International Journal of Engineering Research and Sustainable Technologies
Volume 2, No.2, June 2024, P 3 - 11
ISSN: 2584-1394 (Online version)

INVESTIGATING THE EFFECT OF TRAFFIC SAMPLING ON MACHINE LEARNING BASED NETWORK INTRUSION DETECTION APPROACHES

Cinthiya Joy Godly[1] Sreenithi R[2]
[1]Research Scholar, Federation University, Institute of Innovation Science and Sustainability, Australia
[2]gnitho, Chennai, India
* **Corresponding author email address**: c.godly@federation.edu.au

DOI: https://doi.org/10.63458/ijerst.v2i2.77 | ARK: https://n2t.net/ark:/61909/IJERST.v2i2.77

Abstract

The goal of this Paper is to improve cybersecurity threat detection by thoroughly examining deep learning and machine learning models. The study attempts to solve the difficulty of precisely categorizing and forecasting hostile actions in network traffic by focusing on a dataset that encompasses a variety of cyber threats. Preprocessing the data, using Principal Component Analysis (PCA) to apply dimensionality reduction, and putting a variety of machine learning algorithms into practice— including Logistic Regression, K-Nearest Neighbours, Gaussian Naive Bayes, Support Vector Machines, Decision Trees, and Random Forest—are all part of the methodology. Important conclusions highlight how ensemble models— Random Forest in particular—work well to achieve notable precision and accuracy. Principal Component Analysis's effect on model performance is also examined, providing information about the significance of features and the interpretability of the model. In addition to highlighting the promise of ensemble methods for reliable threat detection, the research provides insightful information about the efficacy of different machine learning algorithms in cybersecurity. The study's insights have practical consequences for cybersecurity practitioners and lay the groundwork for future cybersecurity analytics research projects.

Keywords: *Cyber Threat Intelligence, Machine Learning Models, Deep Learning Techniques, Logistic Regression, KNearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machines (SVM), Convolutional Neural Network (CNN), d*

1. Introduction

In the rapidly evolving field of cybersecurity, creative approaches are required for the effective detection and reduction of cyberthreats. The increased interconnectedness of businesses is leading to a significant problem with network security vulnerabilities. Modern methods are needed to identify threats quickly and accurately because sophisticated cyberattacks are becoming more common. This work aims to address the fundamental problem of enhancing cybersecurity threat detection systems through the integration of traditional and machine learning methodologies. Because of the intricacy of today's cyberthreats, a complete solution is required that can detect malicious activity and adapt to the ever-changing tactics employed by cyber adversaries.

This work aims to investigate the performance of various machine learning models, from more complex methods like ensemble techniques like Random Forests to more conventional classifiers like K-Nearest Neighbors and logistic regression. Furthermore, the study looks at how model performance is affected by dimensionality reduction methods, particularly Principal Component Analysis (PCA). This paper is important because it has the potential to further cybersecurity analytics by giving us a better knowledge of the advantages and disadvantages of different models when it comes to threat identification. The study aims to provide insights that will help firms design security solutions that are more robust and adaptable in the face of an ever-expanding threat landscape.

