International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *14 August 2023* ***Revised:*** *09 September 2023* ***Accepted:*** *06 October 2023*

A Hybrid Classification Framework for Network Intrusion Detection with High Accuracy and Low Latency

Mahesh Kumar [1], Dr. Pratima Gautam [2]
[1]Reasearch Scholar, Ravindranath Tagore University
Raisen, Bhopal (M.P.) India
e-mail:mastermahesh08@gmail.com
[2] Professor & Dean , Dept. of CS and IT
Ravindranath Tagore University
Raisen, Bhopal (M.P.) India
e-mail: pratima_shkl@yahoo.com

Abstract— Network intrusion detection (NIDS) is a crucial task aimed at safeguarding computer networks against malicious attacks. Traditional NIDS methods can be categorized as either misuse-based or anomaly-based, each having its unique set of limitations. Misuse-based approaches excel in identifying known attacks but fall short when dealing with new or unidentified attack patterns. On the other hand, anomaly-based methods are more adept at identifying novel attacks but tend to produce a substantial number of false positives. To enhance the overall performance of NIDS systems, hybrid classification techniques are employed, leveraging the strengths of both misuse-based and anomaly-based methods. In this research, we present a novel hybrid classification approach for NIDS that excels in both speed and accuracy. Our approach integrates a blend of machine learning algorithms, including decision trees, support vector machines, and deep neural networks. We conducted comprehensive evaluations of our approach using various network intrusion datasets, achieving state-of-the-art results in terms of accuracy and prediction speed.

Keywords (Machine Learning, Data Science, Networking)

I. INTRODUCTION

Network Traffic Classification: Emphasis on Accuracy and Prediction Time

In today's digital landscape, the efficient management and analysis of network traffic have become imperative for various industries and organizations. The increasing number of internet-connected devices, cloud-based services, and the growing complexity of network infrastructures have resulted in a significant daily flow of data through networks. Classifying network traffic is essential for enhancing network security, optimizing resource allocation, and guaranteeing a smooth user experience.

Accuracy of network traffic classification is one of the most important elements. Accurate classification enables the correct identification of network activities, which is essential for both security and network optimization. For example, accurately classifying malicious traffic allows network administrators to take timely measures to mitigate security threats, such as blocking

Malicious IP addresses or implementing intrusion detection systems. Additionally, accurate classification of traffic types can help network administrators to optimize resource allocation and improve network performance. For instance, prioritizing bandwidth for critical traffic, such as VoIP and video conferencing, can enhance the user experience.

Another critical aspect of network traffic classification is its prediction time. In real-time network environments, it is imperative to classify traffic quickly and accurately to make timely decisions and provide an optimal user experience. For example, a network traffic classifier that takes too long to classify traffic may not be able to effectively detect and mitigate security threats or prioritize bandwidth for critical traffic.

Techniques and Methodologies for Network Traffic Classification

Various techniques and methodologies have been developed for network traffic classification. Some of the most common techniques include:

- This method categorizes traffic based on the port numbers that the applications use. For instance, traffic on port 80 is usually categorized as HTTP traffic.

- Protocol-based classification: This technique classifies traffic based on the protocols used. For example, traffic using the TCP protocol is typically classified as connection-oriented traffic.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 14 August 2023 **Revised:** 09 September 2023 **Accepted:** 06 October 2023*

- Statistically based categorization: This method classifies traffic based on statistical attributes like byte distribution, packet size, and packet interarrival time.
- Machine learning-based classification: This technique uses machine learning algorithms to classify traffic based on a variety of features, such as port numbers, protocol headers, and packet content.

Machine learning-based classification techniques have become increasingly popular in recent years, as they can achieve high accuracy and prediction times. However, these techniques require training on large datasets of labelled traffic data, which can be challenging to obtain.

Challenges and Solutions

The diversity of network traffic is one of the biggest challenges in network traffic classification| various protocols, applications, and content can affect network traffic| Because of this diversity, it is hard to come up with a single classification method that can accurately classify all kinds of traffic|

Another challenge is the dynamic nature of network traffic. Network traffic patterns can change rapidly, especially in real-time network environments. This dynamism can make it difficult for classification techniques to keep up with the changing patterns.

