turn Step7.

Step7: Create a new clustering process into the next layer, then conduct step1 to 6.

Step8: Output the hierarchical clustering knowledge tree.

E. *Ensemble based on dynamic model Intelligent.*



Sample data: - This module monitors network Data and capture packets to serve for the data source of the NIDS.

Preprocessor: In preprocessing phase, network traffic collected and processed for use as input to the system.

Feature Extractor: This module extracts feature vector from the network packets (connection records) and submits the feature vector to the classifier module.

Classifier: The function of this module is to analyze the network stream and to draw a conclusion whether intrusion happens or not. Decision: When detecting that intrusion happens, this module will send a warning message to the user. Knowledgebase: This module serves for the training samples of the classifier phase. The intrusion samples can be perfected under user participation, so the capability of the detection can improve continually.

Description of Classification on network attackers

i. Denial of Service (DOS):- It is a type of attacks that denies a user to access a machine such as Smurf, Ping, Back, Mail bomb, UDP storm etc. In this attack, a hacker makes memory resources too busy to serve legitimate networking request.

ii. User to Root Attacks (U2R):-In this attacks, the hacker starts off on the system with the normal user account and mainly attempts to abuse vulnerabilities in the system in order to gain super user privileges. Eg. Xterm, Perl.

iii. Probing: - In this attack, a networking device or a machine is scanned by the hacker in order to determine vulnerabilities in the system that may be exploited later so as to compromise the system. eg. Postsweep, Nmap, Mscan, Satun, Saint etc.

iv. Remote to user Attacks (R2L):- This attacks deals with sending packets to a machine over the internet and the user does not have access to those packets. For eg. Xlock, Xnsnoop, Phf, Guest, Send mail dictionary etc.

v. New attackers:- Social engineering: - Gaining unauthorized access to a system or network by Subverting personnel and Brute force attack: Attempt to “crack” passwords by sequentially trying all possible Combinations characters [15][4].

CLUSTERING OF ATTACKS ON KDD DATASET

	Types of attacks	Clustering rules of Attacks
1	DoS	smurf, land, pod, teardrop, Neptune, back, Satan
2	R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient
3	U2R	perl, buffer_overflow, rootkit, loadmodule, perl
4	Probe	ipsweep, nmap, satan, portsweep, ipsweep, port scanning
5	New Attacker	portsweep, syn flood, Brute force attack, social engineering, password lock, Key-lockers, Trojan horse, machine generated malware.

F. Feature selection

Feature selection Before training, the step of feature (or variable) selection may be considered. The process of feature selection identifies which features are more discriminative than the others. This has the benefit of generally improving system performance by eliminating irrelevant and redundant features. Table 6 shows year wise distribution of feature selection considered in related work. This result reveals that not all studies perform feature selection before classifier training. In particular, 26 experiments considered feature selection. On the other hand, 30 experiments do not perform feature selection. In total, feature selection is not very popular procedure in intrusion detection. However, use different feature selection methods for their experiments. This implies that feature selection could improve some certain level of Clustering accuracy in intrusion detection [4].

G. Computational Intelligence Techniques

Network traffic data is usually associated with large volume and having numerous fields that require careful examination by IDS. To alleviate the overhead problem, feature selection was performed prior to classification. Besides, selecting the significant features which signify each traffic class is to find the intrusive patterns or common properties are which often hidden within the irrelevant features. They further commented that there are features that contain false correlation. Some of these features also may be redundant and may have different discriminative power. Therefore, the aim of feature selection is to disclose these hidden significant features from the irrelevant features. Thus, an accurate and fast classification can be achieved [8].

H. Ensemble Intelligence for Classification

The effectiveness of an ensemble or multiple classifier approach also depends on the choice of the decision fusion function. To determine the decision function, the expected degree of diversity among clusters should be taken into account. Here, ensemble machine learning techniques with different learning paradigms were used to classify the network connection. Decision function was determined based on the individual performances on overall accuracy and true positive rates [12].

