Exploring Unsupervised Learning Methods for Automated Protocol Analysis

Arijit Dasgupta
Mechanical Engineering
National University of Singapore
Singapore, Singapore
arijit.dasgupta@u.nus.edu

Yi-Xue Yan
Electrical Engineering
Nanyang Technological University
Singapore, Singapore
yany0025@e.ntu.edu.sg

Clarence Ong
Data Science and Analytics
National University of Singapore
Singapore, Singapore
clarence_ong@u.nus.edu

Jenn-Yue Teo, Bugsy
Electronic Systems Division
DSO National Laboratories
Singapore, Singapore
tjennyue@dso.org.sg

Dr Chia-Wei Lim, Andrew
Electronic Systems Division
DSO National Laboratories
Singapore, Singapore
lchiawei@dso.org.sg

***Abstract*—The ability to analyse and differentiate network protocol traffic is crucial for network resource management to provide differentiated services by Telcos. Automated Protocol Analysis (APA) is crucial to significantly improve efficiency and reduce reliance on human experts. There are numerous automated state-of-the-art unsupervised methods for clustering unknown protocols in APA. However, many such methods have not been sufficiently explored using diverse test datasets. Thus failing to demonstrate their robustness to generalise.**

This study proposed a comprehensive framework to evaluate various combinations of feature extraction and clustering methods in APA. It also proposed a novel approach to automate selection of dataset dependent model parameters for feature extraction, resulting in improved performance. Promising results of a novel field-based tokenisation approach also led to our proposal of a novel automated hybrid approach for feature extraction and clustering of unknown protocols in APA.

Our proposed hybrid approach performed the best in 7 out of 9 of the diverse test datasets, thus displaying the robustness to generalise across diverse unknown protocols. It also outperformed the unsupervised clustering technique in state-of-the-art open-source APA tool, NETZOB in all test datasets.

***Index Terms*—unsupervised learning, automated protocol analysis, protocol feature extraction and clustering.**

I. Introduction

Communication protocols are a predefined set of rules that multiple parties use at different OSI layers to communicate consistently. Despite there being many open-standards protocols (e.g. TCP, IP & 802.11), there are still numerous proprietary unknown protocols owned by companies & organizations. Therefore, there is the need for Protocol Analysis (PA) to infer detailed specifications of unknown protocols for network resource management, IoT interoperability, network protocol security audit, simulation and conformance testing [1]. Furthermore, the ability to analyse and differentiate network protocol traffic at routers (especially those of unknown protocols) is vital for effective network resource management by Telcos for differentiated Quality of Service (QoS).

This paper focuses on PA via Static Traffic Analysis based on analysis of captured network traffic of unknown protocols. This is a two stage process, where: 1) Vocabulary inference involves understanding the protocol messages, and 2) Grammar inference involves understanding the protocol predefined set of rules. Vocabulary inference involves clustering protocol messages into smaller and similar groups for subsequent inference of protocol field boundaries, relationships and semantics.

Traditionally, PA is done manually by experts and is very time-consuming, taking months or even years, with the additional challenges of having to recruit, train and retain such experts. Therefore, the need for APA was first raised in [2], with proposed approaches inspired by disciplines such as Bioinformatics and Natural Language Processing (NLP), due to the likeness in sequential semantics derived from byte sequences in protocol packets [3]–[5]. This has significantly improved the efficiency of the analysis process and even reduce the reliance on human experts. Today, NETZOB [5] is the most comprehensive open-source APA framework that utilizes a bioinformatics-inspired method for protocol message clustering [6] and is our baseline for comparison.

This paper focuses on automated feature extraction and clustering of packets with similar message formats in the vocabulary inference stage. Once this is done, the human expert analyst can analyse packets in each cluster more easily and efficiently. Hence, our framework aids the human expert analyst in PA and provides a crucial and early step affecting subsequent stages of the APA pipeline. We assume no prior knowledge and explore various unsupervised methods. The key contributions are as follows:

1) Developed a comprehensive APA framework for evaluation of various combinations of state-of-the-art unsupervised feature extraction & clustering methods.
2) Proposed novel methods for automated model optimisation for APA and developed greater insights into techniques for automatic field-based tokenization.

978-1-7281-9048-8/21/$31.00 ©2021 IEEE

3) Comprehensive experimentation of unsupervised automated features extraction and unknown protocol message clustering for APA, leading to an improved hybrid approach over state-of-the-art open-source APA tool, NETZOB and other related works.

Section II presents related works, Section III presents our proposed APA framework and methods, Section IV describes the experiment methodology, Section V discusses our experiment results and finally Section VI concludes the paper.

II. Related Works

Related works have typically focused on inferring message format types from packets of a single unknown protocol [5] - [7], using feature extraction and clustering techniques such as sequence alignment [5] and information bottleneck [8]. Today, there are hundreds of different protocols and it is naive and limiting to assume that a stream of unknown packets belong to a single protocol. Hence, our proposed framework (in Section III) aims to automatically differentiate both unknown protocol and message format types via distinguishing features to facilitate further analysis in later APA stages.

