An Improved KNN Classifier for Anomaly Intrusion Detection System Using Cluster Optimization

Oladeji Patrick Akomolafe¹ and Adeleke Ifeoluwa Adegboyega²
¹,²Department of Computer Science, University of Ibadan
¹akomspatrick@yahoo.com, ²gboygeee@gmail.com

Abstract— With the emergence of anomaly intrusion detection system, varieties of unknown intrusions that were not detected by the misuse or signature based intrusion detection system can now be identified. Anomaly intrusion detection system works by building profiles of normal system state or user behavior, applications and network traffic and continuously monitor the network’s activity so that deviations from the established profiles of normal system state are interpreted as attacks or intrusions. Anomaly intrusion system is efficient but has its weaknesses due to problems arising during classification of known and unknown intrusions such as difficulty of building models of robust behaviours, high false alerts rate caused by incorrect classification of events in current existing system. This paper presents an improved, modified KNN classifier using clustering optimization which is more effective at curbing both known and known intrusions in existing anomaly intrusion detection system. In this paper, we input the attributes of the NSL-KDD training dataset to be classified by the improved KNN(Known Nearest Neighbor) classifier with clustering optimizer inclusive after its been verified by k-mean clustering algorithm and optimized by genetic algorithm respectively. We then evaluate the performance of the improved KNN classifier and compare with the existing KNN classifier and the result showed the existing classifier had a correctly classified test data instance of 98.7% efficiency and 0.2395% for the incorrectly classified instances while the newly developed classifier had a 99.6% efficiency for the correctly classified instances and 0.3222% for the incorrectly classified instances.

Index Terms— Anomaly Intrusion Detection, High False Alerts, Known/Unknown Intrusions, Known Nearest Neighbor Classifier, KDD, Clustering Optimizer and Genetic Algorithm

I. INTRODUCTION

SEVERAL attacks or intrusions such as unauthorized access and login to sensitive files by hackers, host-based attacks such as privilege escalation and the four basic categories of computer threats which include denial-of-service (Dos), user-to-root (U2R), remote-to-user (R2U), probing are being faced by several companies globally, these intrusions or attacks have become a serious issue and as the year passes by different solutions have been offered even though intrusion detection technologies are very essential for computer network security. According to Alonso-Bertanzos et al., 2007, intrusions in computer science refers to set of actions that violates the system security and these has attracted a great deal of attention from scientist in recent years. S.N. Sheela Evangelin, 2015 defined intrusion detection as a process of monitoring and analyzing the events arising in a computer network to identify security breaches.

Sodiya A., et al., 2004 defined intrusion detection system as systems that have the ability to detect both internal and external attacks on a computer system and undertake measures to eliminate them. A large number of intrusion detection approaches be present to resolve this issue but the major problem is performance of the intrusion detection system based on its classification ability, which could be measured or evaluated using certain metrics such as accuracy, time complexity, space complexity, memory consumption and error rate.

A) Background

The origin of intrusion detection system dates back to the early ‘60s when administrators had to sit on their desk computers to monitor user activities and know if the system operation works in a legalized format. It was very effective then but this approach was not all welcoming since the administrators cannot leave their desk computers and need to keep focused at all times. After that the next stage of the detection process, from late ‘70s to early ‘80, was to inspect the system logs.

The administrators manually printed the audit logs on paper, which were often stacked up to four to five feet high by the end of a week, and then search the evidence of hacker behavior, an unusual and/or malicious activity, in such a stack. It was obviously very time-consuming. With this overabundance of information and manual analysis, the administrators mainly used the log content as a forensic data to identify the cause of a particular security incident after the incident happened. A problem arose which was of how to safely secure separate classification domains on the same network without compromising security. This problem still exists today. In year 1980, with James Anderson’s technical report on Computer Security Threat Monitoring and Surveillance, for the U.S. Air Force, Intrusion detection itself
was born. In His research plan which he wrote for U.S. air
force, he announced the “Reference Monitor (RF)”, which
helped a lot in developing intrusion detection techniques till
today. James Anderson introduced the concept of audit trails
containing vital information and proposed that audit trails
should be used to monitor threats (Anderson, 1980).

II. LITERATURE REVIEW

An Anomaly intrusion detection system is a system
representing a type of detection approach under the intrusion
detection system taxonomy. Anomaly detection system
compare activities with normal baseline (i.e after building
profiles of normal system state), they look for deviations from
these normal states by reviewing the characteristics of
external activities and comparing them with that of the system
and label them as intrusions. Anomaly detection system have
two advantages over the signature based system, first
advantage is their capability to detect unknown attacks
because they can model the normal operations of the system
and detect deviations from this model. The second advantage
is customization ability of the normal activity profiles for
every system, application and network. This will increase the
difficulty for an attacker to know what activities can be done
without getting detected. However an anomaly intrusion
detection system cannot correctly identify and equally classify
anomalous behaviors due to challenges in the existing systems
such as high false alarm rates and difficulty of detecting
which events triggers those alarms. For a better classification
and to curb these challenges, we introduce an improved KNN
classifier using cluster optimization to help solve this problem
of the existing system.