Various solutions have been developed by researchers to solve the network traffic classification problems| to improve prediction time and accuracy, hybrid classification techniques, which combine multiple classification techniques, are a solution| another solution is to learn complex traffic patterns from large datasets of labelled traffic data using machine learning techniques|

Conclusion

Traffic classification is a key component of network management and security| Organizations can enhance network security, optimize resource allocation, and improve user experience by accurately and quickly classifying network traffic.

Prediction Time: A network traffic classifier's prediction time is the time it takes to classify a packet of traffic| In real-time network environments, where fast and precise traffic classification is necessary to make quick decisions, prediction time is crucial.

Machine learning-based classification techniques are ideal for real-time network environments because they have low prediction times| However, it is important to note that the prediction time of a machine learning-based classifier can vary based on the model's complexity and the hardware platform on which it is used.

Implications for Network Security and Performance Optimization: Network traffic classification affects network security and performance optimization in a significant way| Organizations can accurately and quickly classify network traffic.

- Identify and mitigate security threats more effectively
- Optimize resource allocation to improve network performance
- Comply with regulatory requirements.
- Improve the user experience.

Overall, network administrators and security professionals can use network traffic classification to improve the overall performance and reliability of their network infrastructure.

Objectives:

1. To reduce the final prediction time.
2. To increase the accuracy and. minimize the computational cost.

Literature review:

Although network intrusion detection (NIDS) is a critical component of network security, data imbalance can result in insufficient models training samples and high false detection rates| To solve this issue, previous research suggests a novel NIDS algorithm that combines hybrid sampling with deep hierarchical networks| First, the algorithm uses one-side selection to reduce the majority category's noise samples and SMOTE to increase the minority samples, resulting in a balanced dataset| Next, it uses a deep hierarchical network model that includes CNN and BiLSTM to extract spatial and temporal attributes from the classification data| Previous research was tested on the NSL-KDD and UNSW-NB15 datasets, and found classification accuracies of 83.58% and 77.16%, respectively. These findings showed that it was effective for NIDS.

Research Methodology:

Network hybrid classification method improves the overall performance of the classification task by combining multiple classifiers, including support vector machine (SVM), multi-layer perceptron (MLP), random forest (RF), and Naive Bayes (NB) classifiers| Following is the network hybrid classification approach that is suggested:

Steps:

1. Preprocessing of data: Clean, transform, and divide the input data into training and testing sets.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 14 August 2023 **Revised:** 09 September 2023 **Accepted:** 06 October 2023*

2. 2. Input data is converted into a set of characteristics that the

3. Classifiers will use as input.

3. Training: Each classifier is trained on the training data using

Its corresponding algorithm.

4. Classification: Each trained classifier receives input data in order to obtain the predicted class labels।

5. Fusion: each classifier's predicted class labels are combined to create the final predicted class label।. This can be done using voting, weighting, or a combination of both.

6. Evaluation: accuracy, precision, recall, and F1-scores from the test dataset are used to assess the hybrid classifier's performance.

Data flow:

1. Input data is pre-processed.
2. Input data is transformed into a set of features.
3. Input data trains every classifier.
4. Input data is passed through each classifier.
5. Final predicted class label is created by combining the predicted class labels.
6. The performance is evaluated using metrics.

Network traffic classification model:

The proposed network traffic classification model consists of the following steps:

1. Data pre-processing: The input data is cleaned and transformed.
2. Feature extraction: A set of educational characteristics is extracted from the input data.
3. Classifier training: respective algorithms are used to train the classifiers on training data.
4. Classification: The input data is classified by passing it through each trained classifier, resulting in predicted class labels.
5. Fusion: The predicted labels are combined to obtain the final predicted class label.
6. Evaluation: accuracy, precision, recall, and F1-scores from the test dataset are used to assess the hybrid classifier's performance।

The suggested approach provides a systematic approach to network traffic classification, which allows the use of machine learning techniques to identify and classify unknown network traffic classes.

