Defence Research and Development Canada
Recherche et développement pour la défense Canada

DRDC | RDDC
technologysciencetechnologie

Traffic analysis on encrypted traffic over wireless channels

Traffic classification based on partial knowledge

Ronggong Song
Tricia Willink
DRDC – Ottawa Research Centre

Defence Research and Development Canada
Scientific Report
DRDC-RDDC-2018-R196
November 2018

Canada

IMPORTANT INFORMATIVE STATEMENTS

This document was reviewed for Controlled Goods by Defence Research and Development Canada (DRDC) using the Schedule to the *Defence Production Act*.

Endorsement statement: This publication has been peer-reviewed and published by the Editorial Office of Defence Research and Development Canada, an agency of the Department of National Defence of Canada. Inquiries can be sent to: Publications.DRDC-RDDC@drdc-rddc.gc.ca.

Abstract

This Scientific Report investigates the limitations of traditional supervised traffic classification techniques on classification performance such as accuracy, precision, and recall when there is only limited knowledge available about the traffic, especially an adversary's encrypted traffic. To improve the classification performance under the above scenario, a new modified naïve Bayes kernel (MNBK) classifier is proposed based on optimal weight-based (OWB) kernel bandwidth selection. The proposed OWB kernel bandwidth selection algorithm can make a more accurate learning model for traffic classification compared with the traditional classifiers. By generating several possible major traffic types in tactical edge networks, we demonstrate that the proposed MNBK classifier not only improves the classification performance on the existing classes significantly, but also detects unknown traffic with very high accuracy, precision, and recall compared with the traditional classifiers. In addition, a learning classification model is proposed based on MNBK, that processes received ongoing real time traffic and updates the classification table periodically. Generally speaking, with more and more accurate information retrieved from received real time traffic, the proposed real time classification model should improve the classification performance over time compared with the traditional classifiers that do not consider the ongoing received traffic. This has been demonstrated with our classification performance evaluation.

Significance to defence and security

Traffic analysis plays a very important role in signal intelligence, cyber, and electronic warfare. It has been used to provide valuable intelligence with accurate information about cyber targets. The results from traffic analysis can be used to support decision making in the battlefield such as target selection, cyber effects, smart jamming, or destruction. For this reason, the accuracy of information prediction through traffic analysis becomes a key factor in mission success.

In this Scientific Report, we present a modified naïve Bayes kernel classifier (MNBK) to improve the traffic classification performance of well-known existing supervised traffic classifiers with less knowledge of the adversary's traffic, which is very practical under battle scenarios. The proposed MNBK classifier has much better classification performance in predicting and detecting unknown traffic that does not belong to the classes contained in the training data, and significantly increases the overall classification accuracy compared with the traditional classifiers.

This work is important to the defence and security community. It presents a practical traffic classification approach to improve classification performance and detect unknown traffic with very high accuracy, precision, and recall, as in tactical scenarios where only limited knowledge about the adversary's traffic may be available.

Résumé

Le présent rapport traite des limites au rendement des techniques traditionnelles de classification du trafic supervisé en ce qui a trait à l'exactitude, à la précision et au rappel lorsqu'on dispose de peu d'information sur le trafic, particulièrement sur le trafic crypté d'un adversaire. Le rapport présente également une nouvelle méthode de classification naïve bayésienne modifiée basée sur la sélection de la taille du noyau selon une pondération optimale afin d'améliorer le rendement en matière de classification dans un tel cas. Le nouvel algorithme de sélection de la taille du noyau nous permettra d'instaurer un modèle d'apprentissage pour la classification du trafic qui sera plus précis que les modèles traditionnels. En générant plusieurs grands types de trafic pour les réseaux tactiques en périphérie, la méthode proposée permettra non seulement d'améliorer considérablement le rendement en matière de classification à l'aide des classes existantes, mais également de détecter le trafic inconnu avec beaucoup plus d'exactitude, de précision et de facilité de rappel que les méthodes traditionnelles. Un nouveau modèle de classification par apprentissage est proposé. Celui-ci est fondé sur une méthode naïve bayésienne modifiée qui permettra de traiter le trafic en temps réel et de mettre à jour périodiquement le tableau de classification. Globalement, au fur et à mesure que l'on recueillera de l'information de plus en plus exacte provenant du trafic en temps réel le modèle proposé permettra avec le temps d'améliorer le rendement de la classification par rapport aux modèles traditionnels qui ne tiennent pas compte du trafic continu, comme l'a démontré notre évaluation du rendement en matière de classification.

Importance pour la défense et la sécurité

L'analyse du trafic revêt une très grande importance dans le renseignement d'origine électromagnétique, de même que dans la guerre électronique et cybernétique. Elle fournit de précieux renseignements, entre autres des précisions sur les cybercibles. Les résultats d'analyse du trafic peuvent aider à la prise de décisions sur le champ de bataille, notamment pour sélectionner les cibles, déterminer les conséquences sur le plan informatique, procéder au brouillage intelligent et détruire les cibles. L'exactitude de la prédiction d'information grâce à l'analyse du trafic est devenue un facteur clé dans la réussite des missions.

Dans le présent rapport, on propose une version modifiée d'une méthode naïve bayésienne pour améliorer le rendement obtenu grâce aux méthodes supervisées existantes connues, lesquelles nécessitent moins de connaissance du trafic de l'adversaire. Ceci s'avère fort pratique dans les scénarios de bataille. Comparativement aux modèles traditionnels, celui qui est proposé affiche un meilleur rendement en matière de classification, permet de détecter et de prévoir plus facilement le trafic inconnu qui n'entre pas dans les classes de données d'apprentissage et d'améliorer considérablement l'exactitude générale de la classification.

Les travaux décrits dans le présent rapport sont importants pour la communauté du domaine de la défense et de la sécurité. Ils proposent une démarche pratique pour améliorer le rendement en matière de classification, de détection du trafic inconnu, d'exactitude, de précision et de facilité de rappel, comme dans les scénarios tactiques dans lesquels on ne dispose que de peu d'information sur le trafic de l'adversaire.

