Australian Government
Department of Defence
Science and Technology

Text Classification of Network Intrusion Alerts to Enhance Cyber Situation Awareness and Automate Alert Triage

Iain Dickson

Cyber and Electronic Warfare Division
Defence Science and Technology Group

DST-Group-TN-1640

ABSTRACT

For many Cyber Security Incident Response Teams (CSIRT), reacting and responding to suspicious network activity is predominantly a manual task and lacks the necessary levels of automation required to deal with the volume of alerts. Alerts are signalled from tools such as Intrusion Detection Systems (IDS) to skilled analysts who must then decide on courses of action and remediation activities. The IDS alerts are basic; analysts must manually derive context about the alert using their prior knowledge.

In this paper, we describe Artificial Intelligence (AI) techniques used to automate the derivation of context from IDS alerts. We propose two algorithms based on well-known automated text classification methods and define a multi-level taxonomy to describe classifications of alerts in a semantically hierarchical manner. Consideration is given to the use of these algorithms by a CSIRT, as well as how Situation Awareness (SA) can be improved through automation. Our findings show that a combination of Naïve Bayes algorithms in conjunction with our proposed hierarchical taxonomy can automate alert classification with high accuracy, and low false and unclassified rates.

RELEASE LIMITATION
Approved for Public Release

Produced by

Cyber and Electronic Warfare Division
Defence Science and Technology Group
PO Box 1500
Edinburgh SA 5111

Telephone: 1300 333 362

June 2017
AR-016-883

Text Classification of Network Intrusion Alerts to Enhance Cyber Situation Awareness and Automate Alert Triage

Executive Summary

Organisations are faced with the continual threat of intrusion and infection of their mission critical networks. With networks becoming more complex, detecting suspicious activity has become challenging. Maintaining an awareness of one's own networks and their activities has therefore become of paramount importance.

Network intrusion detection systems (NIDS) are one set of tools available to organisations to detect suspicious activity. NIDS detect suspicious activity and flag it to analysts through alerts. Skilled network security analysts must determine the context of each alert using their prior knowledge and experience. This process is difficult to automate due to a lack of algorithms that can understand the alert context, resulting in the current manually intensive nature of triage.

Algorithms that derive context from NIDS alerts are described within this paper. These algorithms are based on well-known text classification algorithms such as Naïve Bayes. NIDS alerts are classified by these algorithms against a proposed hierarchical taxonomy of suspicious activity, defined in this paper. This taxonomy enables further automation of network security processes.

Our algorithms were compared against existing text classification algorithms. We determined that our proposed combination of Naïve Bayes algorithms is highly effective at classifying alerts accurately, and therefore suitable for Computer Security Incident Response Teams (CSIRT) use.

This page is intentionally blank.

Author

Iain Dickson
Cyber and Electronic Warfare Division

Iain graduated with a Bachelor of Science (Computer Science) and a Bachelor of Education (Secondary Education) from the University of South Australia in 2013. Iain has since been employed by the Defence Science and Technology Group, within the Automated Analytics and Decision Support group of the Cyber and Electronic Warfare Division

This page is intentionally blank.

Contents

This page is intentionally blank.

Glossary

AI	Artificial Intelligence
AV	Antivirus
C2	Command and Control
CDX	Cyber Defence Exercise
COTS	Commercial-off-the-Shelf
CSIRT	Computer Security Intrusion Response Team
CTF	Capture the Flag
HIDE	Hybrid Intrusion Detection Engine
HSnort	Hybrid Snort
IDS	Intrusion detection system
IREP	Incremental reduced error pruning
MACCDC	Mid-Atlantic Collegiate Cyber Defense Competition
NGIPS	Next Generation Intrusion Prevention System
OODA	Observe-Orient-Decide-Act
P2P	Peer-to-Peer
PCAP	Packet Capture
SIEM	Security Information and Event Management
SA	Situation Awareness
SOP	Standard Operating Procedure
SVM	Support Vector Machine

This page is intentionally blank.

1. Introduction

Organisations face a continued threat from cyber-attack, ranging from spear phishing to accidental malware infection (Symantec Corporation 2013). The threat continues to grow with the increasing number and sophistication of adversaries, varying from activist groups to individuals and criminal organisations. The ensuing vulnerability is unlikely to decline due to the increasing reliance on networked computer systems to undertake core business. It is therefore a necessity for an organisation to identify and respond to suspicious network activity and to protect and defend its ICT infrastructure to ensure business continuity (Onwubiko 2012).

Network security analysts use a number of tools to detect and analyse suspicious network activity. These include intrusion detection systems (IDS), anti-virus engines and firewalls. Standard operating procedures (SOP) and security information & event management systems (SIEM) provide the core supporting analysts in their computer network defence (CND) activities (Goodall, Lutters et al., 2009). While a SIEM provides critical situation awareness and decision support, analysts also require deep technical knowledge when conducting incident response functions such as triage and analysis.

An experienced network security analyst can quickly and efficiently derive the origin and intent of the suspicious activity from the information within an IDS alert (Goodall, Lutters et al., 2009). However, analysts are required to deal with an increasing volume of alerts. This increases an individual analyst's cognitive load and places stress on limited analyst resources. Effective automation can reduce the requirement for manual handling of alerts (Grobler and Bryk 2010). Automation of mundane tasks such as triage frees up resources and allows analysts to focus on complex incidents that require greater time investment and experience.

Analysts rely on the free text message in the alert coupled with their experience to determine the nature of the potential attack. However, typically, this message is written by the signature developer in free text and does not conform to a set standard or a machine interpretable format. Having the alert context expressed in a machine interpretable form can support effective automation of incident response functions. A number of techniques have been employed to classify alerts to support machine interpretability. These include text classification algorithms (Sebastiani 2002) and statistical flow analysis (Sperotto, Schaffrath et al., 2010).

This paper describes algorithms designed to automatically classify an alert to provide context in support of automated incident response functions. These algorithms enrich each alert with machine interpretable contextual information that would otherwise require analyst intervention. This enrichment supports further automation of incident handling and triage procedures. This automation in turn decreases operator work and cognitive load and improves response time.