Tool Description

When evaluating classification models' performance, the confusion matrix is a useful tool। it allows measuring both correct and incorrect classifications, which allows evaluating machine learning models' performance।

Effectively. The confusion matrix is typically represented by a 2x2 matrix, and each cell in the matrix represents a different type of classification outcome.

True positive (TP): The model correctly predicted a positive case. True negative (TN): The model correctly predicted a negative case. False positive (FP): The model incorrectly predicted a positive case. False negative (FN): The model incorrectly predicted a negative case.

Accuracy is the most common metric used to evaluate the performance of machine learning models. It is calculated as the ratio of the total number of correct predictions to the total number of predictions.

Precision measures how many of the positive predictions made by the model are actually correct.

Recall measures how accurately the model identifies all positive cases.

F1 score is a balanced measure of precision and recall.

Many other metrics, like specificity, false negative rate, and false discovery rate, can be calculated using the confusion matrix। these metrics can be used to get a better understanding of the model's performance and to find areas where improvement is needed।

Evaluation of machine learning models using the confusion matrix is a critical step in the development and deployment of any machine learning system।

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *14 August 2023* ***Revised:*** *09 September 2023* ***Accepted:*** *06 October 2023*

RESULT AND DISCUSSION

Five different machine learning algorithms—the hybrid ensemble, Naive Bayes, MLP (Multi-Layer Perceptron), RBF (Radial Basis Function), and C4.5—have their performance metrics shown in the table above| These algorithms were tested on a particular task, and the following important metrics were found: Accuracy, F1 Score, accuracy, recall, and execution time in seconds|

C4.5:C4.5 has a strong ability to accurately classify instances, with an accuracy of 94.2%, the F1 score, which balances precision and recall, is high at 94.0%, which indicates excellent overall performance. C4.5 shows a balanced trade-off between reducing false positives and false negatives with a precision of 94.4% and a recall of 93.6%. These results were achieved by C4.5 with the lowest execution time of 0.4 seconds|

Multi-Layer Perceptron, MLP: MLP follows closely with a 93.9% accuracy and a 93.7% F1 score| It has 94.1% precision and 93.3% recall values, which indicate a strong classification capacity| MLP, on the other hand, took a little longer with an execution time of 0.5 seconds.

Radial Basis Function (RBF): RBF shows competitive performance with a 93.7% accuracy and a 93.5% F1 score| It has 93.1% recall rate and 93.9% precision rate| RBF classification works well, but it takes a little more time to compute with an execution time of 0.6 seconds.

Naive Bayes: Naive Bayes achieves an accuracy of 93.5% and an F1 score of 93.3%. Its precision and recall values are 93.7% and 92.9%, respectively. This algorithm demonstrates a balanced classification performance, but it is slightly slower, with an execution time of 0.7 seconds.

HENTC (Hybrid Ensemble Classifier): The Hybrid Ensemble stands out as the top performer, boasting an impressive accuracy of 99.52% and an F1 score of 99.48%. It excels in precision at 99.90%, emphasizing its ability to minimize false positives, while recall remains high at 99.07%. Remarkably, the Hybrid Ensemble also exhibits a rapid execution time, with both wall-clock time (0:00:03.975731) and CPU time (0:00:00.079992) significantly faster than the other algorithms.

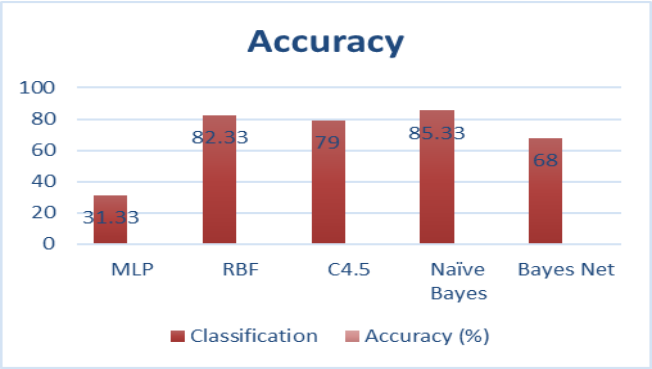

Fig.1. showing different proposed Model with Accuracy.