Previous works have also typically used information from the entire packet for feature extraction [5] - [7]. However, only the header of protocol packets usually contain information with relevance to the protocol's operation. Hence, it is desirable to perform feature extraction on only the packet header. Faced with an unknown protocol, there is no information on the length of this header portion. Therefore, our framework introduces a novel method that aims to infer the header length of the unknown protocol that yields the most amount of useful information for differentiating unknown protocols.

To extract features of an unknown protocol, sequence alignment techniques from bioinformatics and NLP have been employed due to the similarities in structure of DNA sequences and packets in network traces, and the textual nature of packet contents. Bossert et al. proposed NETZOB [5], which uses Needleman-Wunsch Sequence Alignment (NWSA) from bioinformatics to infer message formats and cluster protocols & message types, while Discoverer [2] uses tokenisation, recursive clustering and merging clusters to do so. Both techniques require expert knowledge on common delimiters in the protocol. To account for unknown protocols, the framework in the present study does not rely on common delimiters or any form of expert knowledge unlike these existing methods. The global sequence alignment technique NWSA used by NETZOB is also computationally time expensive ($\mathcal{O}(n^2)$), where n is the number of data packets), and makes use of only observable literal information, ignoring semantic information. Luo et al [7] proposed using Latent Dirichlet Allocation (LDA) from NLP to study the type distributions derived from the statistics of message N-grams to infer protocol message formats of different types. However, the study did not perform an extensive hyper-parameter tuning of the α & β that control the Dirichlet distribution or to select the size of LDA topics.

Kleber et al proposed NEMESYS [9] that uses the delta of the congruence in bit values of consecutive bytes to identify field boundaries in a packet using its intrinsic message structure. For text-based protocols with longer fields, this intrinsic structure-aware approach of obtaining field boundaries is able to produce more meaningful tokens than n-grams. NEMESYS is used by in novel field-based tokenisation approach.

III. Proposed APA Framework

The proposed APA framework comprises steps shown in Fig 1. It does not strictly mandate a linear application of methods, but rather encapsulates a set of unsupervised methods to be potentially used in combination. Section IV-B lists the combination of methods that we evaluated. Methods in each step of the framework are described as follows:

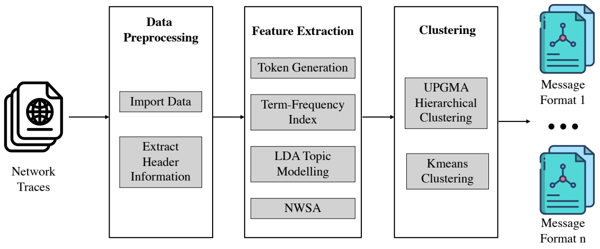


Fig. 1: Overview of Proposed APA Framework

A. Data Pre-Processing

Network traces in PCAP format are imported using the Python SCAPY library as binary or hexadecimal packets to facilitate feature extraction. We assumed that all protocols at OSI layers below the unknown protocol are known and their headers are therefore stripped from the packet. Finally, the header of the unknown protocol can be extracted from the remaining packet using the inferred header length from Section III-B5. Since many application layer protocols do not have a header, we do not extract a header from the application layer and instead, use the remaining payload.

B. Feature Extraction

The framework proceeds to extract distinguishing features, from the header of the unknown protocol, that will be utilized in subsequent clustering stage to differentiate packets from different protocols and message format types.

1) Tokens Generation: In NLP's N-grams tokenisation [8], one gram is a single word and adjacent words in a text string are combined to form tokens. In contrast, protocol headers are typically parsed as binary strings, one gram is represented by a single byte and consecutive bytes combined to form a N-grams token. Alternatively for text-based protocols with longer fields, a novel field-based tokenisation approach is proposed where NEMESYS [9] is used to infer field boundaries within a protocol header and tokenisation performed along these boundaries. The resultant corpus of tokens generated for each protocol header packet can be further analysed by advanced statistical methods like LDA in Section III-B3 to generate distinguishing features for subsequent clustering.

2) Term-Frequency (TF) Index: With a generated corpus of tokens per protocol packet, the next step is to generate an appropriate feature representation that is distinct for different protocols and message formats. First, we explored using the TF index for feature extraction. By recording the raw count of unique tokens in each corpus, a TF matrix of size $p \times n$ is generated (where p is number of protocols and n is number of unique tokens). Despite its intuitive representation, the generated matrix is often sparse, making computational storage unnecessarily expensive. Furthermore, the high dimensionality of the matrix meant that the application of dimensional reduction methods such as Principal Component Analysis (PCA) would often be required as clustering performance generally depreciates with higher dimensions.

3) Latent Dirichlet Allocation (LDA): Due to the limitations of the TF index discussed above, we sought a more efficient alternative. Next, we explored the use of LDA from NLP for feature extraction. LDA is an unsupervised topic modelling approach that allocates the generated tokens to a set of predefined number of LDA topics, based on the statistical extent of dissimilarity in which each individual token shared with other tokens in the corpus. Using the structured topic modelling (stm) package in R [10], a vector representation for each protocol packet in the data set is generated. This vector represents the posterior probability of the packet belonging to a particular LDA topic, given the corpus of tokens.

