A Hybridized SVM-kNN-pdAPSO Approach to Intrusion Detection System
E.G. Dada
Department of Computer Engineering, Faculty of Engineering, University of Maiduguri, Maiduguri, Borno State, Nigeria.
gbengadada@unimaid.edu.ng; +2349084222298

Abstract
Several cases of service attacks on major Internet sites have shown us, no open computer network is immune from intrusions. The wireless ad-hoc network is particularly vulnerable due to its features of open medium, dynamic changing topology, cooperative algorithms, lack of centralized monitoring and management point, and lack of a clear line of defense. The traditional way of protecting networks with firewalls and encryption software is no longer sufficient and effective. The field of machine learning has proved that the hybridization of classifiers usually has a better performance than individual ones. This paper proposes a new approach that hybridizes different classifiers for better accuracy in detection of intrusion attacks. The result of experiments conducted shows that the fusion of Support Vector Machine, k-Nearest Neighbor, and Primal-Dual Particle Swarm Optimization produced better classification accuracy than each of the singular classifiers on the KDD99 dataset.

Keywords: Intrusion Detection System, k-Nearest neighbor, Support Vector Machine, Primal Dual, Particle Swarm Optimization

1.0 Introduction
Intrusion detection is a very important topic of network security that has received much attention according to Patcha and Park (2007). The threats posed to computer users by computer related malicious code or intrusion attacks is on the increase every day. According to Lee, Stolfo, and Mok (1999), KDD data set can be used for evaluating a comprehensive set of pattern recognition and machine learning. Hybridized methods were proved to experimentally and theoretically possess better accuracy than using any one classifier algorithm alone. The success of the hybridized method for intrusion classification greatly depends on the diversity in the outputs of its different classifiers, as well as on the choice of method to combine these outputs into a single one (Chen and Wong, 2014).

Over the past twenty years, many research has been done much to advance the state-of-the-art in this increasingly important area. According to Curtis and Carver (2000), research prototypes and commercial IDS and IRS built during this period now number nearly 100. Jianxiao and Lijuan (2008) presented an Intrusion Detection technique that combines static agent and mobile agent, Host-based IDS and Network-based IDS and a distributed intrusion detection system model based on agents. A new pattern matching algorithm is proposed by Zhang (2009), first sequence letters in the pattern string from low appearance probability to high appearance probability in natural English, and then match one by one according to the algorithm thereby reducing the comparison times. Syarif et al. (2012) used naïve Bayes, J48 (decision tree), JRip (rule induction) and iBK (nearest neighbor) to solve the intrusion detection problem by improving the accuracy and reduce its false positive
Dada., A Hybridized SVM-kNN-pdAPSO Approach to Intrusion Detection System

rates. Bahri et al. (2011) proposed a hybrid technique called Greedy-Boost using the KDD 1999 data set. Bukhtoyarov et al. (2014) proposed a probabilistic method to design base neural network classifiers known as probability-based generator of neural networks structures (PGNS) to the network intrusion detection problem. Cordeiro and Pappa (2011) proposed the PSO-WV technique that used the PSO algorithm for weighting classifications coming from different data views.

In this work, Six k-nearest neighbor (k-NN) were trained and six support vector machine (SVM) on the same dataset. The output of the training is feed into the pdAPSO algorithm. The final decision is reached by combining the opinions of the different SVM-kNN-pdAPSO hybrid approach.

2.0 Methodology
2.1 System framework
In line with the objectives stated earlier, the SVM, k-NN, and pdAPSO were used in training and testing the system. Fig. 1 illustrates the entire process. For ease of comprehension, the system framework was divided into the following four phases: Kdd99 data pre-processing, data classification using SVM algorithm, data classification using k-NN algorithm, and data classification using pdAPSO algorithm.

![Fig. 1: System Framework](image)

2.2. Dataset used for experiments
For the purpose of this research work, the knowledge discovery and data mining 1999 (KDD99) dataset was used for the experiments. The dataset contains hundreds of thousands of connection records. KDD dataset covers four major categories of attacks: Probing attacks (information gathering attacks), Denial-of-Service (DoS) attacks (deny legitimate requests to a system), user-to-root (U2R) attacks (unauthorized access to local super-user or root), and remote-to-local (R2L) attacks (unauthorized local access from a remote machine). Elkan (2000) described the attributes in the KDD datasets that they have all forms - continuous, discrete, and symbolic, with significantly varying resolution and ranges.

