Combating Imbalance in Network Intrusion Datasets

David A Cieslak, Nitesh V Chawla, Aaron Striegel

***Abstract*— An approach to combating network intrusion is the development of systems applying machine learning and data mining techniques. Many IDS (Intrusion Detection Systems) suffer from a high rate of false alarms and missed intrusions. We want to be able to improve the intrusion detection rate at a reduced false positive rate. The focus of this paper is rule-learning, using RIPPER, on highly imbalanced intrusion datasets with an objective to improve the true positive rate (intrusions) without significantly increasing the false positives. We use RIPPER as the underlying rule classifier. To counter imbalance in data, we implement a combination of oversampling (both by replication and synthetic generation) and undersampling techniques. We also propose a clustering based methodology for oversampling by generating synthetic instances. We evaluate our approaches on two intrusion datasets — destination and actual packets based — constructed from actual Notre Dame traffic, giving a flavor of real-world data with its idiosyncrasies. Using ROC analysis, we show that oversampling by synthetic generation of minority (intrusion) class outperforms oversampling by replication and RIPPER's loss ratio method. Additionally, we establish that our clustering based approach is more suitable for the detecting intrusions and is able to provide additional improvement over just synthetic generation of instances.**

***Index Terms*— Computer Network Security, Imbalanced Datasets, Classification, ROC Curves**

I. Introduction

Network intrusion detection refers to the set of techniques used to isolate attacks against computers and networks. An Intrusion Detection System (IDS) thus detects hostile activities in a network. In addition to detection of attacks, such a system must prevent their malicious effects, or assist a human in a system or network administrator role in this prevention. By nature, even basic networks are very complex systems and the further evolution of the Internet has made it difficult to construct a total understanding of the system. However, work by Leland *et al.* suggests that local network traffic may contain complex, yet self-similar patterns [1]. Later, multi-fractal scaling was discovered and reported by Levy-Vehel *et al.* [2]. The results of the 1998 DARPA Off-line Intrusion Detection Evaluation indicated that further research should be performed focusing on techniques to find new attacks [3].

Data mining applications to network security can be broadly categorized into two sets, *anomaly detection* and *signature detection*. Anomaly detection constructs models of normal data and then detects deviations from this norm (anomalous logins, traffic to source port $>$ X), typically using outlier detection. The difficulty in this method is differentiating "normal" behavior from "abnormal" behavior causing these techniques to suffer from high false positive rates. However, there has been work in reducing the false positive rate by using multiple data streams [4]. Other avenues of false positive rate improvement includes hierarchical aggregation of specified portions of total activity [5]. Alternatively, the signature-based approach searches for specific patterns (strings) that denote suspicious behavior. While this approach is sufficient for known attacks, it becomes insufficient when the attack signature or normal traffic makeup is unknown. This breakdown leads signature-based approaches to suffer from a high false positive rate. Thus, it is imperative to design methods of classification with low false positive rates.

Going hand in hand is the compelling and inherent problem of learning such signature based classifiers from highly imbalanced network intrusion datasets. Typically, network intrusions and malicious behavior will represent a very small subset of all network traffic. However, their detection is highly critical for the health of a network. Hence, we cannot afford a high intrusion detection rate at the expense of false alarms, as it will lead to a loss of relevant packets. Thus, learning classifiers from such unbalanced datasets faces a number of relevant problems: improper classification evaluation metrics, absolute or relative lack of data, data fragmentation, improper inductive bias, and noise [6], in addition to the skew of accuracy and probability measures employed by the classifiers.

Axelsson [7] demonstrated that the primary limitation of an intrusion detection system is not the ability to identify behavior as intrusive, rather its effectiveness stems from its abilities to limit false alarms. Therefore, it is the purpose of this paper to study the effectiveness of several techniques to reduce false positives in intrusion datasets. We chose to construct our own "real-world" intrusion dataset by tapping the Notre Dame traffic to avoid the various limitations of the DARPA traffic [8]. Moreover, we constructed two different types of datasets — packet based and destination based. This allowed us to evaluate the efficacy of the approaches on datasets with different features and class distribution. We then used RIPPER [9] and several sampling methods to construct classifiers on these datasets.