4) Optimising LDA Topic Size: The topic size is a key LDA hyper-parameter that determines not only the size, but also the generated feature representation that will be used to represent each protocol packet of a dataset. As such, it is crucial for practical deployment to automate and optimise the selection of this key LDA hyper-parameter, as optimising the topic size results in better clustering performance, with the optimised value being specific to the individual dataset. However despite its importance, such hyper-parameter tuning has not been explored in previous works.

By utilising the mean semantic coherence and exclusivity scores of a LDA topic size [10] as unsupervised metrics, the LDA topic size hyper-parameter can be automatically optimised for a given dataset, thus resulting in better clustering performance. The FREX metric [11] is used as a measure to quantify the degree of exclusivity of a given topic in a way that balances the word frequency, with $\mathrm{FREX}_{\mathrm{k,v}}$ being the weighted harmonic mean of the rank of token v in topic k (Equation 1). While the exclusivity score provides a quantity of measure for the degree of dissimilarity between LDA topics generated from a specific size, the semantic coherence score measures the extent in which tokens in the same topic co-occur together in the same communication protocol (Equation 2).

$$FREX_{\mathrm{k,v}} = \left(\frac{\omega}{ECDF(\beta_{k,v} / \sum_{j=1}^{K} \beta_{j,v})} + \frac{1-\omega}{ECDF(\beta_{k,v})} \right)^{-1} \quad (1)$$

where ECDF is the empirical CDF, $\beta_{k,v}$ is the topic-specific frequency of token v in topic k and ω is the weight set to 0.7 to favor exclusivity.

$$C_{\mathrm{k}} = \sum_{i=2}^{M} \sum_{j=1}^{i-1} log\left(\frac{D(v_{\mathrm{i}}, v_{\mathrm{j}}) + 1)}{D(v_{\mathrm{j}})} \right) \quad (2)$$

where $\mathrm{C_k}$ is the semantic coherence for topic k, $\mathrm{D(v_i,v_j)}$ is the number of times tokens $\mathrm{v_i}$ and $\mathrm{v_j}$ appear together in the same protocol and $\mathrm{D(v_j)}$ is the total number of times the token $\mathrm{v_j}$ appears in the data set.

With the derivation of the exclusivity and semantic coherence scores for each LDA topic in a given size of generated topics, the mean values of the topics in each size were used as a measure for the overall quality of the topics generated from that specific topic size. With the vast difference in scales of the mean exclusivity and semantic coherence scores, the values were normalised to between 0 and 1. The optimum topic size can then be determined graphically by the data point with the greatest Euclidean distance from the origin. This data point will correspond to the selected optimised LDA topic size hyper-parameter. For example, in Figure 2, it is observed that the optimum number of LDA topics selected for the Link Layer dataset (described in Section IV-A) was 6.

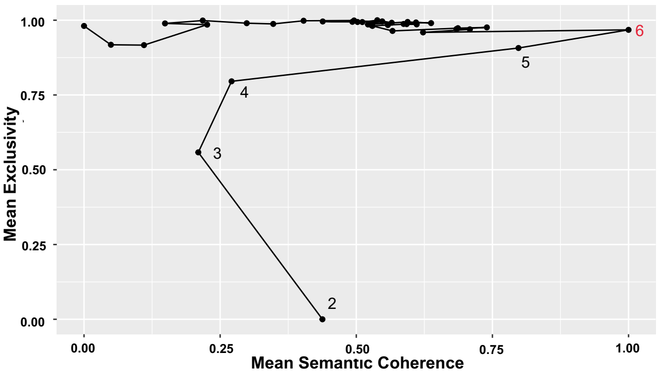


Fig. 2: Mean Exclusivity against Mean Semantic Coherence for Link Layer dataset. Points represent mean values for a topic size, with topic size increasing along the trace.

5) Optimising Extracted Protocol Header Length: In the data pre-processing stage (Section III-A), we would require the length of the unknown protocol header to be accurately estimated, as this would result in the right amount of features to be extracted, in order to obtain good clustering results. Similar to optimising the LDA topic size hyper-parameter (Section III-B4), optimising the extracted header length in the pre-processing is also crucial for practical deployment that has not yet been explored in previous works.

Based on our analysis, key features found in the header of the protocol packets are often sufficient in distinguishing between the different protocol and message format types. Conversely, using the entire protocol packet for feature extraction and clustering would often introduce a significant amount of stochastic noise into the data, thus adversely affecting the quality of LDA topics generated and subsequently clustering performance. Therefore, it is crucial to be able to accurately estimate the appropriate protocol header length to be extracted, and used for subsequent feature extraction and clustering stages. Just like the LDA topic size, it was observed that the extracted header length was also a hyper-parameter value that was specific to the individual data set.

We extend the approach to optimise the LDA topic size hyper-parameter (Section III-B4) by varying the header length, and produced different iterations of the plot in Figure 2. The goal is to determine the appropriate header length to be used for a dataset, given the various plots generated. A novel approach proposed, is to select the header length that generated the plot with the most isolated optimal data point (i.e. highest euclidean distance from origin). Consequently, the optimised header length selected would generate the optimised LDA topic size with greatest difference in exclusivity and semantic coherence scores. Mathematically, the degree of isolation is measured by mean difference in Euclidean distance between the optimum data point and its two adjacent neighbours.