III. RESULTS AND DISCUSSION (EXPERIMENTAL SETUP)

Ensemble model on dynamic Environment Intelligent

The study proposes Anomaly-based techniques model the normal network and system behavior, and identify anomalies as deviations from normal behavior. They are appealing because of their ability to detect zero-day attacks. Another advantage is that the profiles of normal activity are customized for every system, application, or network, thereby making it difficult for attackers to know which activities they can carry out undetected. Additionally, the data on which anomaly-based techniques alert (novel attacks) can be used to define the signatures for misuse detectors. The study has proposed a method, that uses K-NN clustering and five clusters are formed, four clusters for the four different types of attacks and one for normal traffic. Then the distance is calculated between data samples and each cluster center. Anomaly-based Intrusion Detection works assuming that the attacks are different to normal activity, you can reach this inference after a training process, which will be identified, "what is considered normal activity?", analyzing unusual behavior in both host and network traffic which is Machine learning.

Ensembles a brief overview Ensemble is unsupervised learning technique, which is a combination of learning algorithms. Ensemble is the process of utilizing multiple algorithms to obtain better predictive performance compared to the usage of single learning techniques. Hence, they are not bound by the number or type of the individual components being used. Machine learning ensembles, in contrast to statistical ensembles utilizes finite models for building classifiers, however, they allow flexible structures to exist in the mechanism [4].

A. Ensemble Variants and Applicability Levels: A Discussion

Some commonly used ensembles include Bagging, Boosting, randomization and Stacking. An analysis of ensembles and their varied flavors in terms of performance measures are available in literature. However, ensembles, being a fairly new modelling technique, has not been examined in terms of the nature of data that they are being applied on.

Bootstrap Aggregating or Bagging is a machine learning ensemble model with its major focus on increasing the stability and accuracy of the machine learning models. It is referred to as the model averaging approach. Though it is usually applied on tree-based models, it can support heterogeneity. Heterogeneous multi model based bootstrap techniques have not yet been proposed. Random Forest is one of the most well-known bagging techniques. Bagging operates by effectively sampling data and training multiple classifiers on the subsets. The training models are usually multiple instances of the same classifier. Consider m classifier instances and a training set of size n. Bagging generates m new training sets each of size a, where (a < n.) However, it is maintained that the size of a is usually (1-1/e) or ~63.2% of the unique examples in the training data.

Sampling is performed with replacement; hence duplicates can be expected in the training data. Voting is used as the final combination technique. The major advantage of bagging is that it provides improvements for unstable procedures such as ANN, classification and regression trees. Since only a part of the data is used for training individual models, imbalance can be counteracted, as some trees might receive balanced data with equal minority and majority class levels, others might receive minority class data alone and most others a part of majority and minority classes in several ratios. Hence this leads to a mixed training, combination of which can provide an enhanced training model. However, scalability of such a system is in question. Increase in data size leads to increased training data on the ensemble components. Since multiple such components are created, computational complexity increases exponentially, leading to scalability issues when used on huge datasets [3]

Boosting is an ensemble learning technique primarily focused on reducing bias and variance in supervised learning techniques. It operates on the basis that several weak learners can be effectively combined to generate a strong learner. A weak learner has slight correlation with the true classification, better than random guessing, while a strong learner has high correlation with the true classifications. Boosting operates by iteratively training weak classifiers on a single data distribution and hence building. The strong classifier based on the combination of rules generated by the weak classifier. Boosting operates by initially fitting a model f(x) to the data. Being a supervised approach, the model is then reiterated and backtracked to identify the errors. Unlike bagging, boosting reiterates through a single model, hence scalability issues are reduced extensively. Further, due to reiterated training and error handling mechanism, it is believed that boosting can handle very high imbalance levels, provided sufficient data is given for training.