Contribution: The main contributions of our work include a) effective evaluation and comparison of sampling methods, including oversampling by replication, SMOTE (Synthetic Minority Over-sampling TEchnique) [10], and undersampling, on real-world intrusion datasets; b) comparisons with RIPPER's loss ratio implementation for re-weighting the costs of false positives vs. false negatives and c) a clustering based implementation of SMOTE (*Cluster-SMOTE*) that further improves the performance over all the sampling methods.

Our hypothesis is that by generating synthetic intrusion cases to populate the dataset, particularly in the localized clusters, we will evoke the notion of "similarities" in network

Manuscript received January 20, 2006; revised February 4, 2006. This work was supported by the IEEE.

D. Cieslak, N. Chawla, and A. Striegel are with the Computer Science and Engineering Department, University of Notre Dame, Notre Dame, IN 46556 USA. Respective e-mails: *dcieslak, nchawla, striegel*@nd.edu

traffic. SMOTE generates new instances based on the "known" distribution, thus improving the generalization capacity of the learned classifier. SMOTE adds these instances in the space between minority examples, emphasizing the class border in favor of the minority class. To learn efficient discriminative learners it is important to emphasize on such class borders. To demonstrate our claim, we first compare SMOTE to other techniques using ROC curve analysis and demonstrate the success of its emphasis on class borders by comparison to a pure random replication oversampling method. In addition, we will also present a new technique of *Cluster-SMOTE*, which applies unsupervised learning to partition datasets into regions that will enable SMOTE to deliver enhanced results and we will present results indicating that this method may be used as an improvement over SMOTE.

The rest of the paper is organized as follows. Section 2 outlines the approach used to construct our datasets. Section 3 discusses the data mining approaches utilized in our study. Section 4 presents the experiments and Section 5 discusses results. Section 6 draws conclusions and discusses future work.

II. Data Extraction

Many efforts have used the DARPA'98 dataset for testing and training purposes. While this is a benchmark for intrusion detection methods, it has a number of shortcomings. The validity of DARPA'98 was questioned by McHugh [8] for its use of synthetic traffic for generating normal data and using attacks generated from scripts and programs. Additionally, the normal data does not contain natural but noisy traffic behavior such as packet storms or strange fragments. Ultimately, this dataset is not representative of contemporary network traffic. Thus, it was imperative to construct our own dataset based on a collection of contemporary network traffic. Many other approaches construct a data model based on network connections [11]–[15].

We operated using real network data from the University of Notre Dame. This network runs at 100 Mbps and is comprised of over 10,000 primarily residential computers, the majority of which run Windows. Traffic was collected during the summer of 2004. During this collection, a number of on-campus machines fell victim to a Trojan horse style attack and became "zombie" hosts in a distributed denial of service attack. Thus, the data sample used in our experiments represents a fairly "interesting" segment of network traffic.

In order to extract new classifiers for network intrusion detection, we must construct a dataset which further entails a packet labeling method must be elected and applied. Many stored network traffic files were processed using the SNORT open source intrusion detection system and received labeling based on the appropriate rule set [16]. This set makes a wide sweep on potential attacks such as viruses, port-scans, MYSQL attacks, and DoS and DDoS attacks. One limitation with this approach is that SNORT can only apply one label per packet, for instance SNORT performs telnet checking prior to DDoS; thus, a packet violating both rules will only reflect the attack for which it is first scanned. We can accommodate this limitation by focusing our approach and treating this as a 2-class problem. Rather than associating the type of intrusion with the packet, we merely label whether a packet was intrusive. This simplifies our approach in our packet analysis and allows for other flexibility. Using this method, we constructed two datasets: Packets and Destinations. While the Packets dataset measures the intrusiveness on a per packet basis, Destinations generates an aggregate of traffic.

