Improved Classification of Known and Unknown Network Traffic Flows using Semi-Supervised Machine Learning

Timothy Glennan, Christopher Leckie, Sarah M. Erfani

Department of Computing and Information Systems,
The University of Melbourne, Australia
{tglennan@student., caleckie@, sarah.erfani@}unimelb.edu.au

Abstract. Modern network traffic classification approaches apply machine learning techniques to statistical flow properties, allowing accurate classification even when traditional approaches fail. We base our approach to the task on a state-of-the-art semi-supervised classifier to identify known and unknown flows with little labelled training data. We propose a new algorithm for mapping clusters to classes to target classes that were previously difficult to classify. We also apply alternative statistical features. We find our approach has an accuracy of 95.10%, over 17% above the technique on which it is based. Additionally, our approach improves the classification performance on every class.

1 Introduction

Network traffic classification is an important task for a range of network-related areas, including network management, surveillance, and security. Traffic classification has traditionally been performed by inspecting port numbers. However, this is often ineffective due to the number of applications using non-unique and non-standard port numbers [1]. Deep-packet inspection avoids reliance on port numbers, but demands an up-to-date database of application signatures and has significant computational complexity, often making the approach unfeasible for real-world use [2]-[3].

Machine learning techniques have been gaining popularity for their ability to effectively classify network applications using only statistical flow features [1]-[3] and without the drawbacks of more traditional approaches. The open problem we address is how to improve the accuracy of traffic classification from applications that have been difficult to classify using only statistical traffic flow properties.

In this paper, we apply a semi-supervised machine learning technique to automatically identify network applications using only statistical traffic flow properties. Our approach is based on a leading semi-supervised traffic classification approach [4], which can handle flows generated by unknown applications. We propose two innovations to this method in order to further increase its effectiveness. First, our approach introduces an alternate algorithm for identifying applications, Second, we propose introducing feature selection into the system model. Based on an empirical evaluation on a standard benchmark dataset, we show that our approach has an accuracy of

95.10%, an increase of over 17% against the technique on which it is based [4]. Additionally, our approach improves the classification performance on every class.

2 Related Work

Current research into traffic classification has shown various supervised, unsupervised, and semi-supervised machine learning techniques to be viable approaches. Supervised machine learning approaches [5], [6] have been shown to achieve particularly high classification effectiveness. However, these approaches can only predict predefined classes found in the training data. Unsupervised learning approaches [7]-[8] classify from clusters of unlabelled training flows. While using unlabelled data means they can handle known and unknown classes, mapping clusters to classes remains a key challenge.

Semi-supervised approaches aim to address the problems of both supervised and unsupervised approaches. Erman *et al.* [2] developed an effective semi-supervised approach for classifying network applications, combining K-Means clustering with probabilistic assignment. Using a small set of labelled flows with a larger unlabelled set, clusters with labelled flows can automatically be mapped to classes. Clusters without labelled flows represent unknown classes. The key advantage of this technique is simple class mapping and handling of unknown classes. With few labelled instances, however, clusters are often incorrectly labelled "unknown". A recent extension to this approach by Zhang *et al.* [4] countered this weakness by automatically extending the labelled portion of training data. This was done by identifying correlated flows – flows sharing the same destination IP address and port, and protocol – and sharing labels between them. This approach was shown to significantly increase the labels available and thus better label clusters. Furthermore, applying compound classification to correlated test flows further improved effectiveness. It was shown to outperform standard and state-of-the-art machine learning algorithms, including decision trees, K nearest neighbours, Bayesian networks, and the Erman *et al*. approach.

While the Zhang *et al.* approach is a leading semi-supervised approach for traffic classification, certain traffic classes still proved challenging to identify. We aim to target these classes for an overall more consistently effective classifier.