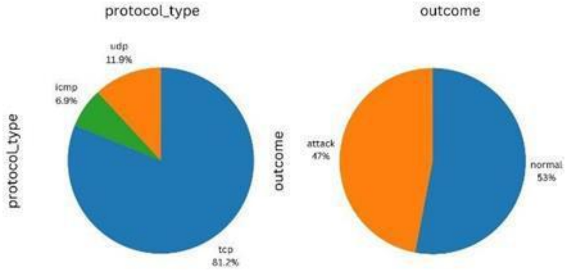


Fig 1. Type of Attack

2. Related Works

[1] Network intrusion detection systems (NIDSs) that are based on machine learning (ML) use flow characteristics that are derived from flow exporting protocols, such as NetFlow. Avg. packet size and other flow information are assumed to be gathered from every packet in the flow by ML and Deep Learning (DL) based NIDS solutions, which have recently shown success. In actuality, flow exporters are frequently used on commodity devices where packet sampling is unavoidable. Therefore, it is unclear if such machine learning- based network intrusion detection systems are applicable when sampling is present (that is, when flow information is gathered from a sampled group of packets rather than the entire traffic). We investigate how packet sampling affects the effectiveness and performance of ML-based NIDSs in this work. In contrast to earlier research, Our suggested evaluation process is unaffected by the various flow export stage settings. As a result, even with sampling, it can offer a reliable assessment of NIDS. We found through sample studies that even at modest sampling rates like 1/10 and 1/100, malicious flows with smaller size (i.e., number of packets) are likely to go undetected. We then looked at the effects of different sampling strategies on the NIDS detection rate and false alarm rate using the suggested evaluation process. The computation of detection rate and false alarm rate is done for four distinct sampling methodologies, three sample rates (1/10, 1/100, and 1/1000), and three (two tree based, one classifier based on deep learning. According to experimental findings, non-linear samplers like Sketch Guided and Fast Filtered sampling perform worse than the systematic linear sampler Sket Flow. Additionally, we discovered that the random forest classifier combined with SketchFlow sampling performed better. In comparison to previous sampler-classifier combinations, the combination demonstrated a greater detection rate and a reduced false alarm rate across numerous sampling rates. Our findings hold true for a variety of sample rates; however, Sketch Guided sample (SGS) exhibits a unique example where a change in sampling rate from 1/100 to 1/1000 resulted in a sharp decline in performance. Our findings offer scholars and network practitioners important new understandings into how packet sampling affects the performance of ML-based NIDS. In this sense, the complete source code for the ML and sampling experiments has been made available.

Researchers and network operators have both recently become interested in the traffic classification challenge. Numerous machine learning (ML) techniques have been put out in the literature as a possible remedy for this issue. Remarkably few research has used Sampled NetFlow data to study the traffic classification challenge. On the other hand, network operators use Sampled NetFlow as a broadly extended monitoring solution. Our goal in this paper is to close this gap. Firstly, we modify a widely used ML-based technique and use NetFlow to examine the performance of existing ML methods. The findings indicate that while the modified approach may achieve approximately 90% accuracy, which is comparable to earlier packet-based approaches, it significantly loses accuracy when sampling occurs. To mitigate this effect, we provide network operators a solution that can function with Sampled NetFlow data and maintain good accuracy even when sampling is present.

[2] Biased sampling techniques are employed by numerous iterations of the Rapidly Exploring Random Tree (RRT) algorithm to address computationally demanding jobs. Planning safe routes while simultaneously intervening in the vehicle's longitudinal and lateral dynamics in intricate traffic scenarios involving several static and moving objects is one of these difficulties. Utilizing an RRT variation known as the Augmented CL-RRT algorithm in conjunction with a 3D convolutional neural network (3D- ConvNet) to forecast appropriate longitudinal acceleration profiles is a recently proposed hybrid statistical learning approach. When there are more than four dynamic objects in a traffic scenario, the algorithm is ineffective due to biasing and a lack of flexibility in the lateral dynamics intervention and the longitudinal dynamics intervention, respectively. Consequently, an expansion of the Augmented CL-RRT algorithm—known as the Augmented CL-RRT+ algorithm—is presented to enhance the longitudinal dynamics intervention with actuator and stable profile limitations. Based on the anticipated longitudinal acceleration and steering wheel angle profiles given by a trained 3D-ConvNet, a biased-sampling approach is also suggested. In order to evaluate various trajectory planning algorithms based on effectiveness and safety, simulations are run. Significant gains in efficiency without compromising safety are demonstrated by the results.

[3] In network security monitoring and traffic engineering, precise and timely traffic classification is essential. When it comes to packet encapsulation and dynamic port allocation, conventional techniques based on protocols and port numbers have been shown to be ineffective. However, the signature matching algorithms can only handle a certain number of IP packet signatures in real-time and require a known signature set in addition to processing

the packet payload. This research proposes an accurate machine learning strategy for classifying Internet traffic using supporting vector machines (SVMs). The technique uses the network flow parameters that are derived from the packet headers to categorize Internet traffic into broad application categories. Many classifier selection techniques are used to get an optimum feature collection. Based on traffic from the campus backbone, experimental results demonstrate that frequent biased training and testing samples can reach an accuracy of 99.42%. Using the identical feature set and unbiassed training and testing samples yields an accuracy of 97.17%. Furthermore, the suggested method may be applied to encrypted network traffic because all feature parameters can be computed from the packet headers.