Table of contents

List of figures

List of tables

1 Introduction

Traffic analysis is "the process of intercepting and examining messages in order to deduce information from patterns in communication" [1]. It is a very important and basic part of signal intelligence and electronic warfare, and has been used to provide valuable intelligence with accurate estimation of the targets' intentions and actions such as their locations, movements, roles, network structure, and communication patterns by intercepting and examining the adversary's network communication. With advances in processing speed, traffic analysis can obtain more in-depth knowledge on what type of traffic packets and data are flowing through a network even when the traffic is encrypted and cannot be decrypted. The information resulting from traffic analysis can be used to support decision making in the battlefield such as target selection, cyber effects, smart jamming, or destruction. Therefore the accuracy of information prediction in traffic analysis is a key factor in mission success.

One of the important techniques used to improve prediction accuracy in traffic analysis is machine learning. In general, the larger and more comprehensive the set of training data has, the more information that can be inferred from the testing (or real time) data. However, effective machine learning is usually very difficult in the battlefield because there is not enough training data and knowledge of the adversary's traffic. In addition, it is hard to find correct and accurate patterns of the traffic if the training data only contains partial knowledge of the traffic. This problem is likely to exist in the battlefield nowadays since most adversaries' traffic is assumed to be protected by encryption or other methods.

One of the research areas in traffic analysis is traffic classification. Traffic classification can be used to support both offence and defence in tactical networks such as protocol attacks and automated intrusion detection, etc. Traditional conventional approaches in traffic classification rely on the deep inspection of a packet's header information (e.g., IP address, port number) [2–4] or payload information (e.g., protocol signatures) [3–5]. Nowadays, this information is usually protected by the adversary's network and the only available information intercepted from the adversary's network might be packet timing and length. By studying the traffic's statistical properties with these limited data, many new traffic classification technologies combined with machine learning techniques were proposed in the last decade [6–16], where Bayesian methods [11–16] have become mainstream and known as simple yet effective. As we mentioned above, with limited training data and knowledge of the adversary's traffic, accurate, real time, and continuous traffic classification are still open challenges.

In this report, in order to improve the traffic classification accuracy, precision, and recall, we investigate the limitations of the naïve Bayesian classifier [11–13] and Gaussian kernel density estimation (KDE) [17] in traffic classification based on packet lengths. In contrast to the traditional naïve Bayes kernel estimation (NBK) [13], which uses rules of thumb [18] or optimal solve-the-equation plug-in [17, 19] approaches for bandwidth selection in KDE, we propose an optimal weight-based (OWB) method in bandwidth selection for constructing a classification model with a Gaussian kernel distribution. We adapt the supervised NBK approach to detect unknown traffic. By selecting several possible major traffic types (generated by the BreakingPoint™ traffic generator [20]) in tactical edge networks, we demonstrate that the new proposed traffic classification technology not only improves the prediction accuracy, precision, and recall significantly but also detects unknown traffic. We further discuss how to use the new proposed traffic classification technology for real-time and continuous classification, and its related performance.

The rest of the report is organized as follows. The background and related work on traffic analysis and classification, machine learning, and naïve Bayes kernel estimation are briefly introduced in the next section. In Section 3, a modified naïve Bayes kernel classifier (MNBK) is presented based on a new optimal weight-based kernel bandwidth selection algorithm. In addition, a classification learning model based on probability rather than probability density, and new traffic detection and prediction is introduced into the proposed MNBK classifier. In Section 4, the traffic classification performance of the proposed MNBK classifier is evaluated and compared with other traditional NBK classifiers based on encrypted wireless traffic generated with the BreakingPoint traffic generator. In Section 5, the application of MNBK to both real time and continuous traffic classification is discussed. Finally, concluding remarks are given in Section 6.

2 Background and related work

In this section, we briefly introduce some background and related research on traffic analysis, traffic classification, machine learning, and naïve Bayes classifier with Gaussian kernel density distribution.

2.1 Traffic analysis and classification

Traffic analysis has been used in the military since World War I when the telegraph was used in communication [21]. For instance, during First World War, British Room 40 got very valuable information about German ships, such as their positions and major fleet operations, by intercepting and examining signals and wireless traffic among German ships, Zeppelins, and shore stations. The French Army identified German combat groups and their locations by intercepting and comparing different call-signs, signal strength, and activity categorizations. At that time, the British and French signal intelligence organizations had the advantage of thorough knowledge about the German communication systems.

Traffic analysis played a significant role in signals intelligence when wireless communication became popular in the military. Back in World War II, traffic analysis provided more valuable intelligence and accurate predication at recognizing the fingerprints of devices and operators. For example, in 1941 British identified and confirmed that a German Air Force unit contained nine planes by intercepting and reconstructing the network structure of the German Air Force radio, which gave very accurate estimation of the strength of the German Air Force [22].

Along with advanced research and the development of computer and machine learning technologies, nowadays traffic analysis is used to support both offence and defence in the cyber security domain. For instance, traffic analysis has been widely used in automated intrusion detection systems, reconstruction of the adversary's tactical networks, target selection, smart jamming, etc. [23–27]. In addition, with more and more advanced security protection applied in adversaries' network communications, it becomes much harder than before to get deep knowledge of the packet contents just by inspection of the intercepted traffic packets directly. Traffic analysis becomes a more and more important tool to gain more knowledge about adversaries' network communications without effort in breaking their security systems such as encryption. Traffic classification is one of the important traffic analysis tools for detecting and predicting known and unknown traffic types (or patterns) based on statistical properties of network communications.