A. Packets Dataset

Our first dataset comprised of collected packets and their SNORT labels. The set of attributes for each element denotes the packet's type, the status of its flags, values of other pertinent fields, and a basic summary of the packet's data payload. The set of characteristics summarizing the payload calculate the percentages of bytes representing printable and unprintable characters, as well as the percentage of digits, white space, punctuation, upper case, and lower case characters within the payload. Such a set of features truncates the total amount of information yielded by a packet capture and provides ample means to differentiate between attack and normal traffic. While the total number of packets studied will be significantly higher than that of the destinations set and the packet dataset should contain less noisy examples, accurate classifier construction may be even more problematic as the the imbalance ratio of this dataset is even less favorable than that of the destinations dataset, as can been seen in Table I. The packet dataset should therefore produce an even more compelling case for SMOTE and our *Cluster-SMOTE* method.

dataset	attributes	alerts	non-alerts	total size
Destinations	25	147	3933	4080
Packets	43	2106	344,514	346,620

TABLE I
DATA DISTRIBUTIONS IN THE DATASETS.

B. Destinations Dataset

We elected to construct another novel dataset. Instead of a connection model, we elected for the construction a destination-based model in which packets are organized by their destination. Attributes are constructed by noting the number, type, and rate of packets received in addition to the number of connections on a given host. Other attributes summarize the overall make-up of the packets themselves and standard deviation calculations are used to note high or low levels of variance in these characteristics. Thus, each dataset member represents a separate host that we have recorded as have received traffic and its attributes represent the composite of traffic received. We use SNORT to identify intrusion packets and label hosts receiving such packets as compromised. This will enable us to perform a survey of compromised and uncompromised hosts and to induce a set of rules for both groups. Studying destination traffic allows for a separate, host-based analysis that is useful to network managers and intrusion detection systems by isolating systems that are likely to have received attack traffic. While administrators are privy to a

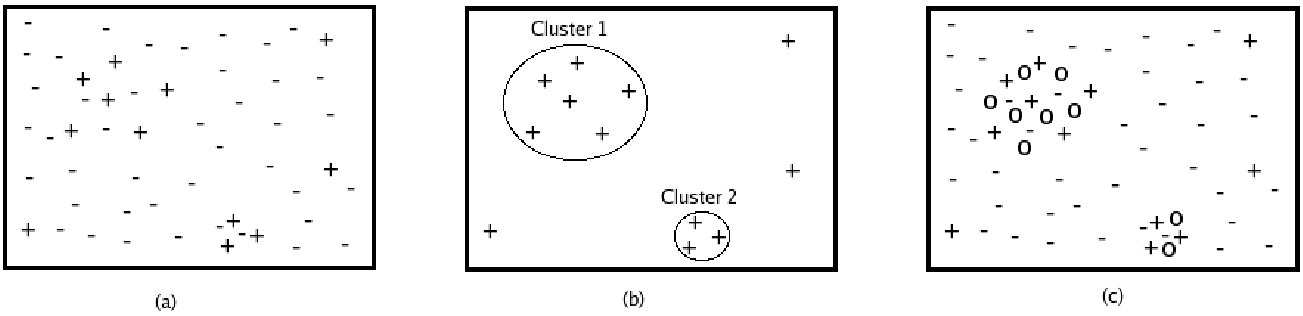

Fig. 1. The attribute space in (a) features a sparse majority class, a minority class region, and several minority outliers. In (b), *Cluster-SMOTE* detects two clusters of minority points and uses this information to generate new synthetic examples, as seen in (c)

substantial amount of traffic passed on their networks, it is unlikely that even this wealth of information is enough to supply number of examples required to accurately forecast, even via data-mining, the relatively rare case of intrusion. Thus, we anticipate that SMOTE will assist us by generating synthetic examples of compromised hosts, improving our classifier performance.