[4] Deep packet inspection (DPI) and intrusion detection systems (IDS) are frequently used to identify network irregularities and attacks, improving cyber-security. IDS and other traditional traffic analyzers are fixed-site devices with a finite capability for DPI on massive amounts of network traffic. These days, full or partial data traffic flows can be captured on SDN- capable switches and directed to one of the network's traffic analysers thanks to software- defined networking (SDN) technology, which offers flexibility, elasticity, and programmability by separating the network control and data planes. Consequently, two of the most important issues facing cyber-security are where to direct the traffic sampled from the network among several traffic analysers and how to sample network traffic. The selection of network traffic sampling sites and rates is still crucial since there is a chance that otherwise valuable information may be lost in uncaptured traffic. Additional network delivery overheads might result from sending sampled traffic to one of the several traffic analysers for traffic inspection after the sampling points and rates have been established. Using a deep deterministic policy gradient (DDPG), a representative deep reinforcement learning (DRL) algorithm for continuous action control, we offer a less intrusive traffic sampling mechanism for numerous traffic analysers on an SDN-capable network. With sampled traffic inspection findings from several traffic analysers, the suggested system learns sampling resource allocation policy under flow distribution uncertainty. We show that the suggested method has a high likelihood of capturing malicious flows while keeping a balanced load of numerous traffic analyzers and lowering flow monitoring overheads through thorough simulations and the SDN based testbed tests.

3. Methodology

Firstly, a dataset representing network traffic is collected, encompassing both normal and anomalous activities. Then, different sampling techniques such as uniform sampling, stratified sampling, or adaptive sampling are applied to create subsets of the original dataset. Machine learning algorithms, ranging from traditional methods like decision trees and support vector machines to deep learning models like convolutional neural networks or recurrent neural networks, are trained and evaluated on each sampled dataset. Performance metrics such as accuracy, precision, recall, and F1score are used to assess the effectiveness of each sampling approach in enhancing the detection capabilities of the machine learning models. Additionally, the impact of sampling on computational resources and training time is also considered.

3.1 Dataset Description

The study uses a large cybersecurity dataset that was obtained from kaggel. The collection includes a wide variety of network activity logs during a certain time period, encompassing both benign and malevolent actions. It has functionality with regard to system logs, network traffic, and other pertinent variables. Understanding the nuances of cyber risks starts with a thorough examination of the dataset.

3.2 Preprocessing Steps

The dataset goes through a number of preparation procedures to guarantee the models' effectiveness. To make the data more homogeneous, these procedures include feature scaling and normalization. The most informative characteristics are captured while lowering computational complexity through the application of Principal Component Analysis (PCA) for dimensionality reduction. Appropriate encoding is used for categorical variables, and well- established imputation methods are applied to missing data.

3.3 Methods And Models

To assess their efficacy in cyber security threat approaches like Random Forests and sophisticated techniques like nvolutional Neural Networks (CNNs) are included. identification, the study makes use of a range of machine learning algorithms. Among these models are well-known classifiers like Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Logistic Regression. Furthermore, to capture complex patterns inherent in cyber threats, ensemble

4. Model Implementation

The implementation of these models relies heavily on the quality and quantity of data samples used for training. Different sampling techniques, such as uniform sampling, stratified sampling, or random sampling, can significantly impact the performance and robustness of the model.

4.1 Logistic Regression

For the purpose of differentiating between benign and malevolent network activity, the linear model known as logistic regression works effectively. A training accuracy of around 87.88% and a test accuracy of 87.63% were attained by the model in this investigation. Based on a threshold, the decision boundary is established by using Logistic Regression to compute the likelihood that an instance belongs to a specific class. It may not be able to capture intricate linkages, but its interpretability and simplicity make it a useful starting point model for identifying cyber security threats.

4.2 K-Nearest Neighbors (Knn)

A non-parametric technique called K-Nearest Neighbors uses the majority class of their k- nearest neighbors to classify instances. Our research showed that a KNN with 20 neighbors may attain impressive accuracy—roughly 99.05% in training and 98.94% in testing. KNN works best in situations when patterns are locally clustered since it depends on the closeness of data points in the feature space. But for big datasets, it might be computationally costly and sensitive to unimportant factors.

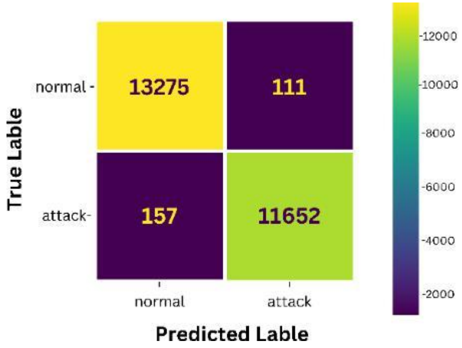


Fig 2. KNN Confusion matrix

4.3 Gaussian Naive Bayes

Gaussian Neural Network Based on the premise of feature independence and the Bayes theorem, the Bayes classifier is probabilistic. Our investigation revealed that it had a 91.80% training accuracy and a 91.61% test accuracy. The simplicity and efficiency of this approach make it a good competitor for cybersecurity applications, and it is especially helpful for handling high- dimensional data.