Stacking is an enhanced extension of bucket of models, in the sense that it supports heterogeneous models in the formation of ensemble [16]. However, unlike its counterpart, stacking requires a combiner algorithm that combines the results of individual models to provide a model that performs better than any of the individual models. The combiner algorithm is a heuristic that effectively operates on the results from individual models. A single layer logistic regression is usually used as a combiner; however, combiner is problem specific and can be effectively used to finetune the result sets to obtain results suiting to the problem domain. The major advantage of this approach is that it utilizes several models, hence can provide the best component of all the available models. Although scalability might be an issue, the improved performance levels and heterogeneity incorporation would provide a huge tradeoff in terms of accuracy [3]. Bucket of models is an ensemble modelling mechanism that operates on a variety of algorithms to provide the best algorithm based on the training data. Hence the bucket of models can produce results that is the best among available algorithms. When operated upon with a single algorithm, this technique provides the best among available results. However, while operated using several algorithms, due to the diverse nature, results obtained would be much better than using single techniques [3]. Creating an ensemble provides the flexibility to use any type of data on the model, rather than the training data that was used to create the trained model. The issue of imbalance and data hugeness will be handled by the best algorithm that can most effectively handle such issues.

B. Total interaction capture

The amount of information available to detection mechanisms are of vital importance as this provides the means to detect anomalous behavior. In other words, this information is essential for post-evaluation and the correct interpretation of the results. Thus, it is deemed a major requirement for a dataset to include all network interactions, either within or between internal LANs.

C. Complete capture

Privacy concerns related to sharing real network traces has been one of the major obstacles for network security researchers as data providers are often reluctant to share such information. Consequently, most such traces are either used internally, which limits other researchers from accurately evaluating and comparing their systems, or are heavily anonymized with the payload entirely removed resulting in decreased utility to researchers. In this work, the foremost objective is to generate network traces in a controlled testbed environment, thus completely removing the need for any sanitization and thereby preserving the naturalness of the resulting dataset.

D. Data source KDD Data Set (NSL-KDD dataset)

Using our system, we ran millions of experiments using the NSL-KDD dataset which is a secondary data, unlabeled dataset that attempts to mimic real network traffic. We then compared the results from different configurations and identified trends which provided insight into how to best perform intrusion detection with unsupervised outlier detection ensembles [12].The competition task was to build a network intrusion detector, a predictive model capable of distinguishing between ``bad'' connections, called intrusions or attacks, and ``good'' normal connections. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment [17]. The number of records in the train and test sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works will be consistent and comparable [3].

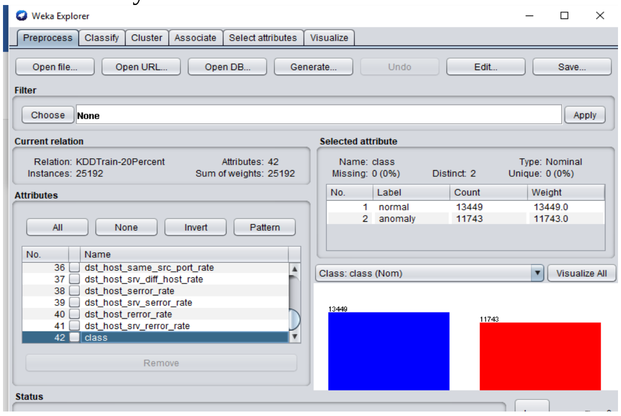
E. Discussion of results

Ensemble learning finishes the learning task by structuring and combining multiple individual classifiers. It is homogeneous for the ensemble of the same type of individual classifiers, and this kind of individual classifier is known as “base classifier” or “weak classifier.” Ensemble learning can also contain the different types of individual classifiers, and the ensemble is heterogeneous. In heterogeneous ensemble, the individual classifiers are generated by different learning algorithms. The classifiers are called as “component classifier.” For the research of homogeneous base classifier, there is a key hypothesis that the errors of base classifier are independent of each other. However, for the actual attack traffic detection, they apparently are impossible. In addition, the accuracy and the diversity of individual classifiers conflict in nature. When the accuracy is very high, increasing the diversity becomes extremely difficult. Therefore, to generate the robust generalization ability, the individual classifiers ought to be excellent and different.