III. Mining For Signatures

The intrusion datasets are highly imbalanced in nature, as revealed in Table I. A dataset is imbalanced if the classification categories are not approximately equally represented. While our main goal is to correctly identify the intrusive instances based on the signatures, the classification techniques can be easily biased towards the majority class (non-intrusive). We are more interested in the trade-offs between true positives and false positives, that is how many false positives are potentially caused as we increase the intrusion detection rate. One can potentially configure the signature based system depending on the system specific trade-offs. We used SMOTE [10] and our proposed *Cluster-SMOTE* along with RIPPER [9] as our classification technique and each method is briefly outlined in the following subsections.

A. RIPPER

RIPPER is a fast, highly noise tolerant rule learner, originally targeting learning problems involving very large and noisy datasets [9]. While this application is used heavily in many, text-driven data-mining experiments, its noise tolerance makes it very useful in many other studies, such as our own. Thus, RIPPER will produce a set of rules outlining intrusive traffic and assume all other traffic to be non-intrusive. Comprehensibility of a classifier can be key for network security for post-analysis by a human expert.

B. SMOTE: Synthetic Minority Oversampling TEchnique

Sampling methods are very popular in balancing the class distribution before learning a classifier, which uses an error based objective function to search the hypothesis space. Over and under-sampling methodologies have received significant attention to counter the effect of imbalanced datasets [10], [17]–[20].

The random under and over-sampling methods have their various shortcomings. The random undersampling method can potentially remove certain important examples, and random oversampling by replication can lead to overfitting. Oversampling by replication can also lead to similar but more specific regions in the feature space as the decision region for the minority class. This can potentially lead to overfitting on the multiple copies of minority class examples.

To overcome the overfitting and broaden the decision region of the minority intrusion class cases, SMOTE can be used to generate synthetic examples by operating in "feature space" rather than in "data space" [10]. The minority class is oversampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general. For the nominal cases, we take the majority vote for the nominal value amongst the nearest neighbors. We use the modification of Value Distance Metric (VDM) [21] to compute the nearest neighbors for the nominal valued features.

The synthetic examples cause the classifier to create larger and less specific decision regions, rather than smaller and more specific regions, as typically caused by over-sampling with replication. More general regions are now learned for the minority class rather than being subsumed by the majority class samples around them. The effect is that classifiers generalize better. This generalization capacity of a classifier can be very pertinent for intrusion detection.

C. Cluster-SMOTE

Our intuition into the class imbalance problem is that having a small group of minority examples makes it difficult to establish proper class borders. Thus, the ability to correctly define the class regions and hence their borders would allow

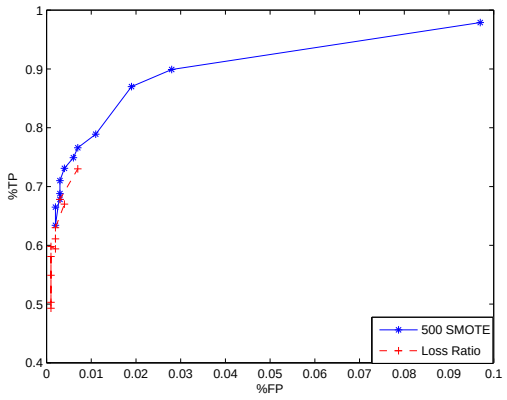

Fig. 2. ROC Curves depicting 500 SMOTE against RIPPER loss ratio on the Packets dataset