The contributions of this paper are threefold. Firstly, an approach is defined to contextualise IDS alerts in a generic fashion. This approach provides a graceful degradation in categorisation to provide maximum support for machine reasoning.

Secondly, these algorithms are evaluated in terms of their efficacy and accuracy. Thirdly, these algorithms are compared with existing algorithms to determine their relative effectiveness.

This paper is structured as follows. Section 2 provides background information; discussing intrusion detection systems, the current incident-handling paradigm and the situation awareness concept. Section 3 describes related work on text classification and efforts to integrate artificial intelligence and intrusion detection systems. Section 4 outlines the proposed solution and the two algorithms developed. Section 5 describes the algorithm evaluation procedure and metrics, while Section 6 details the evaluation results. Section 7 analyses the performance of each algorithm in turn, providing discussion of the merits of each. Section 8 discusses future work, with conclusions in Section 9.

2. Preliminaries

2.1 Intrusion Detection Systems

Administrators of early computer system networks relied on manually reviewing audit logs to identify suspicious activity (Kemmerer and Vigna 2002). Awareness of network state was challenging to maintain due to the lack of real time information. Development of real time monitoring tools was driven by the continuous increase in suspicious activity and enabled by increases in data storage and computer processing power.

IDS are the foremost of tools that detect and report suspicious activity. Two broad categories of these systems exist; host intrusion detection systems (HIDS) and network intrusion detection systems (NIDS). Host intrusion detection systems are installed on critical host machines, detecting suspicious activity specific to that host (Mukherjee, Heberlein et al., 1994). Network intrusion detection systems comprise sensors placed at critical points in a network enabling detection and identification of a broad range of suspicious activity from network traffic.

Historically, it was common for NIDS to employ a detection process as follows:- Suspicious behaviour is detected through techniques such as rule matching and statistical analysis (West-Brown, Stikvoort et al., 2003).When these behaviours are detected, an alert is signalled to network security analysts (Gagnon and Esfandiari 2007). The analyst then uses the information contained within the alert to determine whether the alert is malicious or not, and what remediation actions are required. This information includes the source and destination addresses of the traffic, the time, a complete copy of the raw packet traffic and a text message describing the suspicious activity detected by the signature.

Snort (Roesch 1999) is one of the most influential IDS developed to date. The open source IDS was developed to use a structured rule schema to define alert signatures. Snort and its structured rule schema have become a pseudo-standard for IDS engines and it is employed by several of the solutions described in this paper. Snort's default categorisation schema is also used as the base for our taxonomy, defined in Figure 3 (Sourcefire 2013).

2.2 Assessing Suspicious Network Activity

Computer security incident response teams (CSIRT) have been established by organisations to respond to the rising threat of cyber-attack (West-Brown, Stikvoort et al., 2003). CSIRTs consist of a number of network security analysts whose tasks are to evaluate and act upon cyber security incidents. Their role is threefold: to detect suspicious activity; to coordinate the response; and to provide remediation solutions (Grobler and Bryk 2010). CSIRT analysts are often trained through practical experience rather than through formal tertiary education and qualification.

CSIRTs use a triage process for prioritising alerts based on severity and impact (Rogers, Goldman et al., 2006); after which the ranked alerts are investigated and remediated. Tools such as virus scanners and sandboxes[1] are used to determine the impact of the activity. Web proxy logs, malware databases and captured raw network traffic are used for corroboration. The incident response process is manually driven; the analyst determines the courses of action that should be undertaken based on their personal experience with reference to the organisation's security standard operating procedures (SOP).

Suspicious network activity has become more complex and time consuming to remediate (Onwubiko 2012). It is harder to detect, more difficult to process, and requires more specialist training. Automated solutions have the potential to perform mundane tasks relieving analysts to focus on more complex activity.

2.3 Situation Awareness

Maintaining awareness of an environment is more commonly known as maintaining Situation Awareness. Situation Awareness is defined by Endsley (2003) as:

> the **perception** of the elements in the environment within the volume of time and space, the **comprehension** of their meaning and the **projection** of their status in the near future.

Endsley developed this definition after refining a previous paradigm by Boyd (1987), known as the Observe, Orient, Decide, Act (OODA) loop. Figure 1 illustrates the OODA loop. This framework was developed to explain why United States Air Force (USAF) fighter pilots were more successful in dogfights than Vietnamese pilots during the Vietnam War. Boyd concluded that American pilots were able to complete the loop quickly, leading them to anticipate the moves of their opponents.

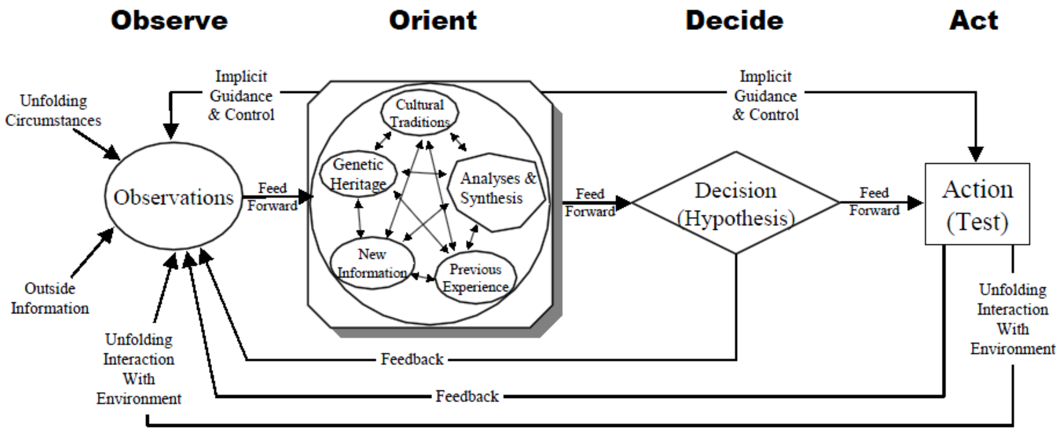

Figure 1: OODA loop as described by Boyd

[1] Sandboxing is defined as "the concept of confining a helper application to a restricted environment, within which it has free reign." (Goldberg et. al, 1996)

Endsley’s initial work focused on applying her paradigm further to the fighter pilot realm. Over time, Situation Awareness has been applied to other traditional battle-spaces such as Land and Sea. The situation awareness model has also been considered when describing networked cyber systems (Endsley 2003, Onwubiko 2012).