3 Problem Statement

We are given a set of training flows $T = \{t_i \mid i = 1, 2, \ldots n\}$ and a set of testing flows $X = \{x_j \mid j = 1, 2, \ldots m\}$, generated on a single network. Each flow represents a bidirectional series of packets between two hosts, sharing the same source and destination addresses, port numbers, and protocol. Each flow has been generated by some known or unknown traffic class c. For each known class c, a subset of T exists such that $T_c = \{L_c \cap U_c\}$ and $\| L_c \| << \| U_c \|$, where L_c is the set of pre-labelled flows of class c and U_c is the set of unlabelled flows of class c. For any unknown class c, the subset of T containing flows of class c is $T_c = \{U_c\}$. That is, none of its flows are pre-labelled.

From T, we aim to create classifier $f(\mathbf{x}) = c$ such that when a flow $\mathbf{x}$ is given, a traffic class c is predicted. The traffic class c indicates that flow $\mathbf{x}$ was generated by a specific known class, or that it was generated by some unknown classes.

4 Our Proposed Approach

Figure 1 illustrates the details of our approach. The flow label propagation algorithm is first applied to a large training set containing a small number of labelled flows per class. The flow label propagation algorithm uses the correlated flows property of network traffic described in Section 2 to automatically increase the number of labelled flows. Feature selection algorithms are then applied to this larger labelled set to identify the strongest features. Next, clustering is performed on all training data, and then labelled flows are used to identify clusters as classes. Finally, the nearest cluster classifier predicts flow classes.

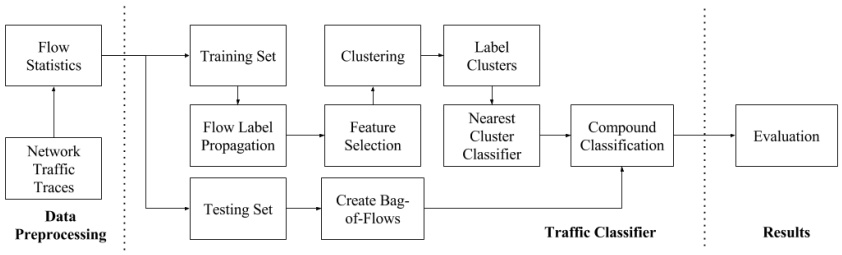

Fig. 1. System Model.

This system model is based on Zhang *et al.* [4], with some key alterations. Like [4], this model's main advantage is its ability to appropriately handle flows generated by unknown applications. Creating and identifying "unknown" clusters achieves this. However, we propose an alternative cluster labelling algorithm for increased effectiveness. After flow label propagation, we also introduce feature selection to identify a stronger feature set. Label propagation can greatly increase the amount of labelled data available, allowing feature selection algorithms to work more effectively. Thus this step can again increase the classification success. Below we describe our alternative cluster labelling algorithm, followed by our feature selection approach.

4.1 Fuzzy Cluster Labelling Algorithm

The cluster labelling algorithm introduced below is our proposed alternative to the algorithm used in [2], [4]. Their algorithm is a simple majority vote; the label for some cluster i is the most common label in i. If i has no labelled flows, then it is an unknown cluster. We follow the same principle, but our algorithm has two key differences. First, "unknown" is treated as a traffic class. Second, clusters can be labelled as multiple traffic classes. For this reason we dub the algorithm *fuzzy cluster labelling*.

```
Input: training flows T; set of k clusters trained on T
Output: traffic class labels, labels_i, for each cluster c_i
for i = 1 ← k
    c_ij = number of flows labelled as class j in cluster c_i
    labels_i = [argmax_j(c_ij)]
    foreach traffic class j
      if j not in labels_i and c_ij * threshold > y:
        append j to labels_i
```

Algorithm 1. Fuzzy Cluster Labelling

The algorithm requires a reasonable number of pre-labelled flows per class, which is achieved in our model by first applying the label propagation from [4]. The threshold ensures we assign additional cluster labels in the case of no clear majority. Otherwise we give it just one label. The labels are then naturally decided between during compound classification. The compound classification stage classifies all correlated test flows together via a majority vote of class labels. Using this algorithm, each test flow can therefore vote for multiple potential classes.