A) Related Works

Rajesh Wankhede, Vikrant chole, 2016 proposed a
combination of misuse detection model (ADTree model) and
Anomaly model (Svm based) using the NSL-KDD as dataset
and then applying association mining to generate frequent
patterns for various known patterns. Association mining was
applied to only known attacks and not unknown attacks.

Tavallaee et. al, 2009 proposed NSL-KDD, which contains
selected records of the KDD data and helped overcome the
issues of poor evaluation of anomaly detection approach
thereby improving the performance of the system. Dataset
suffers from problem and may not be a perfect representative
of existing networks, due to lack of public data set for
network-based IDSs.

Mrutyunjaya Panda et.al, 2008 proposed hybrid intelligent
decision technologies using data filtering by adding guided
learning methods along with a classifier to make more
classified decisions in order to detect network attacks. The
results show that there is no single best algorithm to
outperform others accurately in all situations.

M.Medhi et al, 2007 proposed a new approach of an
intrusion detection system that involves building a reference
behavioural model and use of Bayesian classification
procedure to evaluate the deviations between current and
reference behaviour. Preliminary experimentations show that
proposed algorithms have limitations such as that the kernel
distributions are used to model numerical data with
continuous and unbounded nature.

Hong Kuan Sok et.al, 2013 proposed a paper on using the
ADTree algorithm for feature reduction. ADTree also gives
good classification performance. Also, its comprehensible
decision rules endows the user to discover the features that
heads towards better classification. The error rate is closer to
that of c5.0 algorithm.

F. Amiri et al, 2011 proposed feature selection method in
order to improve the performance of existing classifiers by
excluding non-related features. Furthermore, an improved
Partial Least Squares Support Vector Machine called
PLSSVM has been considered in this work. PLSSVM missed
a big number of dynamic attacks such as Dos and U2R attacks
with behavior quite similar to normal behavior. It also
provides low accuracy when compared to LS-SVM.

B) Existing System Review

Genetic algorithm generates a large rule set after the
verified clustered set from the k-mean clustering must have
been taken as input into it; every row in the GA is a rule. One
of the rules specifies that if a certain procedure is being seen
then it is regarded as an intrusion and if it’s the opposite then
it’s not an intrusion. When an activity is being investigated,
the K-Nearest Neighbor module extracts the characteristics of
that activity and compare it with the characteristics described
in the rules to see how close the characteristics of the
observed activity is to the characteristics in the rule set, if the
characteristics is so near (that is similar) then we regard it as
intrusion but if its far away then its not an intrusion, KNN
judges by NEARNESS.

When characteristics is extracted from an observed activity,
it compares it with every line of rules in the rule set, so
assuming there are 5 million lines of rules, the KNN has to do
the comparison 5 million times which consumes
classification time and affects prediction accuracy.

III. METHODOLOGY

The procedures used include the following:

i). Data preprocessing and cleaning

ii). Extraction of significant features sufficient for
classification using k-mean clustering

iii). Learning and optimizing the already extracted clusters
using genetic algorithm

iv). Designing an improved KNN classifier model using
cluster optimization

K-mean clustering was used to perform essential features
extraction through clustering over data and in unsupervised
manner cluster the whole dataset into parts. The verified data
is produced as input to the genetic algorithm, which by part
learns in order to enhance the performance of the KNN
classifier and by part optimizes the solutions for finding the
more appropriate patterns in learning datasets. These
recognized patterns are then classified using KNN algorithm
and performance of the algorithm is evaluated. The figure
below shows the process phase for the methodology, data is
prepared and cleaned before being fed into k-mean clustering
algorithm module which does extraction of significant features sufficient for classification. The genetic algorithm module then learns and optimizes the already extracted clusters which are then passed to the KNN which then does classification.

A) How the Existing System Works

For existing system, the training dataset is being fed into the k-mean clustering algorithm which explores and analyses variability in the training data set in order to extract significant features sufficient for classification. It clusters unlabelled dataset and then verifies which is then taken as input into the genetic algorithm; GA performs the following sub processes:

- Generates initial population
- Evaluates objective functions
- Is optimization met (if yes, the proceeds to output)
- If no, do
  - Selection
  - Recombination
  - Mutation
- Then go through the initial process again

The genetic algorithm partly learns to enhance the performance of the classifier and partly optimizes the solution to find a more appropriate pattern for the clustered set. Once the genetic algorithm has taken the verified clustered set as input and done the search criteria, the output becomes a rule set which is then classified in its enormous amount by the KNN and the performance of the system is evaluated. At the end of the instance, over 100,000 datasets being fed into the k-mean clustering which become a clustered set when taken into the genetic algorithm generates over 1,000,000(one million) rule set which the KNN has to classify by calculating the approximate distance between the various points on the input vectors and assigning the unlabelled points to its class of KNN.