2.2 Machine learning in traffic classification

Machine learning is known as a powerful technique for data mining and knowledge discovery. It has been used in many applications such as medical diagnosis [28–29], handwriting pattern recognition [30–31], search engines [32–33], traffic classification [6–16], and so on. There are several basic types of machine learning techniques used in traffic classification. For instance, supervised, unsupervised, rule-based, and so on, where supervised and unsupervised learning are the major learning techniques used in traffic classification [4]. In this work, we focused on a mainstream traffic classification technique—naïve Bayes classification which uses supervised learning to construct a learning model and classify traffic. In addition, we use only the packet length as raw data to evaluate the performance of the proposed MNBK classifier, i.e., how much the new classifier can improve on the traffic classification compared with the traditional methods.

Supervised machine learning consists of two phases: a training phase and a testing phase. In the training phase, supervised learning techniques construct a classification model based on the provided training dataset. In the testing phase, the classification techniques predict new instances based on the classification model built in the training phase. The classification model is constructed based on a function which maps input features from the training dataset to output classes, i.e., the output classes are pre-defined in the training dataset and controlled by the function. This function is the key to the classification model in supervised machine learning. To find a function that can best predict the results for any new instances is a challenge.

There are many supervised learning classification algorithms which use different ways to construct classification models, for instance, naïve Bayes Tree [34], Bayesian network [35], naïve Bayes with kernel density estimation [13], C4.5 decision tree [36]. In this work, we investigated naïve Bayes kernel estimation in traffic classification, which can construct a multimodal classification model to reflect the real distribution of more complex data with multiple modes.

2.3 Naïve Bayes classifier with kernel distribution

The naïve Bayes kernel density classification method is a generalization of the naïve Bayes classifier and uses the Gaussian kernel distribution for probability estimation instead of using the standard Gaussian (or normal) distribution. The naïve Bayes classifier with the normal distribution has limitations to construct multimodal classification models for complex data with multiple modes.

2.3.1 Naïve Bayes classifier

The naïve Bayes classifier is a simple classification technique based on applying Bayes Theorem with independence assumptions between features. The theory of naïve Bayes applied to traffic classification is explained below.

Consider a training dataset containing n traffic packet samples $X = \{x_1, x_2, \ldots, x_n\}$ where each packet x_i may have multiple attributes such as length, timing, and so on. Assume there are k known classes $C = \{c_1, c_2, \ldots, c_k\}$ in the training dataset X. Based on the assumptions such as naïve independence and Gaussian (normal) distribution of the attributes in the naïve Bayes classifier, the following probability density equation[1] is used to estimate the probability density of a traffic packet y which comes from a testing dataset

$$f(y|c_i) = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(y-\mu_i)^2}{2\sigma_i^2}} \qquad \text{for} \quad i = 1, 2, \ldots k \tag{1}$$

where μ_i and σ_i are the mean and standard deviation of the packets belonging to class c_i and are calculated based on the training dataset with the following equations

[1] This density equation is based on standard normal distribution, i.e., each traffic packet contains one attribute only. For packets containing multiple attributes, the more complex multivariate normal distribution is required (see more details in [37]).

$$\mu_i = \frac{1}{N_{c_i}} \left(\sum_{x_s : C(x_s) = c_i} x_s \right) \qquad \text{for} \quad i = 1, 2, \ldots k \tag{2}$$

and

$$\sigma_i = \sqrt{\frac{\sum_{x_s : C(x_s) = c_i} (x_s - \mu_i)^2}{N_{c_i}}} \qquad \text{for} \quad i = 1, 2, \ldots k \tag{3}$$

where $C(x_s) = c_i$ stands for the instance x_s belonging to the class c_i and N_{c_i} is the total number of traffic packets belonging to the class c_i in the training dataset, which means $\sum_{i=1}^{k} N_{c_i} = n$.

Based on Bayes Theorem, the posterior probability that a testing traffic packet y belongs to the class c_i (denoted by $P(c_i|y)$) can be calculated with the following equation[2]

$$P(c_i|y) = \frac{p(c_i) * f(y|c_i)}{\sum_{j=1}^{k} p(c_j) * f(y|c_j)} \qquad \text{for} \quad i = 1, 2, \ldots k \tag{4}$$

where $p(c_i)$ is the prior probability of the class c_i calculated with the following equation based on the training dataset

$$p(c_i) = \frac{N_{c_i}}{n} \qquad \text{for} \quad i = 1, 2, \ldots k \tag{5}$$

In the testing dataset, a testing traffic packet y is classified into a class c_i if $P(c_i|y) = \max\{P(c_j|y): j = 1, 2, \ldots, k\}$.

Although the naïve Bayes classifier has worked quite well in many real-world situations [38], it would cause large errors if the real situation is multimodal, i.e., it does not even approximately satisfy the assumption of a Gaussian distribution. For instance, the Figure 1 shows the relative frequency distribution of the JabberChat traffic[3] and the probability density calculated under normal distribution, i.e., JabberChat fitted normal density. Obviously, the Jabberchat fitted normal distribution (red line) in Figure 1 does not match the JabberChat traffic (blue line). One of the solutions to construct a multimode model and match the real traffic in machine learning is to use Gaussian kernel density distribution, which will be discussed in Section 2.3.2.

[2] Actually Equation (4) is not really correct because f(y|ci) calculated with Equation (1) is a probability density, rather than a probability. However, many popular machine learning tools (e.g., RapidMiner) use it that way. We will use probability instead of probability density in our new method in this work.

[3] The JabberChat traffic is generated with the BreakingPoint traffic generator. It is encrypted and forwarded from one wireless router to another. The traffic data is collected through wireless eavesdropping using WiresharkTM and AirpcapTM.