4.2 Feature Selection

Irrelevant or unnecessary features can negatively impact the success of machine learning algorithms [9]. Thus, feature selection methods aim to reduce the feature set to the most relevant subset. For classifying network flows, it is standard for statistical features to be used [3]. However, in our semi-supervised context, we have too few pre-labelled flows for feature selection to be effective. This problem is alleviated by first applying flow label propagation to the dataset. Once this is applied, there is a more reasonable pool of labelled data for feature selection algorithms to use. We reduce an initial set of 40 statistical features by applying the extra trees classifier algorithm [10], selected for its efficiency and simplicity, to identify a feature subset.

5 Experimental Evaluation

This section evaluates our proposed method against the Zhang method on which it is based, as this method has been shown to outperform other standard and state-of-the-art approaches.

5.1 Data Set Description

Table 1. Traffic class breakdown in the sample of the *wide* dataset used.

Traffic Class	# of Training Flows	# of Testing Flows
HTTP	24,000	6,000
BitTorrent	2,448	613
DNS	24,000	6,000

SMTP	24,000	6,000
SSH	24,000	6,000
HTTPS	15,370	3,843

The data used in this experiment originates from a publicly available *wide* (http://mawi.wide.ad.jp/mawi/) network traffic trace. The data used is a sample from traffic captured in March 2008. NetMate [11] is used to convert packets into flows and compute various features. This dataset was then separated into a training set of approximately 114,000 flows and a testing set of approximately 28,500 flows. While we acknowledge identifying ground truth classes through standard port numbers will introduce some error, this is a common labelling approach used in the literature, and the error introduced is expected to be small [12]. A maximum of 24,000 training flows and 6,000 testing flows were selected at random per class to prevent over-representation. Table 1 shows a complete breakdown of classes used.

5.2 Evaluation Metrics

Two standard metrics are used to evaluate the performance of the proposed method. The first method is accuracy, i.e., the number of correctly classified flows out of all classifications made. This metric is used to evaluate overall classifier performance.

$$Accuracy = \frac{Correctly\ Classified\ Flows}{Total\ Number\ of\ Flows} \quad (1)$$

The second metric used is F-measure, i.e., the weighted harmonic mean of precision and recall. Precision is defined as the ratio of flows correctly classified as a class to all flows classified as that class. Recall is defined as the ratio of flows classified as some class to all flows truly belonging to that class.

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2)$$

The F-Measure is used to evaluate the performance for each class individually.

5.3 Experimental Setup

For each experiment, we use 100 pre-labelled flows per known traffic class (HTTP, BitTorrent, SSH, and HTTPS). We select DNS and SMTP as unknown classes with no pre-labelled flows. We use k-Means as our clustering algorithm. The number of clusters for both Zhang's method and the proposed method is set to $k = 500$, and each experiment is repeated 5 times with results averaged. The large k chosen is appropriate since using a large number of clusters has been shown to result in pure clusters for network traffic [8], and the Zhang *et al.* method has been shown to be robust when varying the number of clusters [4]. The features used in our implementation of the Zhang approach are 20 statistical features described in [4].

5.4 Results of Fuzzy Cluster Labelling

The results of the fuzzy cluster labelling algorithm (with a threshold of 2.5) against the original Zhang *et al.* labelling can be seen in Figure 2. The same statistical features from [4] were used in both experimental setups. The labelling threshold parameter was varied between 2.0 and 3.0 and the impact was largely negligible.