4.4 Support Vector Machines (Linear Svc)

Support in Linear Form Strong classifiers called vector machines can identify the best hyperplane to divide several classes into. In our investigation 96.46% training and 96.29% test accuracy were attained with Linear SVC. SVMs can handle complicated decision boundaries and perform well in high-dimensional domains. However, with huge datasets, their performance could deteriorate.

4.5 Decision Tree

Decision Trees create a tree structure by iteratively splitting the dataset according to feature requirements. With a maximum depth of three, our Decision Tree Classifier produced remarkable results: 99.99% accuracy during training and 99.88% accuracy during testing. The structure of decision trees can provide information about the significance of particular features and the processes involved in reaching decisions.

4.6 Random Forest

An ensemble technique called Random Forest constructs many decision trees and aggregates their forecasts. It demonstrated exceptional accuracy in our investigation, surpassing 99.88% in training and testing. The benefit of Random Forest is that it aggregates predictions from different trees, which lowers over fitting and enhances generalization

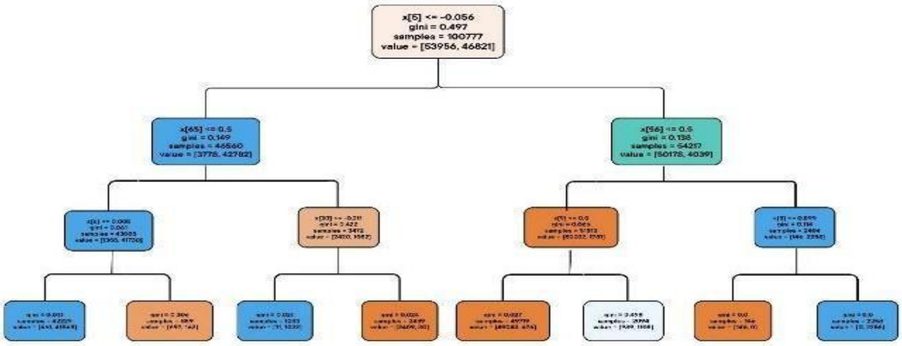


Fig 3. GINI Tree

4.7 Xgboost Regressor

For regression problems, the gradient boosting technique XGBoost is employed. The XGBoost Regressor in our investigation produced training and test errors of 2.82 and 2.82, respectively. A strong model is created by combining weak learners using gradient boosting, and its prediction performance is further improved by the XGBoost regularized objective function.

4.8 Convolutional Neural Network (CNN)

In this study, we modified a deep learning model called a convolutional neural network from one intended for picture data to one tailored for tabular data. Using dropout and several thick layers, the CNN architecture produced a validation accuracy of 97.70% across ten epochs. Because CNNs are capable of autonomously learning hierarchical features, they are well suited for high dimensional, complicated datasets such as cybersecurity logs.

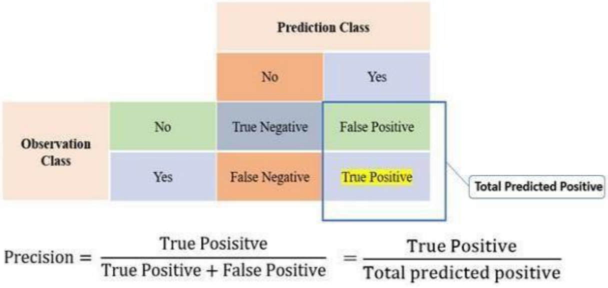
5 Evaluation

ACCURACY- The ratio of successfully predicted instances to all instances is used to compute accuracy, which gauges the model's overall correctness.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{All Samples}}$$

Eqn ...(1)

PRECISION - The accuracy of positive predictions, which shows how well the model can evade erroneous positives, is the main emphasis of precision



		Prediction Class	
		No	Yes
Observation Class	No	True Negative	False Positive
	Yes	False Negative	True Positive

$$\text{Precision} = \frac{\text{True Posisitve}}{\text{True Positive} + \text{False Positive}} = \frac{\text{True Positive}}{\text{Total predicted positive}}$$