With the combined hyper-parameter search space of the LDA topic size and the protocol header length having a modestly small area, we were able to iterate through all permutations of the 2 hyper-parameters, in order to optimize for the best APA performance

6) Needleman-Wunsch Sequence Alignment (NWSA): NSWA from bioinformatics is used by NETZOB [12] to compute the alignment score, by comparing the similarity between packet sequences via global sequence alignment. Iterating over each dataset, a matrix of alignment scores can be generated for clustering. It is more expensive computationally and thus unsuitable for clustering of large datasets.

C. Unsupervised Clustering

The framework proceeds to cluster packets of the same protocol or message format into disjoint sets using the set of representative features extracted via methods in Section III-B in an unsupervised manner via a similarity metric.

1) Similarity Metric: Conventionally, determining the degree of similarity between data points has often been through the use of Euclidean distance in the N-dimensional subspace. However, given that the dimension of the set of representative features increases with LDA topic size, the degree of sparseness also increases exponentially. Therefore, the curse of dimensionality [13] makes Euclidean distance a poor similarity metric candidate for protocols clustering. Alternatively, the Cosine similarity metric is often a better choice for high-dimensional data that considers each data point to be represented by a single vector and scores the similarity between pairwise vectors based on the angle between them.

2) UPGMA Hierarchical Clustering: The unweighted pair group method with arithmetic mean (UPGMA) algorithm is an agglomerative hierarchical clustering algorithm that regards each protocol packet as first belonging to a single cluster and then proceeds to combine the 2 most similar clusters to form a larger overarching cluster in an iterative manner using the cosine similarity metric. This is repeated until the threshold to indicate the minimum degree of dissimilarity between the clusters is exceeded. We use a static threshold of 0.5 and this results in a dendrogram exemplified in Figure 3.

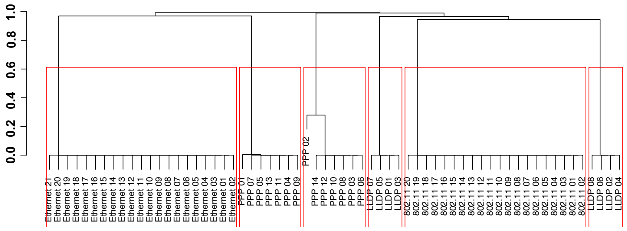


Fig. 3: Example of a dendrogram with 6 disjoint clusters for Link Layer Protocols.

3) K-Means Clustering: Alternatively, the K-Means clustering algorithm is an unsupervised learning algorithm that first assigns the data points randomly into K distinct and disjoint clusters, and then iteratively shifts the K centroids based on the class association of the data points to the nearest centroid. In each step, the algorithm assigns the data points to new clusters, such that the within sum of squares of the clusters are reduced in each iteration, eventually converging to a global minimum at some point. The number of predefined centroids, K, can be determined through the ”elbow method”, which is automated with the use of the Kneedle algorithm [14]. The stochastic nature of the clustering output is due to the random initialisation of the K centroids that may not be desirable.

IV. Experiments

A. Datasets

This study aims to group unknown protocols with similar packet formats at different OSI layers, and group packets of an unknown protocol with similar message formats. Hence, a wide variety of protocol and message format types is necessary to test the framework extensively. The diversity with and within our 9 datasets in total, comprehensively evaluates the robustness and the ability of our proposed framework to generalise for unknown protocols and achieve the aforementioned aims. Each dataset comprises 200 protocol packets. These 200 packets are selected by performing stratified sampling on a larger dataset (see Appendix) for each of the 9 datasets. The packets from the dataset detailed in the appendix are sourced from open-source databases, mainly WireShark Wiki. It is also assumed that the traffic is not encrypted.

We specifically engineered two characteristics of the dataset to make clustering more challenging as these characteristics are meant to replicate the exacting realities of un-encrypted network traces. First, we intentionally used a maximum of only 200 packets per dataset. This is because clustering with less data simulates scenarios with limited data, and so performing better with less data would further highlight the robustness of our framework. Second, the dataset is unbalanced like most un-encrypted network traffic. This additional step makes it challenging to cluster rare message types. In a supervised

TABLE I: Dataset protocol/type description

Dataset	Protocols/Types
Link Layer Protocols	Point to Point Protocol (PPP), Link Layer Discovery Protocol (LLDP), IEEE 802.11, Ethernet
Transport Layer Protocols	Internet Control Message Protocol (ICMP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Stream Control Transmission Protocol (SCTP)
Application Layer Protocols (Text)	Domain Name Server (DNS), Routing Information Protocol (RIP), Transport Layer Security (TLS)
Application Layer Protocols (Binary)	Trivial File Transfer Protocol (TFTP), Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP)
TCP Message Types	7 TCP message types e.g. ACK, PSH ACK, SYN, RST, FIN ACK
SCTP Chunk Types	16 SCTP chunk types e.g. INIT, COOKIE ECHO, DATA, HEARTBEAT, ASCONF, ACK
ICMP Types	4 ICMP types e.g. Reply, Request, Destination Unreachable, TTL Exceeded
HTTP Methods	3 HTTP methods e.g. 200 OK, GET, POST
DNS message types	4 DNS message types e.g. Query, Response Refused, Response No Error, Response No Such Name

problem, designing a dataset with the two mentioned characteristics would generally be seen as a weakness as data-driven models thrive on more ground-truth based data. As the present study presents an unsupervised learning problem, these characteristics instead test the robustness of our framework and shows its ability to perform well under challenging & real-world circumstances.