The situation awareness paradigm informs how security analysts investigate suspicious activity. An organisation’s cyber security function can be conceptualised using the OODA framework and Endsley’s model (Onwubiko 2012). An intrusion detection system *observes* the environment which yields *perception*. This perception of the network state is provided to an analyst in the form of alerts. The analyst performs the function of *comprehension* and *orientation* in the environment by confirming and determining the intent and extent of an unfolding or potential attack. From here the analyst then proceeds to *project* or *decide* and *act*, implementing activities to effectively remediate the situation.

It is this transformation from perception to comprehension that automated systems currently lack. A solution to transform perception to comprehension would enable the construction of systems that automatically understand and comprehend the network situation (Onwubiko 2012). A core requirement is sensors that perceive the environment to provide observations in machine interpretable semantics and syntax. The application of text classification techniques allows for the addition of context to support machine interpretability and thus automation.

3. Related Works

3.1 Text Classification Algorithms

Volumes of data available to information science researchers have increased in recent years (Sebastiani 2002). These researchers have attempted to develop efficient and effective document categorisation and classification algorithms. In particular, significant effort has been expended within the text document problem space. Automatable algorithms such as Support Vector Machines (SVM), Decision Trees, and Naïve Bayes have shown particular promise.

SVMs have demonstrated promise in classifying text documents (Joachims 2001). Labelled training data is mapped to a feature space and linear region-separating rules are defined based on the labels of the data. New documents are classified by optimizing their position within these constraints, and a relative probability is calculated. SVMs have been demonstrated to be an effective text based classifier when considering a number of problem domains (Joachims 1998, Joachims 1999, Joachims 2001, Tong and Koller 2002).

A Decision tree is another approach to text classification. This algorithm defines a classification as a tree structure of decisions and probabilities. (Aggarwal and Zhai 2012). New documents are classified through tree traversal and assessing the relevant decisions made to reach an endpoint. A decision tree can be constructed manually by an expert or learned from an initial training set. Apte, Damerau et al., (1994) and Lewis and Ringuette (1994) have evaluated this algorithm within the context of text classification.

Naïve Bayes classifiers employ conditional probability to determine the likelihood of words appearing in a text given a class (Aggarwal and Zhai 2012). New documents are assigned a cumulative probability based on the sum of individual word probabilities. Naïve Bayes classifiers assume that the appearance of a word is independent of the appearance of any other word (Lewis 1998). This assumption is rarely true for natural language. Lewis evaluated the Naïve Bayes classifier and concluded that despite this assumption, it is highly successful. Denoyer and Gallinari (2004), and De Campos Fernandez-Luna et al., (2008) have worked to improve the Naïve Bayes' accuracy through weighting systems and pre-processing of data respectively. Naïve Bayes classifiers require labelled training data; a time consuming manual task to create, often requiring a subject matter expert.

Sebastiani (2002) undertook a literature review highlighting the wide scope of text classification approaches that have been evaluated. Sebastiani posits that classifiers are context dependent. Evaluation must be conducted to determine the most appropriate classifier for any particular problem space and the data features that are most useful in determining a classification.

3.2 Classification of Intrusion Detection System Alerts using Machine Learning

A number of researchers have investigated augmenting IDSs with machine learning (ML) to improve situation awareness comprehension.

Divya and Surender's (2013) describe a combination ML/IDS system called Hybrid Snort (HSnort) which ingests raw Snort alerts and classifies them using an artificial neural network. These alerts are then passed to a SIEM system for analysts to study further and remediate.

Subbulakshmi et al., also build upon Snort to create a clustering and classification system, with the aim of decreasing the number of false positive alerts issued to analysts (Subbulakshmi, Mathew et al., 2010). Their architecture is shown in Figure 2. Alerts are collected, normalised and pre-processed, before being fused and correlated. These alerts are then classified using pre-learnt rules and issued to the analyst for investigation.

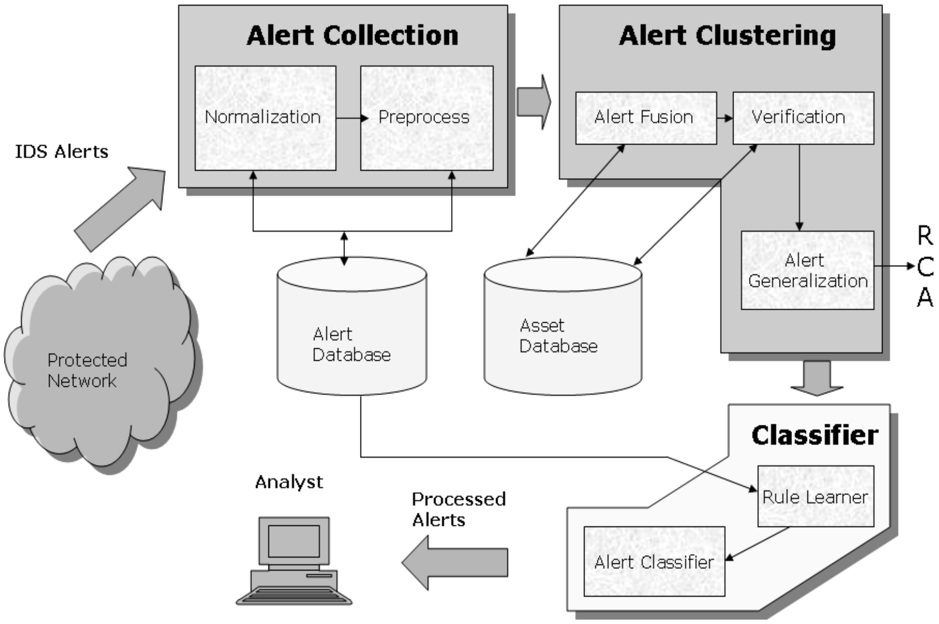

Figure 2: Subbulakshmi et al., alert processing architecture

These rules were learnt using the RIPPER (Cohen 1995) algorithm in conjunction with an artificial neural network classifier. Their results show that the RIPPER rules had a low false negative rate compared to a number of baseline algorithms, including random forest and decision stump. They also determined that human interaction and grooming of the rules was not required to improve their effectiveness.