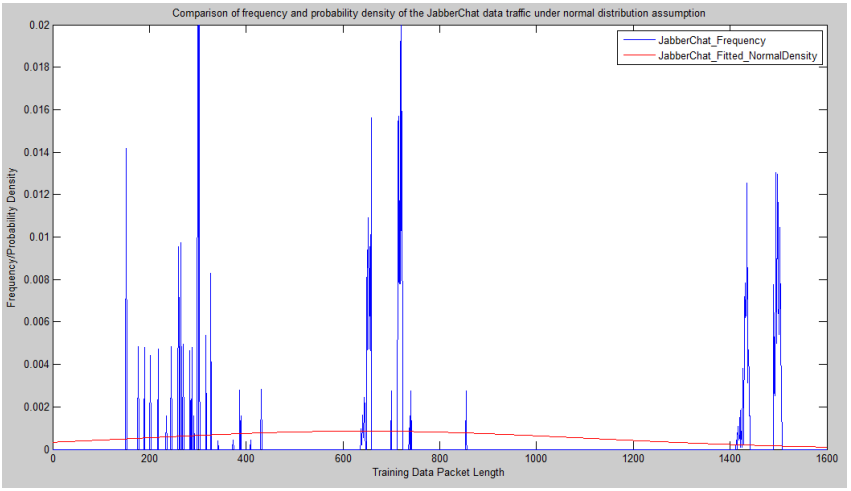


Figure 1: *Comparison of the relative frequency distribution of the JabberChat data traffic with the JabberChat fitted normal density.*

2.3.2 Gaussian kernel density distribution

Naïve Bayes kernel estimation uses the following Gaussian kernel density equation[4] to estimate the probability density of a new testing traffic packet y instead of using the normal density shown in Equation (1)

$$f(y|c_i) = \frac{1}{N_{c_i}} * \sum_{x_s:C(x_s)=c_i} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{\frac{(y-x_s)^2}{2t_i^2}} \right) \quad \text{for } i = 1, 2, \ldots k \tag{6}$$

where t_i is referred as the kernel bandwidth of the Gaussian kernel distribution. The difference in the Gaussian kernel density is that all individual samples (x_s) and the kernel bandwidth (t_i) in the kernel ($\frac{1}{\sqrt{2\pi t_i^2}} e^{\frac{(y-x_s)^2}{2t_i^2}}$) are used for the probability density calculation instead of using the mean (μ_i) and standard deviation (σ_i) in the normal density. The naïve Bayes kernel density classification uses Equation (4) for posterior probability calculation as that in the naïve Bayes classifier. As stated in [13], the naïve Bayes kernel classification performs much better in situations when the normality assumption is strongly violated.

In the Gaussian kernel density, the selection of the kernel bandwidth plays a very important role in the accuracy of the probability density construction. Much research [17–18, 40–42] has been done on kernel bandwidth selection in the past. The most popular kernel bandwidth selection algorithms applied in machine learning tools (e.g., Matlab, RapidMiner) are rules-of-thumb (ROT) based on [18] and solve-the-equation (STE) based on [17]. We will discuss the kernel bandwidth selection in Section 3.1.

[4] The Gaussian kernel density equation is used for one-dimensional traffic data. For packets containing multiple attributes (m-dimensions), the more complex nonparametric kernel density distribution is required (see more details in [39]).

3 Modified naïve Bayes kernel classifier

In this section, we discuss some limitations in the traditional kernel bandwidth selection of the naïve Bayes kernel classifier for traffic with packet length data, which we can collect from encrypted wireless channels without decrypting the packet. We propose a new weight-based kernel bandwidth selection algorithm to improve the kernel probability density pattern constructed by the training data and to construct the learning model with probability rather than probability density. At the end of this section, we introduce our methods for "unknown" [5] traffic detection in our modified naïve Bayes kernel classifier.

3.1 Traditional kernel bandwidth selection

In the naïve Bayes kernel classifier, the kernel bandwidth is a key factor for constructing an accurate probability density model. The most popular kernel bandwidth selection algorithms are rules-of-thumb [18] and solve-the-equation [17], which are applied in many machine learning tools. They are briefly described below.

1. Rules-of-thumb (ROT) kernel bandwidth (t_i) selection is based on the following equation

$$t_i = \left(\frac{4}{3N_{c_i}}\right)^{\frac{1}{5}} * \sigma_i \qquad \text{for} \quad i = 1, 2, \dots k \tag{7}$$

   where σ_i can be calculated from Equation (3), i.e., using the standard deviation. It can also be calculated with the following equation, i.e., using median absolute deviation (MAD)

$$\sigma_i = 1.4826 * MAD\{x_j : C(x_j) = c_i\} \qquad \text{for} \quad i = 1, 2, \dots k \tag{8}$$

   We call ROT bandwidth calculated with standard deviation "ROTs," and the ROT bandwidth calculated with median absolute deviation "ROTm." ROTm usually performs better than ROTs since MAD is a more robust statistic and is more resilient to outliers in a dataset than the standard deviation.

2. Solve-the-equation (STE) kernel bandwidth selection is based on the following equation

$$t_i = \left(\frac{1}{2N_{c_i}\sqrt{\pi}||f''||^2}\right)^{\frac{2}{5}} \qquad \text{for} \quad i = 1, 2, \dots k \tag{9}$$

   where $||f''||$ is estimated by the L-stage direct plug-in bandwidth selector (see details in [17]).

[5] Here "unknown" traffic means that the traffic does not belong to any class contained in the training dataset, i.e., new detected traffic. Traditional supervised learning and traffic classification technologies do not provide this capability.

Figure 2 depicts the probability densities of the JabberChat traffic under three kernel bandwidths: ROTs, ROTm, and STE. The figure shows that all three probability densities constructed with different kernel bandwidths reflect multimodal distribution features of the traffic data rather than the single mode constructed with the normal distribution assumption and shown in Figure 1. In addition, there is a huge difference among the three probability densities, which means kernel bandwidth selection has a big impact on constructing an accurate probability density and, further, an accurate learning model.