5 of the 9 datasets are used for fine-grain type clustering within protocols. The 5 protocols used here are ICMP, TCP, SCTP, DNS and HTTP and the aim is to cluster similar message format types in each protocol. Additionally, we have 4 OSI layers-based datasets, with each dataset containing protocols from the respective OSI layers. The aim is to group protocols with similar packet formats in each layer (e.g. TCP ACK, TCP SYN etc. for the TCP dataset). We also chose to split application layer protocols into textual and binary protocols due to the obvious difference in data structure used in these packets. Hence, the two types of protocols may require different sets of hyper-parameters in our proposed APA framework. The technique for differentiating the two types of application layer protocols is based on domain knowledge and involves searching for special ASCII patterns that occurs in textual protocols but not binary protocols. This method of searching using a predefined rule is very lightweight and and has negligible computational cost. Note that all message format types in the single protocol-based datasets and all protocols in the OSI layers-based datasets are assumed to be unknown. Table I describes the protocols, assumed to be unknown, that comprise the 4 OSI layers-based datasets, and some of the message types, assumed to be unknown, that comprise the 5 fine-grain type clustering datasets.

B. Experimental Setup

Our experiments have three objectives. First, to compare the different tokenisation methods of proposed framework, which is done by comparing N-grams and NEMESYS [9] across the 9 datasets. Second, to evaluate both the proposed LDA topic size and extracted protocol header length optimisation methods. This is done by varying the LDA topic sizes and header lengths to observe how the chosen topic size and header length compare with the actual best-performing topic size and header length. And finally, to compare the overall clustering performance across 5 different combinations of feature extraction & clustering methods of the proposed framework.

The first is to use the open source APA tool, NETZOB that utilizes NWSA feature extraction and UPGMA clustering. The second combines LDA features extraction with K-MEANS clustering. The third combines LDA features extraction with UPGMA clustering.The fourth uses TF index feature extraction with UPGMA clustering. N-grams tokenisation is used with all feature extraction methods, with the exception of NWSA. Finally, the fifth is our proposed hybrid approach that automatically selects the best feature extraction method based on our findings from the previous four approaches. By default, TF index feature extraction is used with UPGMA clustering. However, for application layer binary protocols, LDA feature extraction is used, and for application layer textual protocols, our proposed field-based tokenisation based on NEMESYS is used instead. The detection of binary or textual protocols can be achieved automatically via suitable predefined rules.

We repeat the entire process of proposed APA framework (in Section III) for all 9 datasets. Before starting our experiments, a comprehensive hyper-parameter tuning process was done. We first conducted a sensitivity analysis on which hyper-parameters affected the performance more and sorted them in a hierarchical list. Afterwards we sampled the more important hyper-parameters via grid search and went down the list. Note that as the header length and LDA topic size were dataset-sensitive, they were not fixed via hyper-parameter tuning, but they were determined using the proposed method in Sections III-B4 & III-B5. For NEMESYS, the σ value for bit congruence was optimal at 0.5 and that no tokens should be kept longer than 40 bytes. For N-grams tokenisation, hexadecimal packet representation was more efficient than binary for all feature extraction methods. Moreover, a gram size of 3 bytes was determined to be optimal. Optimal values for all tuned hyper-parameters will be used across all datasets.

All computations were run on a Windows 10 machine with Intel(R) Core(TM) i7-8550U CPU processor @1.80GHz and 16GB RAM. Computation effort of each run of our proposed APA framework for each dataset is strongly dependent on the dataset and ranges between $42.25s$ and $357.62s$ with a mean of $138.18s$. Run-time difference between the techniques compared were negligible for each dataset.

C. Performance Metrics

To quantify the overall clustering performance, the nature of the output must be realised. After clustering, the data packets are grouped together in clusters, but each cluster is not augmented with a class label due to the nature of this problem being unsupervised. For instance, if 100 data packets (of which 50 are TCP and 50 are UDP) are run through the APA framework and the output is 2 clusters (one of size 45 and another 55), the clusters can only be labelled '1' and '2' (cluster labels) but not TCP or UDP (class labels) directly. This lack of association is why traditional classification metrics like Accuracy and F-score cannot be used as metrics.