F. *Network intrusion detection*

NIDS tries to discover the unauthorized access to a computer network. And as machine learning has proved its power in other fields, it can also be used for detecting malicious activities on a network as long as we have enough data to make a machine learn. Let's start the process.

Results analysis



Filter: None

Relation: KDDTrain-20Percent   Attributes: 42
Instances: 25192   Sum of weights: 25192

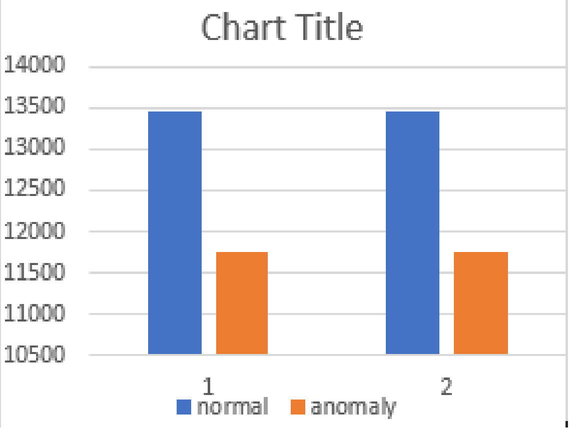
Name: class   Type: Nominal
Missing: 0 (0%)   Distinct: 2   Unique: 0 (0%)

No.	Label	Count	Weight
1	normal	13449	13449.0
2	anomaly	11743	11743.0

No.		Name
36	□	dst_host_same_src_port_rate
37	□	dst_host_srv_diff_host_rate
38	□	dst_host_serror_rate
39	□	dst_host_srv_serror_rate
40	□	dst_host_rerror_rate
41	□	dst_host_srv_rerror_rate
42	□	class

Class: class (Nom)

Classifiers normal and anomaly



Bagging classifier sampling the training set

```
=== Run information ===
Scheme:       weka.classifiers.meta.Bagging -P 100 -S 1 -
num-slots 1 -I 10 -W weka.classifiers.bayes.NaiveBayes
Relation:     KDDTrain-20Percent
Instances:    25192
Attributes:   42
Test mode:    evaluate on training data
=== Classifier model (full training set)
Bagging with 10 iterations and base learner
weka.classifiers.bayes.NaiveBayes
Time taken to build model: 2.19 seconds
=== Evaluation on training set ===
Time taken to test model on training data: 6.24 seconds
=== Summary ===
Correctly Classified Instances      22584   89.6475 %
Incorrectly Classified Instances     2608      10.3525 %
Kappa statistic                 0.7917
Mean absolute error               0.1025
Root mean squared error             0.3079
Relative absolute error           20.5872 %
Root relative squared error         61.7186 %
Total Number of Instances          25192
=== Detailed Accuracy By Class ===
             TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC
Area  PRC Area  Class
             0.912    0.122    0.896      0.912    0.904      0.792    0.969
0.971    normal
             0.878    0.088    0.897      0.878    0.888      0.792    0.965
0.952    anomaly
Weighted Avg.    0.896    0.106    0.896      0.896    0.896      0.792    0.967
0.962
=== Confusion Matrix ===
    a     b   <-- classified as
 12271  1178 |     a = normal
  1430 10313 |     b = anomaly
```

Acc= (a+d)/(a+b+c+d)=(22584)/(25192)= 89.6%
pf=c/(a+c)=/(12271+1430)= 10.4%.....4.6

Bagging classifier supplier test dataset

```
=== Detailed Accuracy By Class ===
             TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
 0.997    0.005    0.995      0.997    0.996      0.992    0.999
0.999    normal
0.995    0.003    0.997      0.995    0.996      0.992    0.999
0.999    anomaly
Weighted Avg.    0.996    0.004    0.996      0.996    0.996
0.992    0.999    0.999
=== Confusion Matrix ===
    a     b   <-- classified as
 13413    36 |     a = normal
    63  11680 |     b = anomaly
```