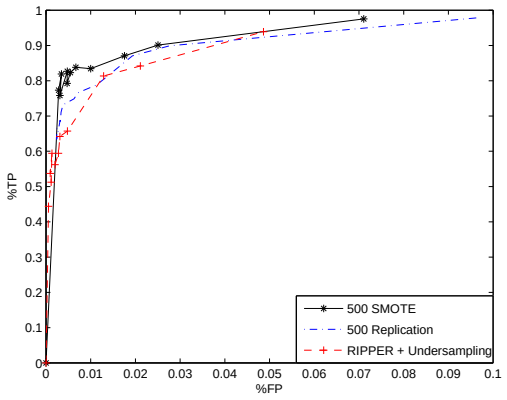

Fig. 3. ROC Curves depicting the optimal SMOTE and random replication methods applied on the Packets Dataset from rules learned by RIPPER.

for trivial classification. As these regions are unknown and even in the best cases may be impossible to deduce from given data, we believe only an approximation of these regions may be inferred. Even approximations may enhance classifier construction.

To develop these minority region approximations, we have applied simple k-means clustering to the set of minority examples in each dataset. We then apply SMOTE to each cluster and then reform the dataset by reinserting the set of original minority examples and the synthetic examples as well. This process allows for focused improvements on a localization basis for the minority class and should improve SMOTE's performance on imbalanced datasets. Leland's observation of self-similar patterns suggests that clustering enable the detection of such distinct patterns and that generating synthetic examples focusing on localizations will enhance global classification [1].

IV. Experiments

Our experiments demonstrate the success of SMOTE in improving the detection of intrusive packets and compromised hosts, with an acceptable relative increase in the rate of false alarms. We show the efficacy of SMOTE both when used globally and locally (*Cluster-SMOTE*). As SMOTE generates additional synthetic examples for the training set by emphasizing the alert and non-alert class borders, it was important to establish that SMOTE's contribution was this emphasis, rather than its creation of a more balanced dataset. Therefore, experiments were performed in which examples of the alert class were replicated at the same rate used by SMOTE and were likewise added to the training set. As this method improves the class imbalance ratio, we expect an improvement over an unaugmented training set. However, as replication does not emphasize the class borders, we expect SMOTE's performance to dominate.

To sweep an ROC curve, we undersampled the majority class by randomly removing a given percentage of majority examples from the training set. As with minority class oversampling via replication, random majority class undersampling improves classification performance by essentially emphasizing the minority class by reducing the ratio of majority to minority examples. The set $\{1\%MAJ, 5\%MAJ, 10\%MAJ, 20\%MAJ, ..., 100\%MAJ\}$ represents the sets of majority examples used in our experimental datasets. At each of the undersampling amounts, we applied SMOTE and oversampling with replication. This allowed us to generate ROC curves with sufficient points.

An alternative to SMOTE is adjusting RIPPER's loss ratio, L, which specifies the relative cost of a false positive against a false negative. Hence, a ratio $L < 1$ penalizes more heavily for missing minority examples, while a ratio $L > 1$ increasingly penalizes for false alarms. As opposed to the more complicated cost-sensitivity matrix which assigns point values to each of the four types of classification: true positives, true negatives, false positives, and false negatives, loss ratio provides a simpler progressive method for adjusting the minority class true positive rate. The effects of this adjustment on RIPPER in producing rules were compared against those of SMOTE.

A. ROC Curves

Receiver Operating Characteristic (ROC) Curves provide an effective basis for comparison between classifiers of imbalanced datasets by tracing the increase in the rate of false alarms as the classifier is tuned to increase the rate of correct alarms raised [22]. Visualization of ROC curves also enables an understanding of the interplay between the rates of generation of false and true positives. Depending on the nature of system, one can choose an operating point from the ROC curve. Thus, an ROC curve study is important to understanding SMOTE's effectiveness in these experiments.