However, one may propose an additional step to convert cluster labels into class labels via a voting algorithm. A majority voting algorithm can be employed for each cluster by labelling it with the class label determined by the most common ground truth label in each cluster. For instance, a cluster with 40 packets, with 10 of them being UDP and 30 being TCP, would wholly be labelled as a TCP cluster with the UDP packets considered mis-clustered with the TCP packets. With this, a confusion matrix can be generated and the standard classification metrics can be determined. One flaw of this voting method is that it is swayed by any dataset bias. If the cluster of 45 packets comes from a dataset of 10% UDP and 90% TCP, then the cluster would have proportionately more UDP packets and hence become classified as UDP wholly. The voting method can therefore be improved to a proportion-based majority voting algorithm. After further review, the present study abstained from using any voting algorithm for two reasons.

First, the voting algorithm is an artificial step and does not do justice to the nature of clustering. Clustering only groups data points together, it does not label data points or groups. It would be more sensible to use metrics that measured how well similar data points are grouped together and how dissimilar data points are grouped separately without the need for class labels for every cluster. Second, the hyper parameters can be tampered with to artificially boost the accuracy score by taking advantage of another flaw of the voting algorithm. By setting the number of clusters to be very high in K-Means (or an extremely low threshold for UPGMA), each data packet can be forced to be in its own cluster, forcing the class label to always be correct. This would output an accuracy of 1, even though the output from the APA framework is nonsensical and of no use to an analyst. On the other end of the spectrum, the APA framework can be made to put all data packets into one cluster. This would mean that a dataset with 70% TCP packets would be labelled with an accuracy of 0.7 even though the output is also nonsensical to an analyst.

To circumvent the issue of having no class labels, the present study looked at few extrinsic clustering metrics; namely the Adjusted Rand Index (ARI) [15], the Fowlkes Mallows Score (FMS) [16] & the Adjusted Mutual Information (AMI) [17]. All three metrics compare between two clusters, which would be the output (in terms of cluster labels) and the ground truth (in terms of class labels). They compare how similarly the two clusters are grouped and adjust for chance. There is a lack of research to support which metric is better given the nature of clusters, hence the present study chose ARI as the main performance metric given its effectiveness & wide use in literature for unsupervised learning [18]–[20].

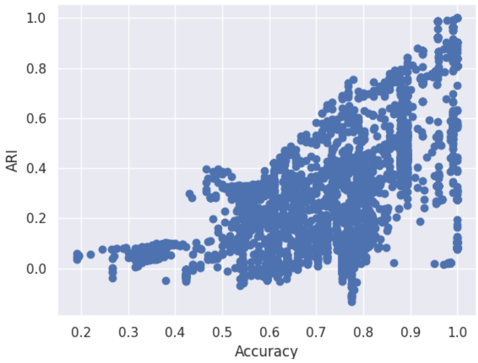


Fig. 4: Comparison of ARI against Accuracy with voting in over 13,000 test instances of the APA framework

Finally, a comparison between ARI & accuracy with voting is made to illustrate the weakness of the latter metric and determine a threshold for a satisfactory performance using ARI. During all of the experimental testing instances of the APA framework (>13,000), the ARI and accuracy with voting results were gathered and compared as shown in Figure 4. The thousands of data points on the bottom right quadrant shows all instances where the clustering performance was poor (ARI), yet the accuracy with voting was indicated to be high. The limitations of accuracy with voting forces the value to be higher. Hence, a high value may not indicate good clustering, but a low value generally indicates bad clustering. Based on Figure 4, any ARI over 0.4 never produces an accuracy lower than 0.6. It is difficult to determine a proper threshold of ARI for a satisfactory outcome for it is not as intuitive as accuracy. Hence, we use an ARI value of 0.4 as the threshold as it has been shown that any value of ARI below 0.4 could be associated with bad clustering (low accuracy).

V. Results and Discussion

Our first objective is to compare the tokenisation methods (Section IV-B). Figure 5 shows the ARI for NEMESYS and N-grams tokenisation across the 9 datasets. There is no clear winner and NEMESYS out-performs in 5 out of 9 datasets. Closer analysis shows that NEMESYS performs significantly better than N-grams tokenisation for both application layer textual protocols and HTTP protocol datasets, which is also an application layer textual protocol. Our results suggest that NEMESYS tokenization is more effective for application layer textual protocols. This leads us to propose a novel field-based tokenisation based on NEMESYS for application layer textual protocols in our proposed hybrid approach (Section IV-B).

Our second objective is to evaluate both the proposed LDA topic size and extracted protocol header length optimisations. Figure 2 shows that the optimised LDA topic size for the link layer dataset was chosen to be 6. For validation, we plotted how ARI varies with changing topic size (Figure 6) and observed that performance was poor (i.e. low ARI) with small

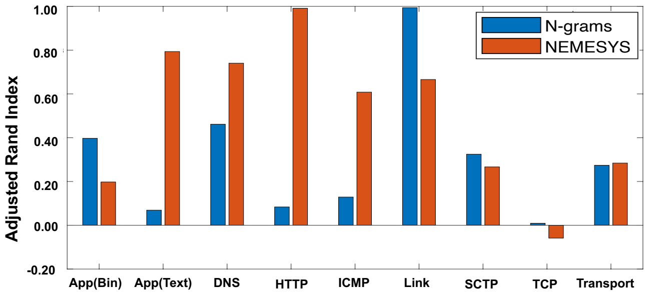


Fig. 5: ARI of NEMESYS vs. N-grams tokenisation

topic sizes, but improves until ARI peaks with optimal topic size 6. Similarly, the LDA topic sizes selected using proposed optimisation method in Section III-B4 are either optimal or near-optimal for the other datasets.