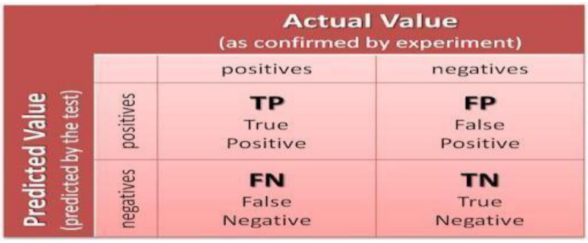
Fig 4. Precision Table

RECALL (SENSITIVITY) The model's recall quantifies its capacity to extract all pertinent examples of a positive class.

$$Recall = \frac{True\ Positive(TP)}{True\ Positive(TP) + False\ Negative(FN)}$$

Eqn ... 2

CONFUSION MATRIX A comprehensive analysis of the model's predictions is given by a confusion matrix, which displays the quantity of true positives, true negatives, false positives, and false negatives.



		Actual Value (as confirmed by experiment)	
		positives	negatives
Predicted Value (predicted by the test)	positives	TP True Positive	FP False Positive
	negatives	FN False Negative	TN True Negative

Fig 5. Confusion Matrix

6. Result and Discussions

We have obtained illuminating results from our testing with several machine learning models for cybersecurity threat identification. Different models performed differently, including Principal Component Analysis (PCA) combined with Random Forest, Decision Trees, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Gaussian Naive Bayes, Random Forest, and a neural network architecture using TensorFlow and Keras.

A training accuracy of around 87.88% and a test accuracy of 87.63% were obtained by the use of logistic regression. It was a strong contender for threat detection because of its impressive accuracy and recall ratings.

A test accuracy of 98.94% and a training accuracy of 99.05% demonstrated the superior performance of K-Nearest Neighbors (KNN). The model's aptitude for identifying patterns was demonstrated by its consistently excellent accuracy and recall ratings.

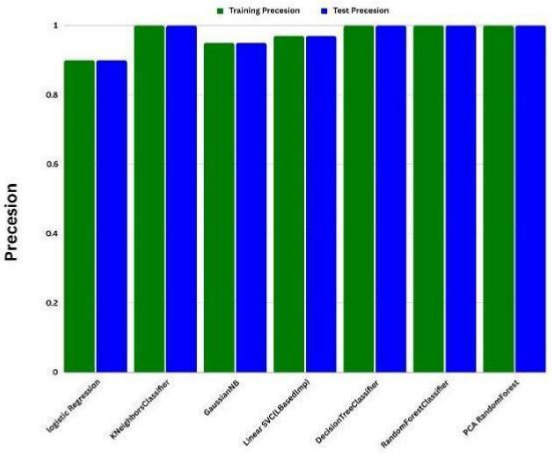


Fig 6. Algorithm Comparison

At 91.61% for test accuracy and 91.80% for training, Gaussian Naive Bayes demonstrated strong performance. Further evidence of its efficacy in differentiating between benign and malevolent activity came from its accuracy and recall ratings. A Linear SVC implementation of Support Vector Machines (SVM) produced impressive results, with a training accuracy of 96.46% and a test accuracy of 96.29%. Its accuracy and recall ratings demonstrate how trustworthy it is in spotting dangers. At 99.99% for training and 99.88% for testing, Decision Trees demonstrated impressive accuracy. The depiction of feature importances and the tree structure improve the interpretability of the decision tree. Random Forest achieved a test accuracy of 99.89% and a training accuracy of 99.99%, considerably improving performance. Understanding the crucial elements impacting the model's predictions was made possible by the feature importances. Dimensionality reduction effectively maintained high accuracy (training: 99.99%, test: 99.84%) when Principal Component Analysis (PCA) and Random Forest were used together.

Each model's strengths and flaws may be fully understood thanks to the complete evaluation, which takes into account factors like accuracy, precision, recall, and decision boundary visualization. These findings open up the possibility of making well-informed choices when deciding which model is best for cybersecurity threat detection given particular constraints and trade-offs. Our research offers insightful information for future studies and useful applications, adding to the expanding corpus of knowledge in the fields of machine learning and cybersecurity.