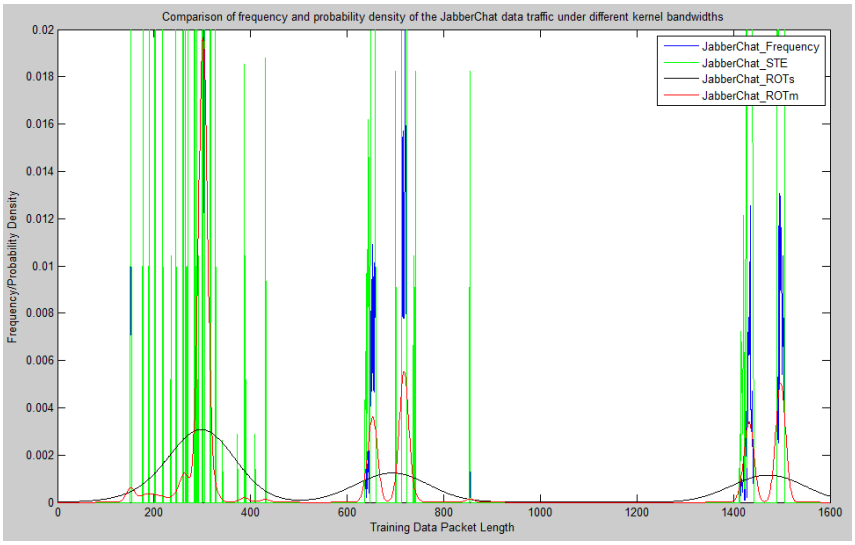


Figure 2: *Comparison of the relative frequency distribution of the JabberChat traffic with the kernel probability density of the traffic under different kernel bandwidths.*

Figure 2 also shows that all kernel bandwidth selections have difficulty in constructing an accurate probability density for traffic classification with packet length data, i.e., they do not match the relative frequency distribution of the traffic.

First, the kernel bandwidth calculated with ROTs is too big, which makes the probability density too smooth and covers almost every packet, although this is better than the normal density shown in Figure 1. A very smooth probability density model can predict that many packets from other classes fall into its class, i.e., increase the false positive rate (or Type I errors).

The probability density constructed based on ROTm is much better than the probability density constructed based on ROTs (see Figure 3 for details) since it is a close match to the relative frequency distribution of the traffic data. The ROTm model is still too smooth compared with the relative frequency distribution of the traffic data and does not match the distribution pattern of the traffic at all, especially the packets with length from 150 bytes to 250 bytes (in Figure 3).

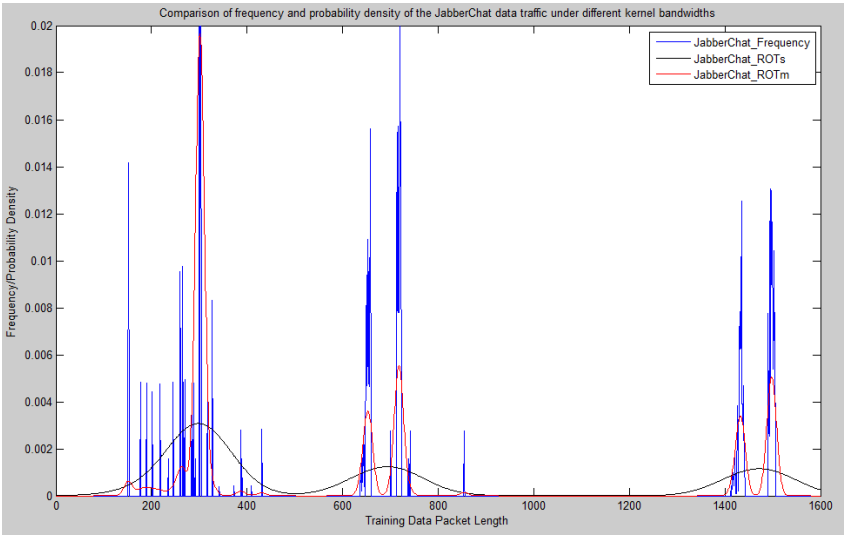
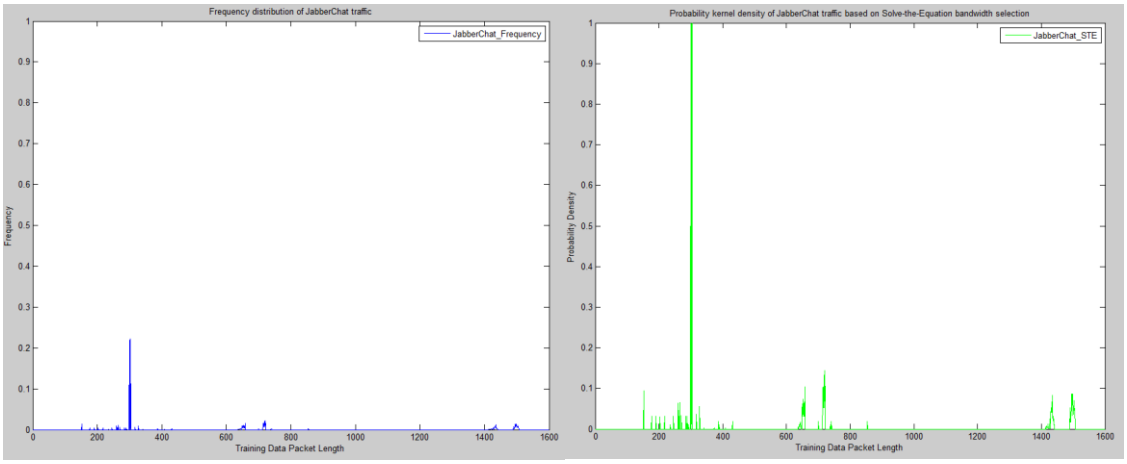


Figure 3: *Comparison of the relative frequency distribution of the JabberChat traffic with the probability kernel density of the traffic based on ROTs and ROTm kernel bandwidths.*

The kernel bandwidth calculated based on STE is too small, which makes the probability density very sharp. The constructed probability density does not match the relative frequency distribution of the traffic well (see Figure 4, which plots the same data from Figure 3 on a larger y-axis). Generally speaking, a very sharp probability density model can wrongly predict that many of its own packets belong to other classes, i.e., increase the false negative rate (or Type II errors).