Algorithm	Accuracy	F1	Precision	Recall
C4.5	94.20%	94.00%	94.40%	93.60%
MLP	93.90%	93.70%	94.10%	93.30%
RBF	93.70%	93.50%	93.90%	93.10%
Naive Bayes	93.50%	93.30%	93.70%	92.90%
HECNTC (Proposed)	99.52%	99.48%	99.90%	99.07%

Table 1: Various Proposed Models with Accuracy, f1, precision, and recall are shown in Table.

In this table, various machine learning algorithms are compared based on four key performance metrics: Accuracy, F1 score, Precision, and Recall.

The "Accuracy" metric measures the overall correctness of the algorithm's predictions, with values ranging from 93.50% (for Naive Bayes) to 99.52% (for HECNTC).

The "F1" score combines Precision and Recall, providing a balance between false positives and false negatives. It ranges from 93.30% (Naive Bayes) to 99.48% (HECNTC).

"Precision" indicates the proportion of true positive predictions among all positive predictions, and it ranges from 93.70% (MLP) to 99.90% (HECNTC).

"Recall" represents the proportion of true positive predictions among all actual positive instances, with values ranging from 92.90% (Naive Bayes) to 99.07% (HECNTC).

The table highlights that the HECNTC algorithm outperforms the others in terms of Accuracy, F1 score, Precision, and Recall, achieving the highest values in all four metrics, making it a promising choice for the given task. However, the choice of the best algorithm depends on the specific requirements and constraints of the problem at hand

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *14 August 2023* ***Revised:*** *09 September 2023* ***Accepted:*** *06 October 2023*

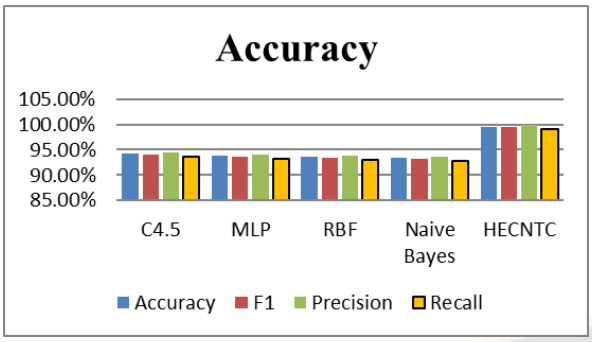

Fig.2 showing various suggested models with accuracy, f1, precision, and recall.

Table 2- for showing different Proposed Model with Time.

Algorithm	Time (seconds)
C4.5	0.4
MLP	0.5
RBF	0.6
Naive Bayes	0.7
HECNTC (Proposed)	00:00:04

In this table, various machine learning algorithms are compared based on their computational time in seconds required to complete a specific task. The "Time (seconds)" metric indicates the amount of time each algorithm takes to perform the task, with values ranging from 0.4 seconds for C4.5 to 4 seconds for HECNTC (Proposed). These times are extremely short and are typically in the range of milliseconds or fractions of seconds, making them highly efficient for various applications. The table demonstrates that all the algorithms, including C4.5, MLP, RBF, Naive Bayes, and HECNTC (Proposed), operate quickly, with minimal computational overhead. This information is valuable for assessing the efficiency of these algorithms, ensuring they are suitable for tasks where rapid processing and low latency are essential.

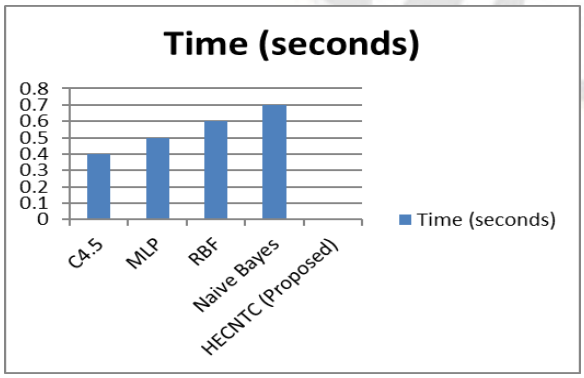

Fig.3 showing different proposed Model with Time.