We undersampled the majority class to generate ROC curves, with each undersampled point representing the rate of true positives against the rate of false positives for the learned classifier, yielding a single curve. Then, we oversampled (using SMOTE, replication, and *Cluster-SMOTE*) for each of the undersampled dataset. This generated a separate set of ROC curves for comparison. The best classifiers are represented by those points closest to the upper-left corner of the graph, the optimal point representing perfect minority detection with no false alarms raised. There was an additional

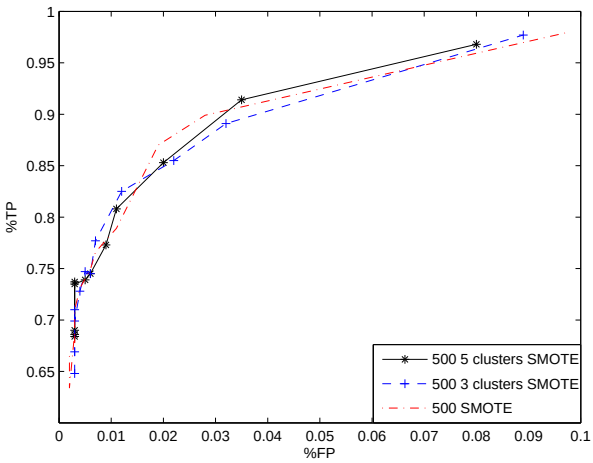

Fig. 4. ROC Curves comparing the performance of SMOTE and *Cluster-SMOTE* on Packets using rules learned by RIPPER.

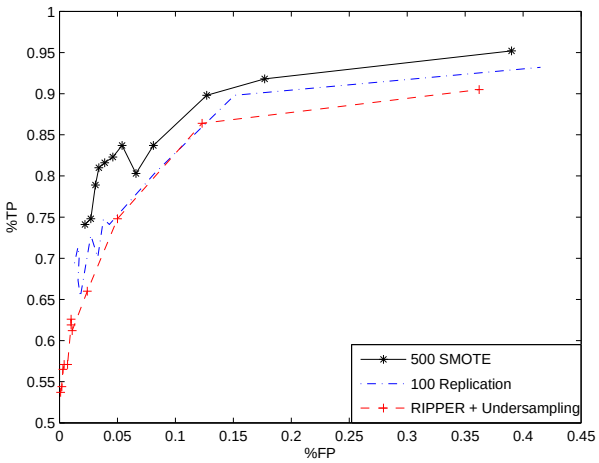

Fig. 5. ROC Curves showing the performance of SMOTE and random replication on the Destinations dataset using RIPPER.

group of experiments studying the effects of varying the loss ratio of Ripper from 1 to .01 used in classifier construction.

B. Training and testing sets

The task of classification requires two separate datasets: one for training and one for testing. The learning algorithm generates rules on the training dataset and its performance is measured by its classification results on the testing set. Given the very large size of the packets dataset, we randomly split the dataset with 75% of the examples used in training and 25% used for testing, while maintaining the original class distribution in both the sets. Considering the low number of alerts in the destinations dataset, a simple split method was insufficient, as it would have led to a very small amount of alerts for training and testing. Thus, we performed a 10-fold cross-validation and averaged the true positive rate and false positive rate across the 10 folds.

V. Results

The first experiment established a baseline by comparing SMOTE's performance against that of RIPPER's loss ratio. Figure 2 displays the results of this comparison on the packets dataset. The loss ratio's sudden halt precludes its use in cases where more generous false alarm rates are allowed. Additionally, the SMOTE curve dominates the loss ratio curve from start to finish. Similar observations were made for the same experiment on the destinations dataset. Therefore, SMOTE is a more effective method than varying loss ratio and our experiments further validate SMOTE as an effective means to combat intrusion.

A. Packets

As seen in Figure 3, adding a layer of oversampling using SMOTE and replication adds value to the prediction of the alert cases. It is important to note that the SMOTE curves clearly dominate all the ROC curves. Thus, the additional emphasis on the class borders placed by SMOTE must be effective in generating a superior AUC and thus allows RIPPER to generate a better classifier.