Acc= (a+d)/(a+b+c+d)=(25093)/(25192)= 99.61
pf=c/(a+c)=63/(13476)= 0.47%.....4.6

Random Forest training data set

```
=== Detailed Accuracy By Class ===
             TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
             0.999    0.022    0.981      0.999    0.990      0.978
1.000    1.000    normal
             0.978    0.001    0.999      0.978    0.988      0.978
1.000    1.000    anomaly
Weighted Avg.    0.989    0.013    0.989      0.989    0.989
0.978    1.000    1.000
=== Confusion Matrix ===
    a     b   <-- classified as
 13434    15 |     a = normal
   264 11479 |     b = anomaly
```

Acc= (a+d)/(a+b+c+d)=(24913)/(25192)= 98.90
pf=c/(a+c)=264/(13434+264)= 1.9%.....4.6

RandomForest supplied test dataset: size unknown (reading incrementally)

```
=== Detailed Accuracy By Class ===
             TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
             0.972    0.361    0.670      0.972    0.793      0.624
0.957    0.954    normal
             0.639    0.028    0.968      0.639    0.769      0.624
0.957    0.964    anomaly
```

```
Weighted Avg.    0.782    0.172    0.840     0.782    0.780
0.624    0.957    0.960
=== Confusion Matrix ===
   a    b   <-- classified as
 9436  275 |    a = normal
 4639 8194 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(17630)/( 22544)= 78.20

      pf=c/(a+c)=4639/(9436+4639)= 32.96%.....4.6
```

Boosting training data set

```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 0.935    0.042    0.962      0.935    0.948      0.891
0.987    0.987     normal
                 0.958    0.065    0.928      0.958    0.943      0.891
0.987    0.986     anomaly
Weighted Avg.    0.946    0.053    0.946     0.946    0.946
0.891    0.987    0.986
=== Confusion Matrix ===
     a     b   <-- classified as
 12576   873 |    a = normal
   497 11246 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(23822)/( 25192)= 94.56
      pf=c/(a+c)=497/(12576+497)= 3.8%.....4.6
```

Boosting supplied test dataset

```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 0.651    0.496    0.226      0.651    0.335      0.119
0.670    0.402     normal
                 0.504    0.349    0.867      0.504    0.638      0.119
0.670    0.883     anomaly
Weighted Avg.    0.531    0.376    0.750     0.531    0.583
0.119    0.670    0.796

=== Confusion Matrix ===
   a    b   <-- classified as
 1400  752 |    a = normal
 4808 4890 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(6290)/( 11850)= 53.08
      pf=c/(a+c)=4808/(1400+4808)= 77.45%.....4.6
```

Stacking classifiers training data set

```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.534      1.000    0.696      0.000
0.500    0.534     normal
                 0.000    0.000    0.000      0.000    0.000      0.000
0.500    0.466     anomaly
Weighted Avg.    0.534    0.534    0.285     0.534    0.372
0.000    0.500    0.502
=== Confusion Matrix ===
     a     b   <-- classified as
 13449     0 |    a = normal
 11743     0 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(13449)/( 25192)= 53.39%
      pf=c/(a+c)=13449/(13449+11743)= 53.39%%.....4.6
```