Zhang and Li employ a similar approach in their system: the Hybrid Intrusion Detection Engine (HIDE) (Zhang, Li et al., 2001, Zhang and Manikopoulos 2003). An artificial neural network is embedded directly within their sensors; termed Intrusion Detection Agents (IDA). Statistical processing is used to identify suspicious network activity and the artificial neural network is used to classify them. Unlike the previous examples, this approach requires custom engineered sensors to be deployed across an organisation rather than providing an overarching solution for any commercial-off-the-shelf (COTS) product.

Gagnon and Esfandiari (2007) contend that artificial intelligence should be integrated directly into intrusion detection sensors rather than as a supplementary component. This would enable the classification of alerts at the time of detection, rather than as a secondary process. This supports Zhang and Li's proposal, which would require dedicated sensors to be deployed across an organisation. Subbulakshmi's approach however, allows the integration of heterogeneous sensors.

4. Alert Classification Taxonomy and Algorithms

4.1 Classification Taxonomy

We present a taxonomy of suspicious network alerts developed to provide consistent, machine interpretable classifications with clear semantic association. The *Suspicious Event Taxonomy* is shown in Figure 3, with definitions for each node provided in Appendix B. Categories from the Snort categorisation schema were grouped by their overarching concept by the author to construct this taxonomy (Sourcefire 2013). The taxonomy is hierarchical in nature with the top layer (Level 1) representing high level concepts for the stages of an intrusion such as *reconnaissance* or *exploitation*. The second layer (Level 2) represents specialisations (of Level 1 concepts) to specific suspicious alert categories.

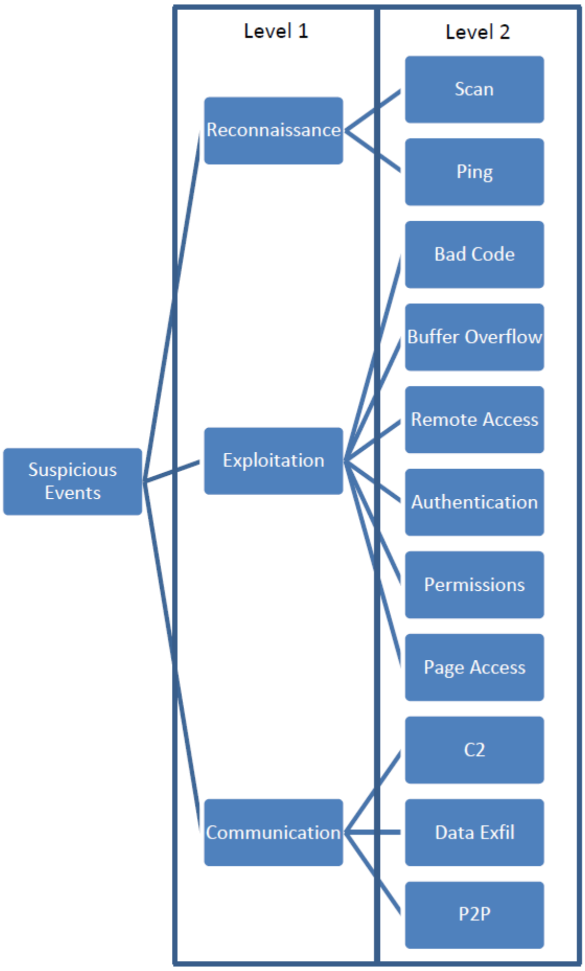

Figure 3: Suspicious Event Taxonomy

The hierarchical structure of the design allows for graceful degradation in classification precision, whereby alerts are classified to as specialised a category as possible. For example, an alert can be classified as *C2*, a Level 2 classification. However, if the algorithm fails to classify that alert at Level 2, it can still be classified as *Communication*, the Level 1 parent of *C2*.

4.2 Baseline Classifiers

Two commonly used text classification algorithms were chosen as a baseline against which to evaluate the effectiveness of our algorithms. These are the Rule Classifier and Naïve Base Classifier.

4.2.1 Rule Classifier

The Rule Classifier was selected to capture expert knowledge applied by an analyst in categorising suspicious activity. The Rule Classifier uses a set of subject matter expert defined rules to classify each alert. The Snort ruleset was used to determine common features held by categories at Level 1 of the taxonomy, which were used to generate a set of Prolog rules (Sourcefire, 2013). An alert is successfully classified when a set of the alert's features completely matches against the set of conditions specified by any rule within the rule set.

The rules developed for our experiment used the *source port*, *destination port* and *protocol* features of the alert for categorisation. These rules are likely to be restrictive; alerts that are only slightly different from the prescribed rule set are likely to be ignored. Constant updating of these rule sets would be required to counter new threats, and because of this, rules only to classify to Level 1 were devised.

4.2.2 Naïve Bayes Classifier

A Naïve Bayes classifier was chosen as a baseline classifier due to its simple, yet effective nature, as discussed by Lewis and Ringuette (1994). This Naïve Bayes Classifier requires a labelled data set for training. The data pre-classification process for these alerts is detailed in Section 5.

The Naïve Bayes classifier was trained utilising the message string of the alert. The test set alerts were then classified against this training data. For this particular algorithm, the class with the highest belief probability was deemed the correct classification.

4.3 Hierarchical Classification Algorithms

We propose the following two algorithms in conjunction with the taxonomy defined in Section 4.1 as a method of determining the context of suspicious activity. These algorithms, termed hierarchical classification algorithms, use combinations of the baseline classifiers to classify at different levels of the taxonomy. This multi-level approach allows for a graceful degradation of classification depending on the information available.

4.3.1 Naïve Bayes – Rule Classifier (NB-R Classifier)

The Naïve Bayes – Rule Classifier is the first of these hierarchical classifiers. Alerts are classified by the first stage of this algorithm, a Naïve Bayes classifier. This works at Level 2 of the *Suspicious Event Taxonomy* using the alert message string. If the alert cannot be classified by the first stage, then it is classified by the second, a Rule Classifier using the *source port*, *destination port* and *protocol*.