Experiments was conducted for this study using training, validation and testing samples were drawn from five datasets downloaded from the KDD 99 dataset. In order to compare efficiency of proposed algorithms, we adopt the accuracy of classification as the yardstick for measuring the performance of the system. Where accuracy of classification is the
A Hybridized SVM-kNN-pdAPSO Approach to Intrusion Detection System

proportion of the test data correctly classified by an algorithm. A classifier’s accuracy is the number of correctly classified observations \( C_i \) of class \( i \) divided by the total number of observations \( s \). As depicted in Fig. 2, each of the algorithms consists of five binary classifiers. Each binary classifier \( B_i \) is used to classify instances of class \( i \) from the set of observations \( O = O_1, O_2, ..., O_s \). Therefore, we determine accuracy \( A_i \) of each binary classifier according to Eq. (1).

\[
A_i = \frac{C_i}{s}
\]  

(1)

All the data were gotten from (KDD99) dataset, and can be categorized into three sets: Training data, testing data, and validation data. The KDD99 dataset is divided into normal traffic and four classes of attacks (Araujo et al., 2010). Table 1 provides information about the number of observations in training and testing datasets for each class. This experiment used five data sets, taken from the training and testing KDD99. As shown in Table 2, all five samples are of the same size, but contain different observations selected at random from the training and testing datasets represented in Table 1.

### Table 1: Number of observations in KDD99.

<table>
<thead>
<tr>
<th>Connection type</th>
<th>Training dataset</th>
<th>Testing dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>97,278</td>
<td>60,593</td>
</tr>
<tr>
<td>DoS</td>
<td>391,458</td>
<td>229,853</td>
</tr>
<tr>
<td>Probe</td>
<td>4107</td>
<td>4166</td>
</tr>
<tr>
<td>R2L</td>
<td>1126</td>
<td>6,347</td>
</tr>
<tr>
<td>U2R</td>
<td>52</td>
<td>70</td>
</tr>
<tr>
<td>Total</td>
<td>494,021</td>
<td>311,029</td>
</tr>
</tbody>
</table>

### Table 2: Dataset size used in experiments.

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>DoS</th>
<th>Probe</th>
<th>R2L</th>
<th>U2R</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>5000</td>
<td>4107</td>
<td>5000</td>
<td>52</td>
<td>1126</td>
<td>15,285</td>
</tr>
<tr>
<td>Testing</td>
<td>7500</td>
<td>3124</td>
<td>7500</td>
<td>52</td>
<td>3750</td>
<td>21,926</td>
</tr>
<tr>
<td>Validation</td>
<td>2500</td>
<td>1042</td>
<td>2500</td>
<td>18</td>
<td>1250</td>
<td>7310</td>
</tr>
</tbody>
</table>

2.3 Data Pre-processing

Data in the real world is in their raw state. This means that they are incomplete (lacking attribute values, lacking certain attributes of interest, or containing only aggregate data). KDD99 data includes three symbolic features that are incompatible with the proposed classification algorithms such as protocol type, service, and flag. Pre-processing operation was performed on the data in two steps: Data mapping and identification of State. This means that each record in the data belongs to one of five major classes: Normal, DoS, Probe, U2R, and R2L. The values for each state are mapped to a numeric value.

2.4 SVM classifier

The support vector machine (SVM) is a supervised learning method that generates input-output mapping functions from a set of labeled training data. The mapping function can be either a classification function, i.e., the category of the input data, or a regression
function. For classification, nonlinear kernel functions are often used to transform input data to a high-dimensional feature space in which the input data become more separable compared to the original input space. Maximum-margin hyperplanes are then created. The model thus produced depends on only a subset of the training data near the class boundaries. SVM is originally an implementation of Vapnik’s Structural Risk Minimization (SRM) principle (Kuang et al., 2014), which is known to have low generalization error or equivalently does not suffer much from overfitting to the training data set.

2.5 k-NN Classifier
The K-nearest neighbour algorithm is a (k-NN) is a simple and effective method for classifying objects based on closest training examples in the feature space. K-Nearest Neighbors (KNN) classification divides data into a test set and a training set. If there are ties for the Kth nearest vector, all candidates are included in the vote. Consider a set of observations and targets \((x_1, y_1), \ldots, (x_n, y_n)\), where observations \(x_i \in \mathbb{R}^d\) and targets \(y_i \in \{0, 1\}\); then for a given \(i\), k-NN rates the neighbours of a test sequence among the training sample, and uses the class labels of the nearest neighbours to predict the test vector class. In k-NN, the Euclidean distance is often used as the distance metric to measure the similarity between two vectors (points):

\[
d^2(x_i, x_j) = \left\| x_i - x_j \right\|^2 = \sum_{k=1}^{d} (x_{ik} - x_{jk})^2
\]

where \((x_i, y_j) \in \mathbb{R}^d\), \(x_i = (x_{i1}, x_{i2}, \ldots, x_{id})\)

2.6 Primal-Dual Asynchronous Particle Swarm Optimisation (pdAPSO) Algorithm
The primal dual asynchronous PSO (pdAPSO) works by first allowing the primal dual section of the algorithm to optimize the particles that have been randomly introduced into the search space. The asynchronous PSO then accept the result of the primal dual algorithm and optimize it to get the optimal solution. The asynchronous PSO computes the personal best (pbest\(_{i,m}\)) or the global best (gbest\(_{i,m}\)) of a particle. The velocity and position of the particles are updated immediately after computing their fitness. The Primal Dual provides a better balance between exploration and exploitation, preventing the particles from experiencing premature convergence and been trapped in local minima easily. For detail information on the Primal Dual APSO the reader is referred to the previous work Dada (2016).