Stacking Classifiers supplied Test data sets

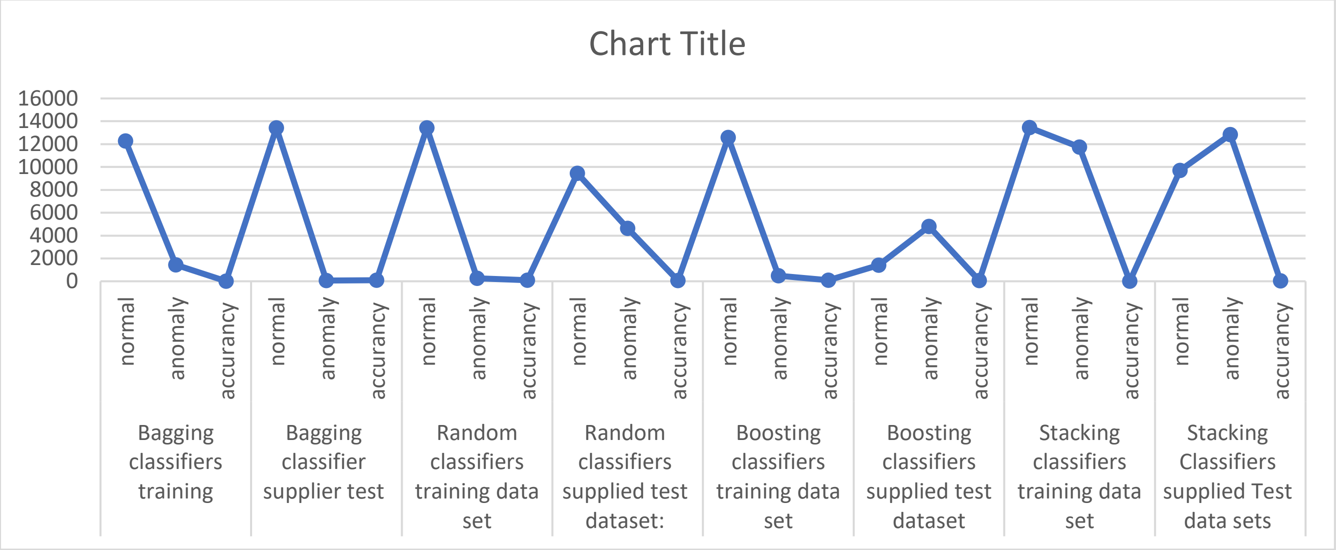
```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.431      1.000    0.602      0.000
0.500    0.431     normal
                 0.000    0.000    0.000      0.000    0.000      0.000
0.500    0.569     anomaly
Weighted Avg.    0.431    0.431    0.186     0.431    0.259
0.000    0.500    0.510
=== Confusion Matrix ===
     a     b   <-- classified as
  9711     0 |    a = normal
 12833     0 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(9711)/(22544)= 43.08
      pf=c/(a+c)=12833/(9711+12833)= 56.92%.....4.6.
```

Random Forest training data set

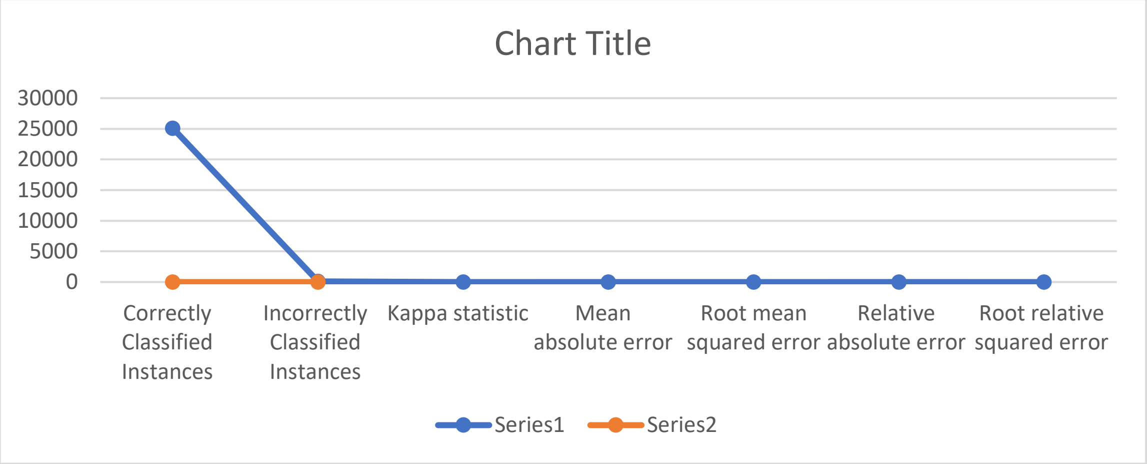
```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 0.999    0.022    0.981      0.999    0.990      0.978
1.000    1.000     normal
                 0.978    0.001    0.999      0.978    0.988      0.978
1.000    1.000     anomaly
Weighted Avg.    0.989    0.013    0.989     0.989    0.989
0.978    1.000    1.000
=== Confusion Matrix ===
     a     b   <-- classified as
 13434    15 |    a = normal
   264 11479 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(24913)/( 25192)= 98.89
      pf=c/(a+c)=264/(13434+264)= 1.93%.....4.6
```