(a) Relative Frequency Distribution. (b) Probability Density based on STE.

Figure 4: *Comparison of (a) the relative frequency distribution and (b) probability kernel density of the JabberChat data traffic based on STE kernel bandwidth.*

In traffic classification, the main goal is to reduce both false positive rate and false negative rate (i.e., Type I and Type II errors), which results in an increase in accuracy, precision, and recall in traffic classification.

3.2 Optimal weight-based kernel bandwidth selection

In order to make the constructed probability density reflect the relative frequency distribution of the traffic, we propose an optimal weight-based (OWB) kernel bandwidth selection algorithm. The OWB bandwidth selection is a straightforward and simple bandwidth selection method, and consists of three steps: packet weight sorting, major packet selection, and kernel bandwidth estimation.

OWB Step 1: Packet Weight Sorting

Sort packet weights (or relative frequencies) in the training dataset for each class c_i ($i=1, 2, ...k$) based on their packet lengths. Note that in this report, we assume there are k classes in the training dataset, i.e., $i=1, 2, ...k$ for all equations.

a. Assume $X_i = \{x_s: C(x_s) = c_i\}$ are all traffic packets belonging to the class c_i in the training dataset and there are m different packet length $\{l_{i1}, l_{i2}, \ldots, l_{im}\}$ in X_i.

b. Calculate the weights of all packets in X_i based on their packet lengths for the class c_i using the following equation

$$w(l_{ij}) = \frac{N_{l_{ij}}}{N_{c_i}} \qquad \text{for } j = 1,2,\ldots m \tag{10}$$

where $N_{l_{ij}}$ is the total number of packets with packet length l_{ij} in X_i.

c. Sort the weights of all packet lengths in X_i in descending order to get $W_i = \{w(l_{i1}), w(l_{i2}), \ldots, w(l_{im})\}$ where $L_i = \{l_{i1}, l_{i2}, \ldots, l_{im}\}$ are the different packet lengths in X_i whose weights are in descending order.

OWB Step 2: Major Packets Selection

In order to make the constructed probability density match the relative frequency distribution of most traffic packets, we select packets based on the following rules.

a. Set a weight threshold ($W_{th} \geq 50\%$, default is 50%) for major packet selection.

b. Choose packet length from L_i in descending weight order (i.e., l_{ij} for $j = 1, 2, 3, \ldots, m$, depending on how many major packets are selected) based on OWB Step 1 c); calculate the total weight of the packets with length l_{ij} and $l_{ij} \pm 1$, i.e., packets with 1 byte length difference[6] with l_{ij}

$$w_{ij} = \sum_{z=-1}^{1} w(l_{ij} + z) \qquad \text{for } j = 1, 2, \ldots, m \tag{11}$$

[6] Considering only packets with 1 byte length difference has the goal of making the probability density neither too smooth nor too sharp.

Note that weights $w(l)$ are used only once for all j and are counted in the packet weight calculation with the highest weight.

c. Choose the smallest u in the following equation

$$\sum_{j=1}^{u} w_{ij} > W_{th} \qquad \text{for } u = 1, 2, \dots, m \tag{12}$$

and let

$$W_{total} = \sum_{j=1}^{u} w_{ij} \tag{13}$$

All packets with the lengths l_{ij} and $l_{ij} \pm 1$ (for j =1, 2, ..., u) are selected for kernel bandwidth estimation used in OWB Step 3. In this way, over 50% of the packets with high relative frequency are selected for the kernel bandwidth estimation.

OWB Step 3: Kernel Bandwidth Estimation

Based on OWB Step 2, packets with lengths in $\{l_{i1}, l_{i2}, \dots, l_{iu}\}$ and $\{l_{i1} \pm 1, l_{i2} \pm 1, \dots, l_{iu} \pm 1\}$ are selected for our Gaussian kernel bandwidth estimation.

The following probability equation is constructed to estimate the probability of each selected packet l_{ij} (j=1, 2, ..., u) and its neighboring packets $l_{ij} \pm 1$, i.e., packets with 1 byte length difference with l_{ij}, based on the Gaussian kernel distribution and related kernel bandwidth t_i,

$$p(t_i, l_{ij}) = \int_{l_{ij}-2}^{l_{ij}+2} \left(\frac{1}{N_{c_i}} \sum_{x_s : C(x_s)=c_i} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{-\frac{(x-x_s)^2}{2t_i^2}} \right) \right) dx \quad \text{for } j = 1 \dots u \tag{14}$$

where x_s are all individual samples belonging to the class c_i in the training dataset.

Similar to Equation (11), if any probability $p(t,l)$ is included in Equation (14), it is included only once and counted in the probability calculation of the packet with the highest weight. In addition, to compute the probability of a packet, we calculate its integral from its 1 byte shorter length packet to its 1 byte longer length packet based on its probability density equation, i.e., $\int_{l-1}^{l+1}$ for a packet with length l. Equation (14) calculates the total probability of three continuous packets $\{l_{i1} - 1, l_{ij}, l_{ij} + 1\}$, i.e., $\int_{l_{ij}-2}^{l_{ij}} + \int_{l_{ij}-1}^{l_{ij}+1} + \int_{l_{ij}}^{l_{ij}+2}$ without overlay calculation.