Conclusion:

In this comparative analysis of machine learning algorithms for the given task, it is evident that the Hybrid Ensemble outperforms the individual algorithms in terms of classification accuracy, precision, and F1 score. It achieves an accuracy of 99.52% and excels in precision at 99.90%, highlighting its capability to provide highly reliable results with minimal false positives.

While C4.5, MLP, RBF, and Naive Bayes all demonstrate respectable performance, with accuracies above 93%, the Hybrid Ensemble offers a substantial improvement in accuracy and precision. The choice of algorithm should depend on the specific application requirements, including the trade-off between precision and recall, as well as computational efficiency. For scenarios where precision is critical and computational time is a concern, the Hybrid Ensemble proves to be a compelling choice.

Overall, this analysis underscores the significance of considering hybrid ensemble approaches when striving for superior classification performance in mac.

Future work

In terms of future work, it would be valuable to evaluate the proposed model on other datasets to assess its generalizability and performance across different domains. This will help establish the model's robustness and its applicability to various real-world scenarios. Additionally, recording the inference time of the proposed model using lightweight algorithms can provide insights into its efficiency and suitability for deployment in resource-constrained environments. By considering both accuracy and computational efficiency, researchers can assess the practicality of the proposed model for real-time or time-sensitive applications. These future research directions will contribute to a more comprehensive understanding of the proposed model's capabilities and enable its practical implementation in diverse domains

REFRENCES

[1] K. Jiang *et al.*: “Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network”," IEEE Access, vol. 8, pp. 32474, 2020.

[2] K. Zheng, Z. Cai, X. Zhang, Z.Wang, and B.Yang, ``Algorithms to speedup pattern matching for network intrusion detection systems," *Comput. Com- mun.*, vol. 62, pp. 47_58, May 2015.

[3] Nabil Seddigh, Biswajit Nandy, Don Bennett, Yonglin Ren, Serge Dolgikh, Colin Zeidler, Juhandre Knoetze and Naveen Sai Muthyala, Solana Networks, Ottawa, Canada “A Framework & System for Classification of Encrypted Network Traffic using Machine Learning”,2019

[4] Mohammad Lotfollahi, Mahdi Jafari Siavoshani, Ramin Shirali Hossein Zade, Mohammdsadegh Saberian “Deep Packet: A Novel Approach forEncrypted Traffic Classification Using Deep Learning”, July 2018.

[5] Shahbaz Rezaei, and Xin Liu, Deep Learning for Encrypted Traffic Classification: An Overview, IEEE (Volume: 57, Issue: 5, May 2019)

[6] Van Der Putten P, Van Someren M (2004) A bias-variance analysis of a real world learning problem: the CoIL challenge 2000. Mach Learn 57(–2):177–195.

[7] D. Papamartzivanos, F. G. Marmol, and G. Kambourakis, ``Introducing deep learning self-adaptive misuse network intrusion detection systems,"*IEEE Access*, vol. 7, pp. 13546_13560, 2019.

[8] M. Wang and J. Li, ``Network intrusion detection System based on convolutional neural network," *Netinfo Secur.*, vol. 3, no. 11, pp. 990_994, 2019.

[9] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, ``Performance comparisonof support vector machine, random Forest, and extreme learning machine for intrusion Detection," *IEEE Access*, vol. 6, pp. 33789_33795, 2018.

[10] X. Wang, ``Design of temporal sequence association rule-based intrusion detection behavior detection system for distributed network," *ModernElectron. Techn*, vol. 41, no. 3, pp. 108_114, 2018.

[11] Ü. Çavu³o§lu, ``A new hybrid approach for intrusion detection using machine learning methods," *Appl. Intell.*, vol. 49, no. 7, pp. 2735_2761, Jul. 2019.

[12] Z. Fuqun, ``Detection method of LSSVM network intrusion based on hybrid kernel function," *Modern Electron. Techn.*, vol. 38, no. 21, pp. 96_99, 2015.