Random Forest supplied test data set

```
=== Detailed Accuracy By Class ===
                 TP Rate  FP Rate  Precision  Recall   F-Measure
MCC      ROC Area  PRC Area  Class
                 0.972    0.361    0.670      0.972    0.793      0.624
0.957    0.954     normal
                 0.639    0.028    0.968      0.639    0.769      0.624
0.957    0.964     anomaly
Weighted Avg.    0.782    0.172    0.840     0.782    0.780
0.624    0.957    0.960
=== Confusion Matrix ===
   a    b   <-- classified as
 9436  275 |    a = normal
 4639 8194 |    b = anomaly
 Acc= (a+d)/(a+b+c+d)=(17630)/( 22544)= 78.2
      pf=c/(a+c)=4639/(9436+4639)=32.97 %.....4.6
```

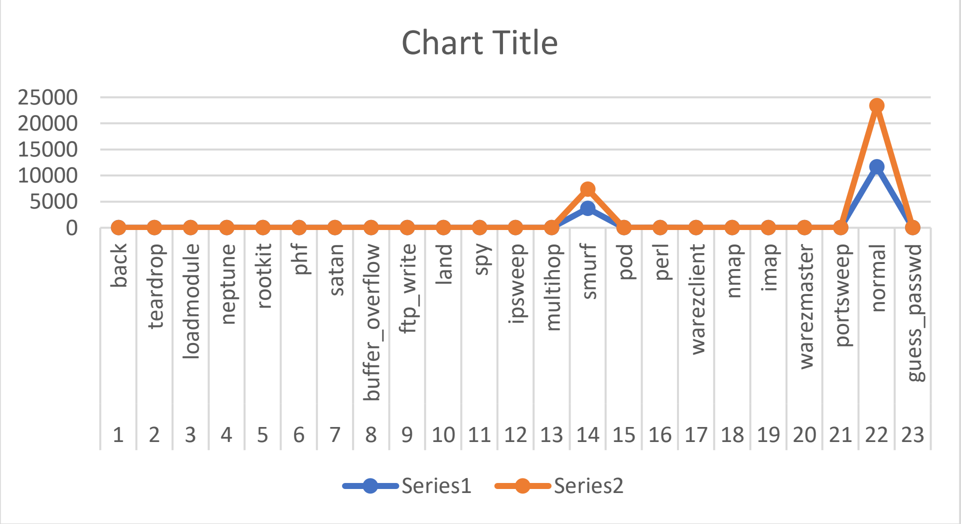


Voting of different classifier using Ensemble committee
Bagging classifiers training machine learning qualify to be more accurately as compared to the other three classifiers.



Classifier Ensembles. We constructed several anomaly IDS by combining multiple classifiers using the simple majority voting rule. In this work, the ensemble Bagging method of machine learning for intrusion detection Bagging with clustering base accuracy, model building system is presented.

The classifier is proposed for detection of anomaly packet over network. The proposed method is evaluated on test dataset and cross validation of 10-fold. The performances of classifies are measured in terms of classification time and false positives.



Bagging classifiers training	normal	12271	1178
	anomaly	1430	10313
	accurancy	89.60%	89.60%
Bagging classifier supplier test	normal	13413	36
	anomaly	63	11680
	accurancy	99.61	95

Confuse matrix machine learning using Bagging classifiers training

Line graphical showing Analysis of algorithm performance for intrusion detection Bagging classifiers training.