For each class c_i, a total probability equation related to kernel bandwidth t_i is constructed based on all probabilities of the selected packets as shown in Equation (15):

$$P(t_i) = \sum_{j=1}^{u} p(t_i, l_{ij}) \tag{15}$$

To estimate the kernel bandwidth t_i for the class c_i, an absolute estimation error[7] ε is set up and a maximum kernel bandwidth t_i is selected to make t_i satisfy the following equation

$$|P(t_i) - W_{total}| \leq \varepsilon \tag{16}$$

Figure 5 depicts an example of the proposed Gaussian kernel bandwidth estimation with Equation (16) applied to the JabberChat data packets and its supported protocol traffic types. For instance, when the absolute estimation error ε is set to 1 x 10^{-3} (i.e., the red horizontal line in the middle of Figure 5), the Gaussian kernel bandwidth related to traffic TLS, XMPP, TCP, DNS,[8] JabberChat are 0.398, 0.398, 0.608, 0.646, 1.017 respectively. Figure 5 shows the relationship between kernel bandwidth and estimation error as well.

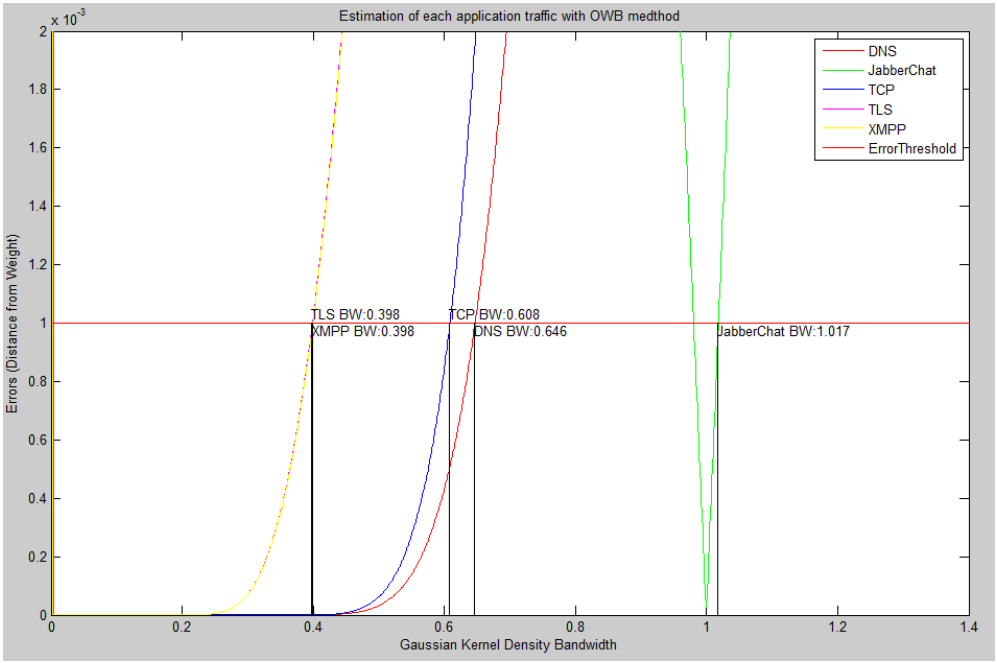
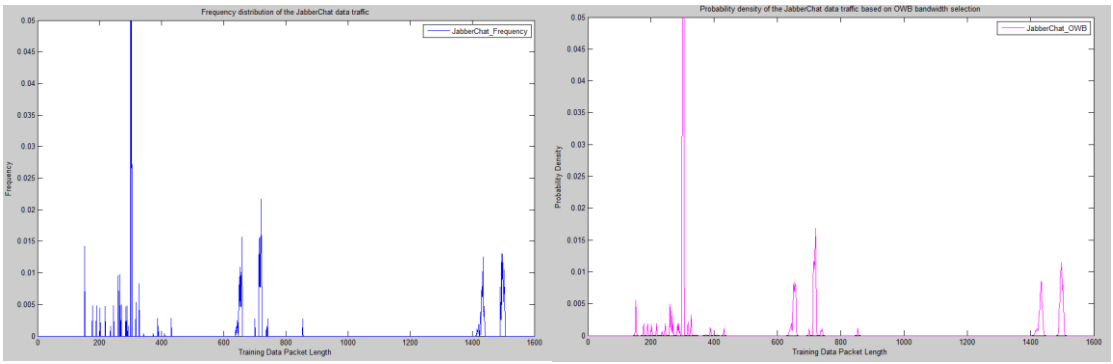


Figure 5: *The results of Gaussian kernel bandwidth estimation based on the proposed OWB method and applied to the JabberChat and its supported protocol traffic.*

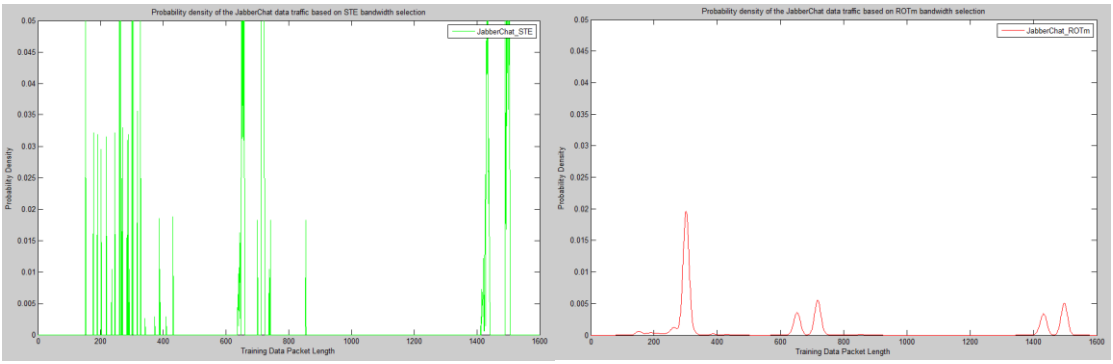
[7] In this work, we choose three decimal places for ε, i.e., ε = 0.001, when considering ROTm in Equation (8) using four decimal places in previous research literatures. We found that it requires significant computational effort when using higher precision. Three decimal places are good enough for the kernel bandwidth estimation based on packet length data.