3.0 Results and Discussion
Experiments were conducted, and the pdAPSO was used to optimize the selected dataset that was chosen for the purpose of our experiment. From the result of our experiment, the pdAPSO converged optimally for each of the dataset that represent the different attack categories that we are considering. Examples of datasets used include normal, DoS, probe, R2L, and U2R. There was no case of premature convergence or particles of the pdAPSO been trapped in the local optimal. Depicted in Fig. 3 is the graph of convergence of pdAPSO algorithm towards optimal solution, for each of our five categories of attacked considered in this study.
Dada., A Hybridized SVM-kNN-pdAPSO Approach to Intrusion Detection System

Fig. 3. Convergence of pdAPSO

Table 3: Mean result of 5 datasets

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Normal</th>
<th>DoS</th>
<th>Probe</th>
<th>R2L</th>
<th>U2R</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM 1</td>
<td>76.212%</td>
<td>96.299%</td>
<td>93.9912%</td>
<td>98.5825%</td>
<td>84.5462%</td>
</tr>
<tr>
<td>SVM 2</td>
<td>76.518%</td>
<td>95.229%</td>
<td>96.9701%</td>
<td>99.4846%</td>
<td>83.8420%</td>
</tr>
<tr>
<td>SVM 3</td>
<td>79.731%</td>
<td>94.792%</td>
<td>98.0965%</td>
<td>99.6142%</td>
<td>83.7937%</td>
</tr>
<tr>
<td>SVM 4</td>
<td>76.972%</td>
<td>94.401%</td>
<td>97.9521%</td>
<td>99.6789%</td>
<td>83.4562%</td>
</tr>
<tr>
<td>SVM 5</td>
<td>73.359%</td>
<td>93.654%</td>
<td>95.2101%</td>
<td>99.7419%</td>
<td>83.4115%</td>
</tr>
<tr>
<td>SVM 6</td>
<td>70.401%</td>
<td>92.311%</td>
<td>94.0982%</td>
<td>99.7528%</td>
<td>83.1743%</td>
</tr>
<tr>
<td>k-NN 1</td>
<td>81.1667%</td>
<td>96.2729%</td>
<td>97.6329%</td>
<td>99.666%</td>
<td>83.2364%</td>
</tr>
<tr>
<td>k-NN 2</td>
<td>79.8540%</td>
<td>95.8880%</td>
<td>96.5283%</td>
<td>99.7118%</td>
<td>83.2674%</td>
</tr>
<tr>
<td>k-NN 3</td>
<td>78.5606%</td>
<td>95.7776%</td>
<td>95.3316%</td>
<td>99.7710%</td>
<td>83.2099%</td>
</tr>
<tr>
<td>k-NN 4</td>
<td>76.6487%</td>
<td>95.8407%</td>
<td>93.3157%</td>
<td>99.7920%</td>
<td>82.8222%</td>
</tr>
<tr>
<td>k-NN 5</td>
<td>76.6734%</td>
<td>95.9591%</td>
<td>92.7429%</td>
<td>99.7948%</td>
<td>82.8158%</td>
</tr>
<tr>
<td>k-NN 6</td>
<td>76.9698%</td>
<td>96.1087%</td>
<td>92.7091%</td>
<td>99.7948%</td>
<td>82.8003%</td>
</tr>
<tr>
<td>pdAPSO</td>
<td>84.1814%</td>
<td>96.7170%</td>
<td>98.8567%</td>
<td>99.8023%</td>
<td>84.8629%</td>
</tr>
</tbody>
</table>

The Figs. 4 – 6 are the graphs depicting the percentage of accuracies of SVM, k-NN and the pdAPSO algorithm for dataset 1 respectively.

Fig. 4. Accuracy of SVM
The result depicted in fig. 4 above shows that SVM have highest accuracy of about 97% for R2L attacks for each of the datasets considered. The lowest accuracy was recorded for Normal dataset.

![Accuracy of k-NN Classifier](image)

**Fig. 5: Accuracy of K-NN classifier**

The result for k-NN classifier is in figure 5 above. The k-NN have highest accuracy of about 98% for R2L attacks for each of the datasets considered. The lowest accuracy was recorded for Normal and U2L datasets.

![Accuracy of pdAPSO Algorithm](image)

**Fig. 6: Accuracy of pdAPSO algorithm**

The result depicted in fig. 6 above shows that pdAPSO has highest accuracy of about 98% for R2L attacks for each of the datasets considered, and closely followed by prober attack.

4.0 Conclusion

This study explores the use of hybrid SVM, k-NN, and pdAPSO algorithms to detect intrusion. It has been established from the experimental results that classification accuracy can be enhanced by hybridizing different classifiers. The paper compared the performance of the three classifiers using the statistical results obtained from the KDD99 datasets. The SVM, k-NN and pdAPSO hybrid approach outperform the other algorithms compared by generating classification accuracy of 98.55%.
References


Curtis A. and Carver Jr (2000). Intrusion Detection System: A Survey. Department of Computer Science, Texas A & M University, College Station, TX 77843-3112, USA