The Fig. 2 shows the flow of the methodology and contains the cluster optimization module which helps to improve the existing for a better classification scheme.

B) Modified KNN Classifier

When cluster optimizer is introduced, it picks the characteristics of the observed activity and holds on to it, after

which it clusters the rule set into groups i.e., group A, Group B, Group C, Group D. so when comparison is to be done for an incoming activity, it is not compared against each item anymore instead it is compared against the characteristics of each group. For example given 5000 rules and 1,500 rules is in group A, 2500 rules is in group B and 1000 is in group C. we now pick the incoming activity, then check which of the groups the characteristics of the incoming activity is close to, if its closest to group C, it automatically discards the rest of the group. Since 1000 rules are present in group C, it checks the comparison against the 1000 rules because it’s the nearest to group C so we know the nearest will come from the C cluster, so instead of doing 5000 comparisons, it does 1000 comparisons (the ones showing high resemblance are retained and one not showing high resemblance are discarded) and saves classification time and accuracy.

The Fig. 3 highlights the difference in the existing system (without cluster optimization) and the proposed system (with cluster optimization).

C) How Cluster Optimization Improves the Existing System

The dataset is fed into the k-mean clustering module which explores and analyses the variability in the training dataset and the cluster this unlabelled dataset then calculates the mean to extract significant features sufficient for classification. This clustered set is then verified and taken as input into the GA and then by default it partly learns to enhance the performance of the classifier and partly optimizes the solution
to find more appropriate pattern for the clustered set. After which it generates a rule set after searching entirely through the clustered set (at the end, an input of 100,000 datasets generates a rule set of 5 million as output from the genetic algorithm). The rules are gathered in the rule set and by default have to be classified by the KNN, KNN classifies by calculating the approximate distance between various points on an input vector and assigns the unlabelled points to the class of the KNN. When an activity is being investigated, the characteristics of that activity is being extracted and compared with the characteristics of the rules in the rule set to see how close they are to each other. KNN deals with nearness, how close the characteristics of both parties are to one another. So the activity characteristic that is close to the rule’s characteristics is classified as an intrusion and those that are far away are seen as non-intrusions.

IV. RESULT ANALYSIS

A) Comparison of Existing KNN Classifier Data with Modified KNN Classifier Data

We have input records of the algorithms used in the improved KNN model using WEKA tool which is given in the following Table I.

Table I: Comparison of Data

<table>
<thead>
<tr>
<th>BENCHMARK</th>
<th>K-MEAN CLUSTERING</th>
<th>GENETIC ALGORITHM</th>
<th>KNN</th>
<th>MODIFIED KNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correctly classified instances</td>
<td>22840/253.3775%</td>
<td>121046/96.33%</td>
<td>181934/81%</td>
<td>42693/69.678%</td>
</tr>
<tr>
<td>Incorrectly classified instances</td>
<td>19946/46.8229%</td>
<td>4507/30.5%</td>
<td>2405 (19%)</td>
<td>338 (0.3222%)</td>
</tr>
<tr>
<td>True positive rate</td>
<td>0.534</td>
<td>0.389</td>
<td>1.0</td>
<td>0.997</td>
</tr>
<tr>
<td>False positive rate</td>
<td>0.534</td>
<td>0.058</td>
<td>0.0</td>
<td>0.003</td>
</tr>
<tr>
<td>recall</td>
<td>0.285</td>
<td>0.357</td>
<td>0.534</td>
<td>0.997</td>
</tr>
<tr>
<td>Mean absolute error</td>
<td>0.4976</td>
<td>0.3636</td>
<td>0.4976</td>
<td>0.0032</td>
</tr>
<tr>
<td>Relative absolute error</td>
<td>0.2841</td>
<td>0.7205</td>
<td>0.0128</td>
<td>0.6498</td>
</tr>
</tbody>
</table>

From above we found that the improved knn classifier (with cluster optimizer inclusive) gave a better classification result to determine the attack by having an efficiency of 99.67% for the correctly classified instances and 0.3222% for the incorrectly classified instances. When we applied it for existing knn test set, it had a correctly classified test data instance of 81% efficiency and 19% for the incorrectly classified instances. While for an existing related work using weka, the existing classifier system had a correctly classified test data instance of 98.7% efficiency and 0.2395% for the incorrectly classified instances. All these differences give the improved knn classifier as the optimal solution to the classification problem for anomaly intrusion detection system.

V. CONCLUSION

This paper presents an improved KNN classifier using clustering optimization for an anomaly based IDS which would provide a more effective classification scheme for existing anomaly intrusion detection system. From our experimental findings, we concluded that the known nearest neighbor is a good classifier for anomaly intrusion detection system but with addition of the cluster optimizer; better classification time, prediction accuracy and less error rate is obtained.

REFERENCES