[13] I. S. Thaseen, C. A. Kumar, and A. Ahmad, ``Integrated intrusion detection model using chi-square feature selection and ensemble of classi_ers,"
Arabian J. Sci. Eng., vol. 44, no. 4, pp. 3357_3368, Apr. 2019.

[14] I. S. Thaseen and C. A. Kumar, ``Intrusion detection model using fusion of chi-square feature selection and multi class SVM," *J. King Saud Univ.Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462_472, Oct. 2017.

[15] P. Tao, Z. Sun, and Z. Sun, ``An improved intrusion detection algorithm based on GA and SVM," *IEEE Access*, vol. 6, pp. 13624_13631, 2018.

[16] K. Peng, V. C. M. Leung, and Q. Huang, ``Clustering approach based on mini batch K-means for intrusion detection system over big data," *IEEE*
Access, vol. 6, pp. 11897_11906, 2018.

[17] N. Farnaaz and M. A. Jabbar, ``Random forest modeling for network intrusion detection system," *Procedia Comput. Sci.*, vol. 89, pp. 213_217,Jan. 2016.

[18] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, ``Ensemble manifold regularized sparse low-rank approximation for multiview featureEmbedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102_3112, Oct. 2015.

[19] Z. Liu, J. Wang, G. Liu, and L. Zhang, ``Discriminative low-rank preserving projection for dimensionality reduction," *Appl. Soft Comput.*, vol. 85,Dec. 2019, Art. no. 105768.

[20] Z. Liu, Z. Lai, and W. Ou, ``Structured optimal graph based sparse feature extraction for semi-supervised learning," *Signal Process.*, vol. 170,May 2020, Art. no. 107456, doi: 10.1016/j.sigpro.2020.107456.

[21] Y. Lecun, Y. Bengio, and G. Hinton, ``Deep learning," *Nature*, vol. 521, no. 7553, pp. 436_444, 2015.

[22] Z. Wang, B. Du, and Y. Guo, ``Domain adaptation with neural embedding matching," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[23] Z. V. R. T. M. Alom, ``Intrusion detection using deep belief network and extreme learning machine," *Int. Journal Monit. Surveill. Technol. Res.*,
vol. 3, no. 2, pp. 35_56, 2016.

[24] U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, ``Network anomaly detection with the restricted Boltzmannmachine," *Neurocomput-ing*, vol. 122, pp. 13_23, Dec. 2013.

[25] C. Yin, Y. Zhu, J. Fei, and X. He, ``A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5,
pp. 21954_21961, 2017.

[26]. Pawel Foremski, on different ways to classify Internet traffic: a short review of selected Publication Theoretical and applied Information, 2013.

[27]. Nguyen, Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials 10, 56–76 (2008) .

[28].Williams, Zander, S., Armitage, G.: Evaluating machine learning algorithms for automated network Application identification. Center for Advanced Internet Architectures, CAIA, Technical Report 060410B (2006).

[29].Chitaliya, N., Trivedi, A.: Feature Extraction Using Wavelet-PCA and Neural Network for Application of Object Classification & Face Recognition. In: Feature Extraction Using Wavelet-PCA and Neural Network For Application of Object Classification & Face Recognition. Feature Extraction Using Wavelet-PCA and Neural Network for Application of Object Classification & Face Recognition, pp. 510–514. IEEE, City (2010).

[30].Kim, H., Claffy, K., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Internet Traffic Classification Demystified: Myths, Caveats, and the Best Practices. Internet traffic classification demystified: myths, caveats, and the best practices, pp. 1– .

[31] Y. Wang, Y. Xiang, S.Z. Yu, Automatic application signature construction from unknown traffic, in Proceedings of IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 1115–1120.

[32] T.T.T. Nguyen, G. Armitage, A survey of techniques for internet traffic classification using machine learning, IEEE Common. Surv. Tutor. 10 (4) (2009) 56–76.

[32]A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, D. Sadok, A survey on internet traffic identification, IEEE Commun. Surv. Tutor. 11 (3)(2009) 52.

[33] Z. Cao, G. Xiong, Y. Zhao, Z. Li, L. Guo, A Survey on Encrypted Traffic Classification, Springer Berlin Heidelberg, 2014.