In between the first and second stage of the algorithm, the concept of a *difference threshold* is introduced. This is designed to decrease the effects of misclassification when two potential Level 2 classes have similar probabilities. If the difference between the two highest probabilities for the first stage of the classifier is within the threshold, then the classification is discarded, and the alert is classified by the second stage. This value was set to 10% in these experiments, although the effects of different settings should be evaluated as part of any future work.

4.3.2 Naïve Bayes – Naïve Bayes Classifier (NB-NB Classifier)

The Naïve Bayes – Naïve Bayes classifier is another two phase algorithm, consisting of two Naïve Bayes classifiers. In the first phase, alerts are classified using a Naïve Bayes classifier at Level 2 of the taxonomy based on their *message* string. If this classifier is unable to classify the alert, it is then classified by the second phase, a second Naïve Bayes classifier using the *source port*, *destination port* and *protocol*. This algorithm also uses the same difference threshold introduced in the NB-R classifier.

5. Methodology

Finding an open source data set of IDS alerts to evaluate the proposed algorithms proved difficult. The primary reason for this is the organisational sensitivity of publishing real IDS alerts. To generate a data set, it was determined that the use of open source packet capture (PCAP) in conjunction with an IDS system could be used; an approach suggested by Sangster (2009).

For this experiment, two datasets were chosen as follows:

- Mid-Atlantic Collegiate Cyber Defence Competition (MACCDC) (2012): MACCDC provides college students with real life cyber defence experience. Teams of competitors from a number of colleges defend against a series of cyber-attacks whilst ensuring that their network remains operational for their fictional corporation.
- DEFCON 17 (2013): The annual DEFCON Conference provides information regarding current security techniques and systems. The Capture the Flag (CTF) competition allows for practical experience with these new exploits and techniques.

Each of these datasets was evaluated to determine the relative proportions of each alert type.

Snort in "Read PCAP" mode was used to generate IDS alerts for each dataset. Snort's wide spread adoption, open signature specification and log format were the reasons for its choice. It is expected that similar IDS products could be evaluated with these algorithms if proven successful. The Snort signature library used within this research was taken from the *Sourcefire Vulnerability Research Team* daily update, on 26 July 2013 (Sourcefire 2013).

The corpus of Snort alerts were then labelled based on the *Suspicious Event Taxonomy* to create a supervised data set. Each classifier was evaluated against this data set 25 times, using a random 90/10 split between training and test data.

Four metrics were captured during each test run. The mean of each metric over all 25 runs was calculated, and are listed as follows:

- **Average Level 1 True Rate**:
  The proportion of alerts that were successfully classified only at Level 1 across all test runs.
- **Average Level 2 True Rate**:
  The proportion of alerts that were successfully classified at Level 2 across all test runs.
- **Average False Rate**:
  The proportion of alerts that were classified incorrectly across all test runs.
- **Average Unclassified Rate**:
  The proportion of alerts that were not classified across all test runs.

Finally, Wilcoxon Signed-Rank (Wilcoxon 1945) tests were conducted using results for each classifier over each dataset. This test was chosen due to the two dependent populations that were to be evaluated. The hypotheses for these tests were as follows:

H_0: There is no median difference between the pairs.

H_1: There is a median difference between the pairs.

These tests were used to determine whether there was a significant statistical difference between the two classifiers' performance in all metrics, using a confidence interval of 95%.

6. Results

6.1 Initial Data Comparison

Several important features can be drawn out from an initial evaluation of the two datasets and their relative proportions of labelled alerts. These proportions can be seen in Table 1.

Firstly, *Reconnaissance* activity makes up the largest proportion of alerts in both datasets. According to our taxonomy, this category consists of simple *Pings* and *Scans*. As discussed by Onwubiko, *Scans* and *Pings* are a common precursor to further malicious activity (2012). This high proportion therefore corresponds with suspicious activity levels that organisations would likely face.

Table 1: Proportions of alerts in the datasets

Classification	**MACCDC**	**DEFCON**
Reconnaissance	76.08%	70.74%
■ Scan	15.00%	1.85%
■ Ping	61.08%	68.89%
Exploitation	23.89%	29.23%
■ Bad Code	9.09%	1.60%
■ Buffer Overflow	0.88%	0.83%
■ Remote Access	0.01%	2.54%
■ Authentication	5.35%	2.62%
■ Permissions	0.01%	2.56%
■ Page Access	8.54%	16.08%
Communication	0.03%	0.03%
■ C2	0.01%	0.01%
■ Exfil	0.01%	0.01%
■ P2P	0.01%	0.01%

Secondly, each dataset contains a large proportion of *Exploitation* alerts, although the proportions of Level 2 categories vary. For example, the MACCDC dataset shows a large proportion of *Bad Code*, *Authentication* and *Page Access* based alerts. The DEFCON dataset shows a larger proportion of alerts were related to *Page Access* and an even spread amongst other sub categories.

The percentage of *Reconnaissance* alerts classified as *Scan* or *Ping* in each dataset also exhibits differences. The DEFCON set contains a very small proportion of *Scan* alerts when compared to the MACCDC data set. This appears to be due to the sources of each dataset. The DEFCON capture the flag activity focused on exploiting a small network with specific vulnerable servers. The MACCDC competition focused on exploiting a large scale corporate network with unknown vulnerabilities. Attackers would therefore have had to scan systems on the network to determine vulnerabilities, rather than target them directly.

6.2 Calculated Metrics

The calculated metrics for each of the datasets and algorithms are described in Table 2 and Table 3.