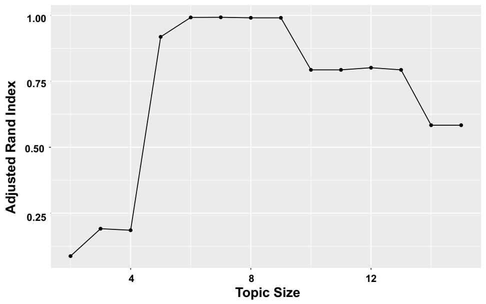
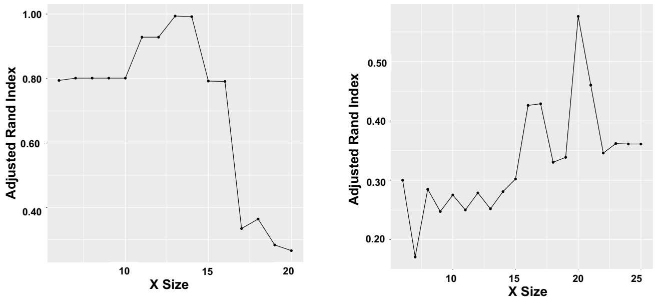


Fig. 6: ARI of LDA feature extraction against topic size

For optimising the extracted protocol header length needed in pre-processing stage (Section III-A), Figures 7a & 7b show that the optimised protocol header lengths for link layer and transport layer datasets were chosen to be at 14 and 20 bytes respectively, which corresponded to highest ARI scores. Similarly, the header lengths selected using proposed optimisation method in Section III-B5 were either optimal or near-optimal for the other datasets. Therefore, both proposed LDA topic size and extracted protocol header length optimisations have been validated and are deemed crucial for practical deployment, which has not yet been previously explored.

Figure 8 compares final clustering performance of the 5 approaches (Section IV-B) across all 9 datasets. Results show that our proposed hybrid approach is best performing in 7 out of 9 datasets, with ARI $>$ 0.4 for 6 datasets. Unfortunately, all approaches did not achieve satisfactory performance (ARI $<$ 0.4) for STCP, TCP and Transport layer datasets. Upon further investigation, we realised that these 3 datasets comprised of more protocols and message format types to differentiate with less distinct features, thus making clustering challenging. However, after further analysis of the UPGMA dendrograms generated from our proposed approach, we observed that by optimising the static UPGMA threshold of 0.5 (Section III-C2), it is possible to significantly improve the performance, which we leave for future works.



(a) Link Layer dataset

(b) Transport Layer dataset

Fig. 7: ARI of LDA feature extraction against protocol header length for Link and Transport datasets

Finally, due to the extensive coverage of our 9 datasets across all the OSI layers with diverse protocols and message format types, we have proven the robustness of our proposed hybrid approach to generalise for unknown protocols, which is crucial for practical deployment and has not been adequately addressed in previous works.

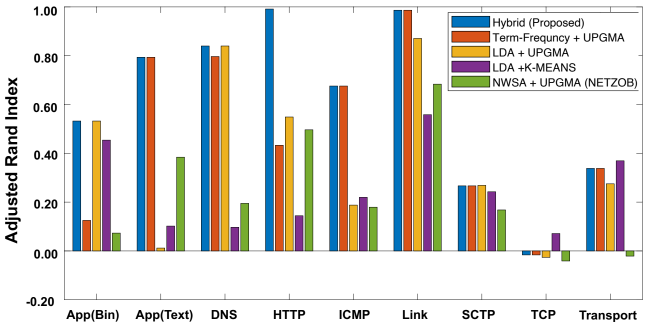


Fig. 8: Comparing ARIs of 5 approaches across 9 datasets

VI. Conclusion and Future Works

In conclusion, we proposed a comprehensive APA framework and evaluated various combinations of feature extraction and clustering methods, including those used by NETZOB [5]. Our proposed hybrid approach, that utilizes a novel field-based tokenisation based on NEMESYS for application layer textual protocols, is best performing in 7 out of 9 datasets with ARI $>$ 0.4 for 6 datasets. This result proves the robustness and generalising ability of our proposed hybrid approach. We also validated our proposed automated optimisation methods, for both LDA topic size and extracted protocol header length, that is crucial for practical deployment. However, since computational cost was not the primary focus of our present study, more work can be done to optimise our code and this will help us make more detailed comparisons and analysis of the computational costs of our proposed framework and other existing frameworks in future.

We also hope that our works contributed as crucial foundation stones for future APA works to be built upon. With recent advances in Deep Learning, like Deep Auto-Encoders for automated features extraction, it will be exciting to explore the application of these advanced Machine Learning (ML) methods for unsupervised learning in APA. Finally, we have only explored the tip of the APA iceberg and in the future, we hope to build upon our proposed APA framework to explore application of advanced ML methods in more areas of APA.