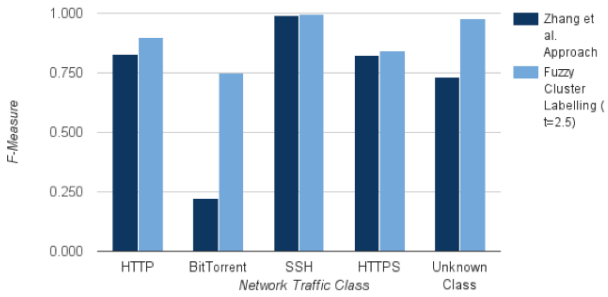

Fig. 2. F-Measure per traffic class when applying alternate cluster labelling methods.

Our proposed labelling algorithm resulted in an increase in F-Measure for every class. For classes where the Zhang approach performed well, there was always a slight, albeit sometimes insignificant, improvement. For example, the algorithm produced an increase in F-Measure of just 0.071 and 0.021 for HTTP and HTTPS classes respectively. For classes where the Zhang approach did not perform as well, our algorithm made more noticeable improvement. The unknown class improved from 0.733 to 0.980, an increase of 0.247. The BitTorrent class improved from 0.222 to 0.750, an increase of 0.528. We note that the BitTorrent class performed much better in [4] than in our implementation of the Zhang's approach. We attribute this to using different samples of the same dataset and having few training and test instances for this class.

5.5 Results of Fuzzy Labelling and Feature Selection

Table 2. Final feature set used after feature selection.

Feature Category	Description	# Of Features
Bytes (Forwards)	Minimum, maximum, and standard deviation of packets.	3
Bytes (Backwards)	Mean, maximum, and standard deviation of packets.	3
Inter Packet Time (Forwards)	Minimum, mean, maximum, and standard deviation of inter packet time in the forward direction.	4
Inter Packet Time (Backwards)	Mean, maximum, and standard deviation of inter packet time in the reverse direction.	3
Duration	Duration of the flow.	1
Flag	Whether there was a PSH flag in the forward direction.	1
Headers	Total size of the headers in each direction.	2

Applying feature selection reduced an initial set of 40 statistical features to the 17 described in Table 2. Applying both the new feature subset and the proposed cluster-

ing algorithm together completes our approach. The combined impact can be seen in Figure 3. Figure 3(a) shows the overall accuracy found is an increase from 77.77% to 95.10%, a significant increase of over 17% against [4].

The effect on F-Measure in Figure 3(b) shows that our approach improved the F-Measure for each class when again compared against [4]. The HTTP class increased by 0.087 to an F-Measure of 0.913. The F-Measure for the SSH class was 0.997, and 0.860 for HTTPS. These rose by a very minor 0.002 and 0.043 respectively. The unknown class grew from 0.733 to 0.980, and BitTorrent from 0.225 to 0.821.

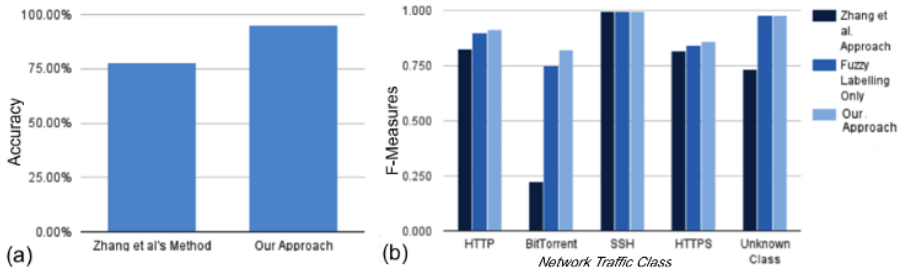

Fig. 3. Overall accuracy and F-Measures of the Zhang *et al*. approach against our approach.

Using our alternative feature improved only marginally over the Zhang feature set. However, each class performed as well or better than before. Most significantly, the BitTorrent class grew by a further 0.070. The HTTP and HTTPS classes found minor improvements of 0.013 and 0.016 respectively. The other classes remained as before.

6 Analysis and Discussion

The results in Section 5 demonstrate that our approach can significantly improve traffic classification effectiveness against a state-of-the-art method. The overall accuracy improvement of over 17% demonstrates the potential of our approach.