Table 2: MACCDC Calculated Results

Algorithm	Average Level 1 True Rate	Average Level 2 True Rate	Average False Rate	Average Unclassified Rate
Rule	82.23%	N/A	0.16%	17.61%
Naïve	N/A	92.30%	5.85%	1.86%
NB-R	3.12%	87.88%	5.19%	3.81%
NB-NB	5.76%	87.88%	5.74%	0.63%

Table 3: DEFCON Calculated Results

Algorithm	Average Level 1 True Rate	Average Level 2 True Rate	Average False Rate	Average Unclassified Rate
Rule	83.85%	N/A	0.18%	15.97%
Naïve	N/A	96.78%	2.35%	0.87%
-NB-R	10.72%	76.92%	1.28%	11.08%
NB-NB	21.12%	76.92%	1.67%	0.29%

The Rule Classifier is highly effective at classifying alerts to Level 1, as visible from the consistently high average Level 1 true rate. This classifier also has the lowest average false rate of any algorithm evaluated. This shows that proportionally few alerts are being classified incorrectly, and that the rules generated are accurate. However, the high average unclassified rate for this algorithm in both datasets demonstrates that these rules were unable to classify a larger proportion of alerts.

Important characteristics of the proposed algorithms are revealed from the Level 1 classification rates. The average Level 1 true rate for the NB-NB algorithm is almost double that of the NB-R algorithm for both data sets. This in conjunction with the average unclassified rates indicates that the NB-NB algorithm is more successful at classifying alerts based on its learnt rules. The higher average false rate of the NB-NB algorithm however, indicates that some alerts are being classified incorrectly more often than the NB-R algorithm. The significance of this is discussed in Section 6.3

The use of a *difference threshold* in the two algorithms has a visible effect on their results as compared to the baseline algorithms. The two algorithms exhibit a lower average Level 2 true rate than the baseline Naïve Bayes classifier. This may suggest that the Naïve Bayes classifier was more effective than the two algorithms at classifying to Level 2 but is actually due to the *difference threshold*. Those alerts which were not classified at Level 2 by the two algorithms had probabilities which fell within the *difference threshold* (10%). In this case, those alerts were re-classified at Level 1. As a large proportion of those alerts were still classified correctly, this may highlight that the difference threshold needs to be lowered in future experiments.

The algorithms also exhibit differences when evaluated against the two distinct datasets. For example, a larger proportion of alerts are classified at Level 2 by the algorithms against the MACCDC data set as compared to the DEFCON data set. The average Level 1 true rates and the average false rates for the DEFCON data set were also higher than the MACCDC data set. This appears to be directly related to the relative proportion of alert types within each dataset, as described in Section 6.1

6.3 Statistical Testing

Wilcoxon Signed Rank tests can be used to determine statistical significance when considering two dependent populations. When considering Wilcoxon Signed Rank results, Z values greater than 1.645, the critical value for a 95% confidence interval, indicate a rejection of the null hypothesis. When considering these results, several significant statistical differences are visible. These values can be seen in Table 4.

Table 4: MACCDC and DEFCON Wilcoxon Signed Rank Results

	MACCDC		**DEFCON**	
	Z Score	**Reject?**	**Z Score**	**Reject?**
Average Level 1 True Rate	4.58	Yes	4.29	Yes
Average Level 2 True Rate	-0.01	No	-0.01	No
Average False Rate	4.54	Yes	4.45	Yes
Average Unclassified Rate	4.58	Yes	4.29	Yes

These tests show that we reject the null hypothesis for all metrics other than the average Level 2 True Rate. This means that there is a significant statistical difference between the results of the NB-R and the NB-NB classifiers for all metrics captured except the average Level 2 True Rate. The reason for the lack of rejection for the average Level 2 True rate is due to our use of an identical Naïve Bayes algorithm as our first stage. The results also show that the there is a statistically significant difference between the average false rate of each algorithm; in this case NB-R algorithm is lower than the NB-NB algorithm. The NB-NB algorithm however has a statistically higher average Level 1 true rate and a lower average unclassified rate than the NB-R. The NB-NB classifier is therefore more effective at classifying alerts overall, but at the cost of a higher average false rate.

7. Discussion

The Rule classifier, as demonstrated, can classify alerts to Level 1 accurately. The high average Level 1 true rate in conjunction with the low average false rate supports this conclusion. However, the relatively high average unclassified rate indicates that a large proportion of alerts are failing to be classified. This appears to be due to the simplistic nature of the rules created rather than an issue with the algorithm itself. To alleviate these issues, more discriminatory rules could be devised, using more of the alert's characteristics. In most CSIRT environments it would be infeasible to continuously generate these rules due the fast paced nature of new threats. However, the Rule classifier could be used to supplement a CSIRT and network security analysts by bulk classifying unimportant or low priority alerts with minimal cost.

The Naïve Bayes classifier has the highest average Level 2 classification rate of all of the algorithms evaluated; however it also exhibits the highest average false rate. These results suggest that although the classifier is highly successful at classifying to Level 2, these classifications are more likely to be incorrect. Incorrect classifications can result in lost time, inappropriate remediation strategies, and risks further damage to an organisations network. As a supervised training algorithm, the Naïve Bayes classifier requires a pre labelled context specific data set. Creating training sets can however be time consuming to create initially.

The NB-R algorithm has a relatively low average false rate and high average unclassified rate. Alerts that are classified by the NB-R algorithm successfully are therefore more likely to be correct than the NB-NB classifier. However, as exhibited by the rule classifier, a large proportion of alerts are not classified, due to the simple nature of the rules provided. Again, more expressive and complex rules could be developed, but in conjunction with the pre-labelled context specific dataset, this algorithm is the most infeasible to maintain

The NB-NB algorithm exhibits high average Level 1 and Level 2 classification rates and a low average unclassified rate. This algorithm is therefore ideal for ensuring that a large proportion of alerts are classified. Compared to the NB-R classifier however, the NB-NB algorithm has a high average false rate. Alerts are therefore more likely to be falsely classified. As with the Naïve Bayes algorithm, the NB-NB classifier only requires a single data set for training, as different attributes of the same labelled alert are used for each stage. Due to its performance and its minimal ongoing cost, this algorithm is the most suitable for use by network security analysts.

The two proposed algorithms appear to inherit traits of the baseline classifiers they are comprised of. The NB-R algorithm for example, appears to exhibit the same low average false rate of the Rule classifier. The NB-NB algorithm also appears to inherit the Naïve classifier's higher average false rate and low average unclassified rate. It is proposed that combining classifiers with desirable traits could result in a much stronger classifier tailored to a particular context. Investigating this potential relationship is out of the scope for this paper; however it is discussed further under future work in Section 8.1

The use of a multi-level taxonomy has shown promise within this experiment. Large proportions of alerts were classified successfully at Level 1 after failing to be classified at Level 2. This graceful degradation ensures that alerts can be classified at a coarser level rather than remaining unclassified. Automated systems can then be developed which triage the alert further, rather than remain unable to process.