Acknowledgment

We wish to thank Mr Chia Yong Kang and Mr Tan Ping Liang for their invaluable contributions and feedback.

References

[1] J. Duchene, C. Le Guernic, E. Alata, V. Nicomette, and M. Kaâniche, "State of the art of network protocol reverse engineering tools," *Journal of Computer Virology and Hacking Techniques*, vol. 14, no. 1, pp. 53–68, 2018.

[2] W. Cui, J. Kannan, and H. J. Wang, "Discoverer: Automatic protocol reverse engineering from network traces." in *USENIX Security Symposium*, 2007, pp. 1–14.

[3] T. Krueger, N. Krämer, and K. Rieck, "Asap: Automatic semantics-aware analysis of network payloads," in *International Workshop on Privacy and Security Issues in Data Mining and Machine Learning*. Springer, 2010, pp. 50–63.

[4] J. Antunes, N. Neves, and P. Verissimo, "Reverse engineering of protocols from network traces," in *2011 18th Working Conference on Reverse Engineering*. IEEE, 2011, pp. 169–178.

[5] G. Bossert, F. Guihéry, and G. Hiet, "Towards automated protocol reverse engineering using semantic information," in *Proceedings of the 9th ACM symposium on Information, computer and communications security*, 2014, pp. 51–62.

[6] M. A. Beddoe, "Network protocol analysis using bioinformatics algorithms," *Toorcon*, 2004.

[7] X. Luo, D. Chen, Y. Wang, and P. Xie, "A type-aware approach to message clustering for protocol reverse engineering," *Sensors*, vol. 19, no. 3, p. 716, 2019.

[8] Y. Wang, X. Yun, M. Z. Shafiq, L. Wang, A. X. Liu, Z. Zhang, D. Yao, Y. Zhang, and L. Guo, "A semantics aware approach to automated reverse engineering unknown protocols," in *2012 20th IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2012, pp. 1–10.

[9] S. Kleber, H. Kopp, and F. Kargl, "{NEMESYS}: Network message syntax reverse engineering by analysis of the intrinsic structure of individual messages," in *12th {USENIX} Workshop on Offensive Technologies ({WOOT} 18)*, 2018.

[10] M. E. Roberts, B. M. Stewart, and D. Tingley, "Stm: An r package for structural topic models," *Journal of Statistical Software*, vol. 91, no. 1, pp. 1–40, 2019.

[11] J. Bischof and E. Airoldi, "Nd "poisson convolution on a tree of categories for modeling topical content with word frequency and exclusivity."," 2013.

[12] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.

[13] M. Köppen, "The curse of dimensionality," in *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, vol. 1, 2000, pp. 4–8.

[14] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a" kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st international conference on distributed computing systems workshops*. IEEE, 2011, pp. 166–171.

[15] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.

[16] E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.

[17] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *The Journal of Machine Learning Research*, vol. 11, pp. 2837–2854, 2010.

[18] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *Journal of Network and Computer Applications*, vol. 108, pp. 87–111, 2018.

[19] H. Pirim, B. Ekşioğlu, A. D. Perkins, and Ç. Yüceer, "Clustering of high throughput gene expression data," *Computers & operations research*, vol. 39, no. 12, pp. 3046–3061, 2012.

[20] Y. Yang, *Temporal data mining via unsupervised ensemble learning*. Elsevier, 2016.

Appendix

Dataset Support Description

Dataset	Protocols/Types	Support
Link Layer Protocols	Point to Point Protocol	14
	Link Layer Discovery Protocol	8
	IEEE 802.11	86
	Ethernet	78
Transport Layer Protocols	Internet Control Message Protocol	22
	Transmission Control Protocol	100
	User Datagram Protocol	26
	Stream Control Transmission Protocol	38
Application Layer Protocols (Binary)	Domain Name Server	38
	Routing Information Protocol	12
	Transport Layer Security	20
Application Layer Protocols (Text)	Trivial File Transfer Protocol	20
	Hypertext Transfer Protocol	19
	Simple Mail Transfer Protocol	28
TCP Message Types	ACK	3357
	PSH ACK	348
	SYN	315
	SYN ACK	288
	RST	2
	RST ACK	3
	SIN ACK	157
SCTP Chunk Types	INIT	2
	INIT ACK	2
	COOKIE ECHO	2
	COOKIE ACK	2
	DATA	120
	SACK DATA	1
	SACK	108
	SHUTDOWN	3
	SHUTDOWN ACK	2
	SHUTDOWN COMPLETE	2
	HEARTBEAT	73
	HEARTBEAT ACK	63
	HEARTBEAT ACK DATA	1
	ASCONF	3
	ASCONF ACK	3
ICMP Types	Reply 0	23
	Request 8	27
	TTL Exceeded 11	12
	Destination Unreachable 3	2
HTTP Methods	200 OK	146
	GET	537
	POST	6
DNS message types	Query	36
	Response Refused	1
	Response No Error	23
	Response No Such Name	6