The proposed fuzzy cluster labelling algorithm made the most significant impact. There are two reasons for this. First, the Zhang approach ignores unlabelled flows when labelling, while we make use of them. Many of the unlabelled instances are truly of the unknown class, hence our cluster labelling accounts for this. Otherwise there is strong bias towards known classes, even when clusters are overwhelmingly unknown. While this incorrectly treats some unlabelled known class flows as unknown, we counter this error with label propagation, multiple labels, and compound classification. The second reason for improvement is to allow multiple labels per cluster. The labelling method in [2], [4] would label entire clusters based on its most common labelled class. However, there are circumstances when it does not make sense to apply this method. While we expect pure clusters in this domain with a large k [8], a brief analysis showed some clusters had as low as 35% purity. In these cases, majority labelling fails to represent the cluster, and thus explains why multiple labels allow such improvement. Our results show that a good choice of threshold can improve the performance of every class. This parameter ensures that pure clusters confi-

dently vote once, while less pure clusters are given multiple class votes. The classes that were already classified effectively remained successful. Meanwhile, classes that were previously frequently mislabelled exhibited more significant improvement. Additionally, fuzzy clustering labelling is seen as efficient in terms of computational complexity. Let n represent the number of flows in a cluster, and c represent the number of traffic classes. The total time complexity for our labelling algorithm is thus $O(n + c)$. There are typically very few classes c compared to flows n. Thus, this is approximately equivalent to the $O(n)$ of the method from [2], [4].

7 Conclusion

This paper presented a new take on an existing semi-supervised approach for network traffic classification. An overall accuracy of approximately 95% demonstrated the effectiveness of our approach to traffic classification. Furthermore, an improvement in F-Measure for every class demonstrated the effectiveness of fuzzy cluster labelling. This allowed our approach to consistently outperform the state-of-the-art method on which it is based. The alternative feature set considered also demonstrated how stronger feature subsets could be considered to further improve effectiveness.

References

1. Karagiannis, T., Broido, A., Faloutsos, M.: Transport layer identification of P2P traffic. In ACM SIGCOMM Conference on Internet Measurement. pp. 121-134 (2004).
2. Erman, J., et al.: Offline/realtime traffic classification using semi-supervised learning. Performance Evaluation. Vol. 64(9), pp. 1194-1213 (2007).
3. Williams, N., Zander, S., Armitage, G.: Evaluating machine learning algorithms for automated network application identification. Center for Advanced Internet Architectures (CAIA), Technical Report B, 60410. (2006).
4. Zhang, J., Chen, C., Xiang, Y., Zhou, W., Vasilakos, A. V.: An effective network traffic classification method with unknown flow detection. IEEE Transactions on Network and Service Management,. Vol. 10(2), pp. 133-147 (2013).
5. Erman, J., et al.: Offline/realtime traffic classification using semi-supervised learning. Performance Evaluation. Vol. 64(9), pp. 1194-1213 (2005).
6. Auld, T., Moore, A. W., Gull, S. F.: Bayesian neural networks for internet traffic classification. IEEE Transactions on Neural Networks. Vol. 18(1), pp. 223-239 (2007).
7. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow clustering using machine learning techniques. In PAM. pp. 205-214 (2004).
8. Erman, J., Arlitt, M., Mahanti, A.: Traffic classification using clustering algorithms. In SIGCOMM Workshop on Mining Network Data. pp. 281-286 (2006).
9. Nguyen, T. T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Comm. Surveys and Tutorials, Vol. 10(4), pp. 56-76 (2008).
10. Scikit-Learn.: http://scikit-learn.org/stable/modules/ensemble.html (as of March 2016)
11. NetMate.: http://sourceforge.net/projects/netmate-meter/ (as of March 2016).
12. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review. Vol. 36(5), pp. 5-16 (2006)