Our results indicate that high levels of SMOTE have improved ROC curves on both datasets. However, we can improve results further by applying our outlined *Cluster-SMOTE* technique. We applied simple k-means clustering to the minority examples of the packets dataset and performed two experiments using three and five centroids, values seeded by the user. Oversampling varied at the rates in the previous experiments on individual clusters and RIPPER was applied to the entire dataset. Figure 4 depicts some of the best results along with a 500-SMOTE baseline for comparison purposes.

Among these curves, there is no single classifier that clearly dominates. However, it is possible to fit a convex hull to the curves presented on an ROC graph [23]. This allows for the approximation of the optimal classifier, based on known classifiers. When this procedure is applied to Figure 4, all but one point within this convex hull come from classifiers based on the *Cluster-SMOTE* method. Therefore, the optimal classifier for a given acceptable false-positive rate will be selected from *Cluster-SMOTE* classifiers in all but one case. Hence, *Cluster-SMOTE* is shown to be an effective method of classifier enhancement on the packets dataset.

B. Destinations

Likewise, the effectiveness of the SMOTE technique is demonstrated through our destinations dataset experiments. The visual trade-off between true positives and false positives is presented as an ROC in Figure 5. Given the unstable nature of the rules formed by RIPPER, there will tend to be certain points in the ROC space that will slightly deviate from the trend, but in general, the trend of the curves is as one might expect. Another deviation is that the best ROC curve for replication was at 100% level, indicating that more replication led to severe overfitting. However, SMOTE successively improved performance as we added more synthetic examples, giving best

performance at 500%. As this dataset features an extremely low number of alert examples, the ROC curves produced are especially prone to drastic sudden changes, as can be seen in Figure 5. However, the same general observations hold from the destinations dataset as they did from packets: the SMOTE curve dominates the random replication curve. As this dataset contained a very small minority class and clustering on these examples yielded a single cluster, this indicates that the alerts within destinations fall within a compact region of the total feature space; thus, *Cluster-SMOTE* would be unable to generate improved classifiers on this dataset.

VI. Conclusion

Efficient intrusion detection is a difficult problem because of the difficulty inherent in identifying intrusive behavior while maintaining the ability to limit false alarms. Thus, we have conducted an investigation into methods of false positive limitation. We began by outlining a procedure for building a dataset from collected network traffic. Using basic payload analysis and destination address grouping, we generated two imbalanced datasets featuring a large set of attributes. Using an open source IDS, SNORT, we were able to label examples from each dataset as alert and normal.

These datasets were then used in our investigation of applications of SMOTE and a new method, *Cluster-SMOTE*, in terms of restricting false positive rates while generating rules using RIPPER. ROC curves were presented establishing that SMOTE's emphasis on class borders in rule learning improves classifiers beyond the level of class balance restoration through simple random minority example replication. These experiments held true through both datasets. An additional experiment on the packets dataset demonstrated SMOTE's effectiveness over RIPPER loss ratio.

In addition, we have investigated the effectiveness of *Cluster-SMOTE* as a technique for imbalanced class learning above SMOTE within the scope of these datasets. We have concluded that *Cluster-SMOTE* provides an improvement on SMOTE for the packets dataset, but cannot be used on destinations due to the limited feature space size of the minority class. Our success with the packets dataset indicates that *Cluster-SMOTE* is a technique which may be useful in application settings outside of intrusion detection, but within the set of class imbalance problems.

Looking forward, we will further pursue *Cluster-SMOTE* through a thorough examination of its effectiveness on other datasets. Additionally, the simple k-means method employed in this paper is at best limited in its effectiveness and it is our goal to develop superior localizations of the minority class through an approach using hyper-rectangles. In conjunction with this investigation of methods for localizing class imbalanced datasets, we will investigate new methods of synthetic point generation that yield superior class boundary definition. This in turn will enable us to strike at the heart of the class imbalance problem: class region definition.

Acknowledgment

We would like to thank Chad Mano for his work in collecting network data and for allowing us to use this data for the experiments outlined in this paper.

References

[1] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, 1994.