7. Conclusion

Ultimately, the exploration of machine learning models for cybersecurity threat identification has uncovered a complex environment in which many algorithms demonstrate unique capabilities and outcomes. A thorough grasp of their capabilities was offered by the group of models, which ranged from more sophisticated ones like Support Vector Machines, Decision Trees, Random Forest, and neural networks to more conventional ones like Logistic Regression, KNearest Neighbors, and Gaussian Naive Bayes. The ability of models such as Decision Trees and Random Forest to distinguish between benign and malevolent behavior is demonstrated by their high accuracy. A better comprehension of the underlying patterns is made possible by the interpretability of decision trees, which also provides transparent insights into feature significance. Principal Component Analysis (PCA) in conjunction with Random Forest demonstrated the feasibility of dimensionality reduction while retaining accuracy, hence resolving issues with computing effectiveness and resource consumption. It achieved competitive accuracy in the investigation of a neural network architecture with TensorFlow and Keras, introducing a contemporary method. The array of models evaluated gains a vital dimension from the neural network's capacity to capture complex patterns and correlations in the data. A more nuanced knowledge of the models' performances is made possible by the thorough evaluation, which includes accuracy, precision, recall, and visual representations. A basis for well-informed decision-making is provided by each model's distinct qualities, advantages, and possible drawbacks when choosing a suitable model for certain cybersecurity applications. Machine learning models' resilience and flexibility are becoming more and more important as cybersecurity threats continue to change. By providing light on the relative effectiveness of various models, our research advances this dynamic area and benefits practitioners, scholars, and policymakers with insightful information. The results open up new possibilities for machine learning and cybersecurity research and development, resulting in a more secure digital environment.

References

1. Alikhanov, J., Jang, R., Abuhamad, M.,Mohaisen, D., Nyang, D., & Noh, Y., “ Investigating the effect of traffic sampling on machine learning-based network intrusion detection approaches”. IEEE Access, 10, 58015823. 2021
2. Carela-Español, V., Barlet-Ros, P., Cabellos-Aparicio, A., & Solé-Pareta,, “ Analysis of the impact of sampling on NetFlow traffic classification”. Computer Networks, 55(5), 1083-1099.2011
3. Chaulwar, A., Botsch, M., & Utschick, W. , “A machine learning based biased- sampling approach for planning safe trajectories in complex, dynamic traffic scenarios”. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 297-303). IEEE.2017
4. Yuan, R., Li, Z., Guan, X., & Xu, L., “ An SVM-based machine learning method for accurate internet traffic classification. Information Systems Frontiers”, 12, 149-156. 2010
5. Kim, S., Yoon, S., & Lim, H., “ Deep reinforcement learning-based traffic sampling for multiple traffic analyzers on software defined networks”. IEEE Access, 9, 4781547827. 2021
6. Nguyen, T. T., & Armitage, G., “A survey of techniques for internet traffic classification using machine learning”. IEEE communications surveys & tutorials, 10(4), 56-76. 2008
7. Carela-Espanol, V., Barlet-Ros, P., SoléPareta, J., “Traffic classification with sampled netflow. Traffic”, 33, 34. 2009.
8. Jin, Y., Duffield, N., Erman, J., Haffner,P., Sen, S., & Zhang, Z. L., “A modular machine learning system for flow-level traffic classification in large networks”. ACM Transactions on Knowledge Discovery from Data (TKDD), 6(1), 1-34. 2012
9. Arndt, D. J., & Zincir-Heywood, A. N. , “ comparison of three machine learning techniques for encrypted network traffic analysis”. In 2011 IEEE symposium on computational intelligence for security and defense applications (CISDA) (pp. 107-114). IEEE. 2011
10. Jadav, N., Dutta, N., Sarma, H. K. D., Pricop, E., & Tanwar, S., “A machine learning approach to classify network traffic”. In 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI) (pp. 1-6). IEEE. 2021
11. Singh, R., Kumar, H., & Singla, R. K., “Sampling based approaches to handle imbalances in network traffic dataset for machine learning techniques”. arXiv preprint arXiv:1311.2677. 2013.

12. Krasniqi, F., Elias, J., Leguay, J., & Redondi, A. E., “End-to-end delay prediction based on traffic matrix sampling”. In IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 774-779). IEEE. 2020.
13. Li, W., & Moore, A. W. ,”A machine learning approach for efficient traffic classification”. In 2007 15th International symposium on modeling, analysis, and simulation of computer and telecommunication systems (pp. 310-317). IEEE. 2007.
14. Knapińska, A., Lechowicz, P., & Walkowiak, K., “Machine-learning based prediction of multiple types of network traffic”. In International Conference on Computational Science (pp. 122-136). Cham: Springer International Publishing. 2021.
15. Singh, R., Kumar, H., & Singla, R. K., “Issues related to sampling techniques for network traffic dataset”. International Journal of Mobile Network Communications & Telematics, 3(4), 75-85. 2013.