5 CONCLUSION AND FUTURE DIRECTIONS

As faster and more effective countermeasures are required to cope with the ever-growing number of network attacks, AI comes as a natural solution to the problem. Though briefly, this paper has reviewed various intrusion detection systems (IDS) and their classification based on various modules. A comprehensive review of various AI based techniques used in intrusion detection (ID) is presented. A multi classifier-based technique (ensemble approach) is discussed that results into detection of known and unknown attacks with high accuracy. Various studies of artificial intelligence (AI) based techniques in ID are compared by considering many parameters like source of audit data, processing criteria, technique used, classifier design, dataset, feature reduction technique employed and classification classes. It can be observed that by considering appropriate base classification techniques, training sample size &c combinations method, detection accuracy of hybrid and/or ensemble approach can be improved. But ensemble approach has increased the computational overhead. In future, there is acute need to research following issues related to AI based techniques in ID.

REFERENCE

1. Parvania, M., et al. Hybrid Control Network Intrusion Detection Systems for Automated Power Distribution Systems. in 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. 2014. IEEE.
2. Stampar, M. and K. Fertalj. Artificial intelligence in network intrusion detection. in Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on. 2015.
3. Gaikwad, D.P. and R.C. Thool. Intrusion Detection System Using Bagging Ensemble Method of Machine Learning. in 2015 International Conference on Computing Communication Control and Automation. 2015.
4. Buczak, A.L. and E. Guven, A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 2016. **18**(2): p. 1153-1176.

Prevention of intrusion is highly accurate about 99.61 % in Test dataset.

5. Zhao, Y. Network intrusion detection system model based on data mining. in 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). 2016.
6. Singh, S., et al. Intrusion Detection Based On Artificial Intelligence Techniques. in International Conference Of Advance Research And Innovation (Icari-2014). 2014.
7. He, D., et al., Software-De&#xFB01;ned-Networking-Enabled Traf&#xFB01;c Anomaly Detection and Mitigation. IEEE Internet of Things Journal, 2017. **PP**(99): p. 1-1.
8. Kwon, D., et al., A survey of deep learning-based network anomaly detection. Cluster Computing, 2017.
9. Dadhich, A. and S.K. Yadav, Evolutionary Algorithms, Fuzzy Logic and Artificial Immune Systems applied to Cryptography and Cryptanalysis: State-of-the-art review. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2014. **3**(6).
10. Norouzian, M.R. and S. Merati. Classifying attacks in a network intrusion detection system based on artificial neural networks. in 13th International Conference on Advanced Communication Technology (ICACT2011). 2011.
11. Tang, T.A., et al. Deep learning approach for Network Intrusion Detection in Software Defined Networking. in 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM). 2016.
12. Ruoti, S., et al. Intrusion Detection with Unsupervised Heterogeneous Ensembles Using Cluster-Based Normalization. in 2017 IEEE International Conference on Web Services (ICWS). 2017.
13. Varuna, S. and P. Natesan. An integration of k-means clustering and na&#x00EF;ve bayes classifier for Intrusion Detection. in Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on. 2015.
14. Shirbhate, S., S. Sherekar, and V. Thakare. A Novel Framework of Dynamic Learning Based Intrusion Detection Approach in MANET. in Computing Communication Control and Automation (ICCUBEA), 2015 International Conference on. 2015. IEEE.
15. Idris, N.B. and B. Shanmugam. Artificial Intelligence Techniques Applied to Intrusion Detection. in 2005 Annual IEEE India Conference - Indicon. 2005.
16. Al-Jarrah, O. and A. Arafat. Network Intrusion Detection System using attack behavior classification. in 2014 5th International Conference on Information and Communication Systems (ICICS). 2014.
17. Farid, D.M. and M.Z. Rahman, Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm. JCP, 2010. **5**(1): p. 23-31.