As discussed in Section 6, the MACCDC and DEFCON datasets exhibit differences in the proportion of their labelled alerts. The algorithms also exhibited differences in their classification performance between datasets. For example, proportions of Level 1 average true rates and average unclassified rates for the MACCDC dataset were one third of those for DEFCON. This difference between datasets highlights the statement posited by Sebastiani (2002); that classifiers need to be tailored to their context. CSIRTs looking to implement similar algorithms should take this into account and evaluate all potential algorithms when considering their context.

8. Future Work and Improvements

8.1 Rule Learning Algorithms

The Rule classifier evaluated in this paper uses pre-defined rules to categorise alerts. These rules were generated by subject matter experts, and proved accurate in classifying a large proportion of alerts to Level 1. However, they required a subject matter expert to encode them using a machine understandable format. This task would be infeasible on a larger scale and would be inadequate when considering the changing landscape of alerts. To augment this approach a number of algorithms have been developed to continuously learn and improve the rules required to classify objects. These include incremental reduced error pruning (IREP) (Furnkranz and Widmer 1994) and RIPPER (Cohen 1995). These techniques could also generate rules which would allow classification at Level 2.

8.2 Combining Classifiers

The algorithms described in this paper made use of a combination of classifiers to improve performance. The NB-NB algorithm demonstrated this by classifying a larger proportion of alerts than the pure Naïve Bayes algorithm. Further work employing this approach would combine other forms of classifiers in a similar fashion. Examples of algorithms that could be combined include Neural Networks, Decision Trees, or Support Vector Machines. The usefulness of graceful degradation of classification and the defined taxonomy should be evaluated. This would also confirm whether a combination classifier inherits traits from its component algorithms.

9. Conclusion

Current incident handling paradigms have become inadequate in today’s threat environment. The volumes of threats to organisations are increasing and becoming more complex to detect and remediate. When considered against the low numbers of network security analysts available, automated systems are a necessity. Automatically classifying suspicious IDS alerts provides context that network security analysts would otherwise be required to manually derive. Furthermore, using machine derived and interpretable context would enable further automation in future systems.

IDS alert classifiers based on algorithm combinations have been designed and implemented. These classifiers have been evaluated against baseline classifiers based on two commonly used text classification algorithms to determine their effectiveness. Our results show that our classifiers have advantages over the baseline classifiers, including the graceful degradation of classification depending on a confidence threshold. The NB-NB classifier in particular excelled due to its high average Level 1 and Level 2 classification rates and low average unclassified rate.

The algorithms discussed in this paper derive important context from suspicious IDS alerts that would otherwise require a network security analyst to determine. Implementing these algorithms enables the development of automated triage and response systems. We expect these automated systems would decrease cognitive load on CSIRT analysts and improve situational awareness, especially comprehension, of the issues affecting the organisation.

10. References

1. Aggarwal, C. C. and C. Zhai (2012). A survey of text classification algorithms. Mining Text Data, Springer**:** 163-222.

2. Apté, C., F. Damerau and S. M. Weiss (1994). "Automated learning of decision rules for text categorization." ACM Transactions on Information Systems (TOIS) **12**(3): 233-251.

3. Boyd, J. R. (1987). A discourse of winning and losing.

4. Cohen, W. W. (1995). Fast Effective Rule Induction. Proceedings of the Twelfth International Conference on Machine Learning, Lake Tahoe, California.

5. Cohen, W. W. (1995). Fast effective rule induction. Proceedings of the twelfth international conference on machine learning.

6. De Campos, L. M., J. M. Fernández-Luna, J. F. Huete and A. E. Romero (2008). Probabilistic methods for structured document classification at inex'07. Focused Access to XML Documents, Springer**:** 195-206.

7. DEFCON. (2013). "DEFCON Capture the Flag Archive." Retrieved 1/11/2013, 2013, from https://www.defcon.org/html/links/dc-ctf.html.

8. Denoyer, L. and P. Gallinari (2004). "Bayesian network model for semi-structured document classification." Information processing & management **40**(5): 807-827.

9. Divya, Surender, L (2013). "HSNORT: A Hybrid Intrusion Detection System using Artificial Intelligence with Snort." Int.J.Computer Technology & Applications **4**(3): 466-470.

10. Endsley, M. R. (2003). Designing for situation awareness: An approach to user-centered design, Taylor & Francis US.

11. Furnkranz, J. and G. Widmer (1994). Incremental reduced error pruning. International Conference on Machine Learning.

12. Gagnon, F. and B. Esfandiari (2007). Using Artificial Intelligence for Intrusion Detection.

13. Goodall, J. R., W. G. Lutters and A. Komlodi (2009). "Developing expertise for network intrusion detection." Information Technology & People **22**(2): 92-108.

14. Grobler, M. and H. Bryk (2010). Common challenges faced during the establishment of a CSIRT. Information Security for South Africa (ISSA), 2010, IEEE.

15. Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features, Springer.

16. Joachims, T. (1999). Transductive inference for text classification using support vector machines. ICML.

17. Joachims, T. (2001). A statistical learning learning model of text classification for support vector machines. Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM.

18. Kemmerer, R. A. and G. Vigna (2002). "Intrusion detection: a brief history and overview." Computer **35**(4): 27-30.

19. Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. Machine learning: ECML-98, Springer**:** 4-15.

20. Lewis, D. D. and M. Ringuette (1994). A comparison of two learning algorithms for text categorization. Third annual symposium on document analysis and information retrieval, Citeseer.

21. MACCDC. (2012). "7th CyberWatch Mid-Atlantic CCDC." Retrieved 1st November, 2013, from http://maccdc.org/about/ccdc-2012/.

22. Mukherjee, B., L. T. Heberlein and K. N. Levitt (1994). "Network intrusion detection." Network, IEEE **8**(3): 26-41.