[2] J. Vehel, E. Lutton, and C. Tricot, *Fractals in Enginnering: From Theory to Industrial Applications.* New York:Springer-Verlag, 1997.

[3] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "The 1998 DARPA Off-line Intrusion Detection Evaluation," in *Proc. of DARPA Information Survivability Conference and Exposition (DISCEX)*, Los Alamitos, CA, January 2000, pp. 12–26.

[4] M. Roughan, T. Griffin, Z. Morley Mao, A. Greenberg, and B. Freeman, "IP Forwarding Anomalies and Improving their Detection Using Multiple Data Sources," in *Proc. of ACM SIGCOMM 2004 Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality*, August 2004.

[5] Y. Zhang, S. Singh, S. Sen, N. Duffield, and C. Lund, "Online Identification of Hierarchical Heavy Hitters: Algorithms, Evalutaion, and Applications," in *Proc. of ACM SIGCOMM Internet Measurement Conference*, October 2004.

[6] G. Weiss, "Mining with Rarity: A Unifying Framework," *SIGKDD Explorations*, vol. 6, no. 1, pp. 7–19, 2004.

[7] S. Axelsson, "The Base-Rate Fallacy and the Difficulty of Intrusion Detection," *Information and System Security*, vol. 3, no. 3, pp. 186–205, 2000.

[8] J. McHugh, "The 1998 Lincoln Laboratory IDS Evaluation (A Critique)," in *Proc. of the Recent Advances in Intrusion Detection*, October 1998, pp. 145–161.

[9] W. Cohen, "Fast effective rule induction," in *Proc. of Machine Learning: the 12th International Conference*, July 1995, pp. 115–123.

[10] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "SMOTE: Synthetic Minority of Over-sampling TEchique," *Journal of Artificial Intelligence Resarch*, vol. 16, pp. 341–378, 2002ta Mining for Network Intrusion. Detection: How to Get Started, MITRE Techni.

[11] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," in *Proc. of 3rd SIAM Conference on Data Mining*, January 2003.

[12] M. Thottan and C. Ji, "Anomaly Detection in IP Networks," *IEEE Transactions on Signal Processing*, vol. 51, pp. 2191–2204, 2003.

[13] W. Lee and S. Stolfo, "Data Mining Approaches for Intrusion Detection," in *Proc. of the 7th USENIX Security Symposium*, January 1998.

[14] E. Bloedorn, A. Christiansen, W. Hill, C. Skorupka, L. Talbot, and J. Tivel, "Data Mining for Network Intrusion Detection: How to Get Started," Mitre, Tech. Rep., 2001.

[15] H. Javitz and A. Valdes, "The NIDES Statistical Component: Description and Justification," Computer Science Lab, SRI Interational, Tech. Rep., 1994.

[16] Sourcefire, *Snort Users Manual: The Snort Project*, 2005.

[17] N. Japkowicz, "The Class Imbalance Problem: Significance and Strategies," in *Proc. of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning*, vol. 1, 2000, pp. 111–117.

[18] G. Weiss and F. Provost, "Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[19] N. V. Chawla, N. Japkowicz, and A. Kolcz, "Editorial: Special Issue on Learning from Imbalanced Data sets," *SIGKDD Explorations*, vol. 6, no. 1, pp. 1–6, 2004.

[20] J. Laurikkala, "Improving Identification of Difficult Small Classes by Balancing Class Distribution," University of Tampere, Tech. Rep. A-2001-2, 2001.

[21] S. Cost and S. Salzberg, "A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features," *Machine Learning*, vol. 10, no. 1, pp. 57–78, 1993.

[22] J. Swets, "Measuring the Accuracy of Diagnostic Systems," *Science*, vol. 240, pp. 1285–1293, 1988.

[23] Provost, F. and Fawcett, T., "Robust Classification for Imprecise Environments," *Machine Learning*, vol. 42, no. 3, pp. 203–231, 2001.