23. Onwubiko, C., Owens, Thomas (2012). Situational Awareness in Computer Network Defence, IGI Global.

24. Roesch, M. (1999). Snort: Lightweight Intrusion Detection for Networks. LISA.

25. Rogers, M. K., J. Goldman, R. Mislan, T. Wedge and S. Debrota (2006). Computer forensics field triage process model. Conference on digital forensics, security and law.

26. Sangster, B., T. O'Connor, T. Cook, R. Fanelli, E. Dean, W. J. Adams, C. Morrell and G. Conti (2009). Toward instrumenting network warfare competitions to generate labeled datasets. Proc. of the 2nd Workshop on Cyber Security Experimentation and Test (CSET'09).

27. Sebastiani, F. (2002). "Machine learning in automated text categorization." ACM computing surveys (CSUR) **34**(1): 1-47.

28. Sourcefire. (2013). "Snort :: snort-rules." Retrieved 4th November, 2013, from http://www.snort.org/snort-rules/.

29. Sourcefire. (2013). "Snort Users Manual." Retrieved 1st November, 2013, from http://manual.snort.org/.

30. Sperotto, A., G. Schaffrath, R. Sadre, C. Morariu, A. Pras and B. Stiller (2010). "An overview of IP flow-based intrusion detection." Communications Surveys & Tutorials, IEEE **12**(3): 343-356.

31. Subbulakshmi, T., G. Mathew and S. M. Shalinie (2010). "Real Time Classification and Clustering Of IDS Alerts Using Machine Learning Algorithms." International journal of Artificial & Application **1**(1): 20.

32. Symantec Corporation (2013). "Symantec Internet Security Threat Report 2013." Symantec Internet Security Threat **18**: 1-36.

33. Tong, S. and D. Koller (2002). "Support vector machine active learning with applications to text classification." The Journal of Machine Learning Research **2**: 45-66.

34. West-Brown, M. J., D. Stikvoort, K.-P. Kossakowski, G. Killcrece and R. Ruefle (2003). Handbook for computer security incident response teams (csirts), DTIC Document.

35. Wilcoxon, F. (1945). "Individual comparisons by ranking methods." Biometrics bulletin **1**(6): 80-83.

36. Zhang, Z., J. Li, C. Manikopoulos, J. Jorgenson and J. Ucles (2001). HIDE: a hierarchical network intrusion detection system using statistical preprocessing and neural network classification. Proc. IEEE Workshop on Information Assurance and Security.

37. Zhang, Z. and C. Manikopoulos (2003). Investigation of neural network classification of computer network attacks. Information Technology: Research and Education, 2003. Proceedings. ITRE2003. International Conference on, IEEE.

Appendix A: Suspicious Event Taxonomy

The following table details the taxonomy that was used in the implementation of this capability, and provides descriptions for each classification.

Category	Parent	Description
Reconnaissance	N/A	An event in which an attempt is made to gain knowledge about the target network.
Exploitation	N/A	An event in which an attempt is made to exploits systems on a network.
Communication	N/A	An event in which an attempt is made to communicate between components on a network, or outside of it.
Scan	Infiltration	An event where an attempt is made to detect open ports.
Ping	Infiltration	An event where an attempt is made to find computers on a network.
Bad Code	Exploitation	An event due to an exploit caused by badly implemented code.
Buffer Overflow	Exploitation	An event due to an exploit caused by a buffer overflow.
Remote Access	Exploitation	An event due to running a command on a remote computer.
Authentication	Exploitation	An event involving authentication to a system.
Permissions	Exploitation	An event involving access to files which permissions are not held for.
Page Access	Exploitation	An event which involves accessing a web page or folder.
C2	Communication	An event which is triggered by a malicious application sending or receiving Command and Control messages.
Exfil	Communication	An event which is triggered by the exfiltration of data.
P2P	Communication	An event which is triggered by Peer to Peer Traffic.

DEFENCE SCIENCE AND TECHNOLOGY GROUP DOCUMENT CONTROL DATA	1. DLM/CAVEAT (OF DOCUMENT)

2. TITLE Text Classification of Network Intrusion Alerts to Enhance Cyber Situation Awareness and Automate Alert Triage	3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (U/L) NEXT TO DOCUMENT CLASSIFICATION) Document (U/L) Title (U) Abstract (U)
4. AUTHOR(S) Iain Dickson	5. CORPORATE AUTHOR Defence Science and Technology Group PO Box 1500 Edinburgh SA 5111

6a. DST GROUP NUMBER DST-Group-TN-1640	6b. AR NUMBER AR-016-883	6c. TYPE OF REPORT Technical Note	7. DOCUMENT DATE June 2017

8. OBJECTIVE ID AV14897636	9.TASK NUMBER N/A	10.TASK SPONSOR N/A	11. MSTC Systemic Protection and Effects	12. STC Automated Analytics and Decision Support

13. DOWNGRADING/DELIMITING INSTRUCTIONS	14. RELEASE AUTHORITY Chief, Cyber and Electronic Warfare Division

15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT

Approved for Public Release

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE, PO BOX 1500, EDINBURGH, SA 5111

16. DELIBERATE ANNOUNCEMENT

No Limitations

17. CITATION IN OTHER DOCUMENTS Yes

18. RESEARCH LIBRARY THESAURUS

Cyber Security, Artificial Intelligence

19. ABSTRACT

For many Cyber Security Incident Response Teams (CSIRT), reacting and responding to suspicious network activity is predominantly a manual task and lacks the necessary levels of automation required to deal with the volume of alerts. Alerts are signalled from tools such as Intrusion Detection Systems (IDS) to skilled analysts who must then decide on courses of action and remediation activities. The IDS alerts are basic; analysts must manually derive context about the alert using their prior knowledge.

In this paper, we describe Artificial Intelligence (AI) techniques used to automate the derivation of context from IDS alerts. We propose two algorithms based on well-known automated text classification methods and define a multi-level taxonomy to describe classifications of alerts in a semantically hierarchical manner. Consideration is given to the use of these algorithms by a CSIRT, as well as how Situation Awareness (SA) can be improved through automation. Our findings show that a combination of Naïve Bayes algorithms in conjunction with our proposed hierarchical taxonomy can automate alert classification with high accuracy, and low false and unclassified rates.