[8] TLS is Transport Layer Security Protocol; XMPP is Extensible Messaging and Presence Protocol; TCP is Transmission Control Protocol; DNS is Domain Name System.

Figure 6 depicts (a) the relative frequency distribution of the JabberChat training data compared with (b) the Gaussian kernel probability density based on the proposed OWB kernel bandwidth selection, (c) the Gaussian kernel probability density based on the solve-the-equation kernel bandwidth selection, and (d) the Gaussian kernel probability density based on the median absolute deviation ROT kernel bandwidth selection.



(a) Relative frequency distribution of the JabberChat traffic. (b) Probability density of the traffic with OWB.



(c) Probability density of the traffic with STE. (d) Probability density of the traffic with ROTm.

Figure 6: *Comparison of (a) the relative frequency distribution of the JabberChat traffic with (b) the Gaussian kernel probability density based on the OWB kernel bandwidth selection, (c) the probability density based on the STE bandwidth selection, and (d) the probability density based on the ROTm bandwidth selection.*

Figure 6 shows that the Gaussian kernel probability density pattern constructed with the proposed OWB kernel bandwidth selection method is much closer to the relative frequency distribution of the training data compared with the other kernel probability density patterns constructed based on the traditional kernel bandwidth selection algorithms such as ROTm and STE.

3.3 Learning model construction with probability

As we mentioned in Section 2.3.1, some machine learning tools use the probability density (rather than probability) for learning model construction, which may have a negative impact on prediction accuracy. In this work, we prefer using probability of packets to construct the learning model rather than using probability density since the constructed learning model with probability is consistent with Bayes Theorem used in the naïve Bayes classifier and Equation (4). In addition, with traffic packet length data, it is easy for us to calculate the probability of a packet by computing its integral applied to the Gaussian

kernel probability density equation since packet lengths are discrete data and we can easily choose the lower and upper limits (i.e., 1 byte difference in nature) for calculating their integral rather than continuous data such as timing.

Based on the above reasons, the following conditional probability for each testing packet is proposed for constructing the learning model with packet length data for our traffic classification

$$f(y|c_i) = \frac{1}{2}\int_{y-1}^{y+1}\left(\frac{1}{N_{c_i}}\sum_{x_s:C(x_s)=c_i}\left(\frac{1}{\sqrt{2\pi t_i^2}}e^{-\frac{(x-x_s)^2}{2t_i^2}}\right)\right)dx \qquad \text{for } i = 1, 2, \ldots, k \tag{17}$$

where x_s are all individual samples belonging to the class c_i in the training dataset. The probability of a testing packet y given the class c_i is calculated by the integral of the Gaussian kernel density equation with the proposed OWB kernel bandwidth t_i. The lower and upper limits of the integral is set as 1 byte shorter length (*y-1*) and 1 byte longer length (*y+1*) respectively. The probability is normalized by the total integral of all packets,[9] i.e., $\sum_{for\ all\ y}(\int_{y-1}^{y+1}) = 2$, such that the total probability of all packets equals to 1, i.e., $\sum_{for\ all\ y} f(y|c_i) = 1$.

The conditional probability $f(y|c_i)$ of each testing packet y calculated by Equation (17) is applied into the naïve Bayes classifier, i.e., Equation (4). The posterior probability $P(c_i|y)$ of each testing packet y related to each class c_i can be calculated and, further, a learning model can be constructed.

3.4 Traffic classification and unknown traffic prediction

For traffic classification based on packet length data, we modified the traditional naïve Bayes kernel classifier by introducing "unknown" traffic detection. The modified naïve Bayes kernel classifier (MNBK) consists of five steps: OWB kernel bandwidth estimation, posterior probability calculation, traffic class prediction, and "unknown" traffic detection.

MNBK Step 1: OWB Kernel Bandwidth Estimation

The kernel bandwidth t_i related to class c_i is calculated using the optimal weight-based kernel bandwidth selection algorithm described in Section 3.2.

MNBK Step 2: Posterior Probability Calculation

For each testing packet y, its conditional probability $f(y|c_i)$ is calculated based on Equation (17) and the OWB kernel bandwidth t_i from MNBK Step 1. Then, its posterior probability $P(c_i|y)$ related to each class c_i is calculated based on the naïve Bayes classifier, i.e., Equation (4).

[9] The integral is calculated by $\int_{y-1}^{y+1}$ for packet y having half overlay with the integral $\int_{y-1}^{y}$ of packet *y-1*, and another half overlay with the integral $\int_{y}^{y+2}$ of packet *y+1*. Therefore, the total integral for all packet y is $\sum_{for\ all\ y}(\int_{y-1}^{y+1}) = 2$.

Note that in Equation (17), the e^z value will be set to zero by the computer when the exponent z is too small since each computer (both hardware and software) has its limitations when processing digits. On our computer, the e^z value is set to zero when $e^z < 4.9407*10^{-324}$. Therefore, the conditional probability value $f(y|c_i)$ could be computed as zero and the denominator in Equation (4) could be computed as zero as well. Under either situation, we set $P(c_i|y)$ to zero. We could set a higher threshold value for e^z (e.g., 10^{-100}) to decide whether $f(y|c_i)$ should be zero or not. However, we do not discuss the zero threshold setting in this report and leave it for possible future research.

MNBK Step 3: Traffic Classification

For a testing packet with length y, choose the maximum posterior probability from all posterior probabilities related to each class ($P(c_i|y)$ for $i = 1,2,\ldots,k$) with the following equation

$$P_{max}(y) = \max\{P(c_i|y): i = 1,2,\ldots,k\} \tag{18}$$

Predict the classification of the testing packet y based on the following traffic classifier

$$C(y) = \begin{cases} unknown_{class} & if \quad P_{max}(y) = 0 \\ c_i & otherwise, if\ P(c_i|y) = P_{max}(y) \end{cases} \tag{19}$$

Equation (19) means that the packet y belongs to "unknown" class if $P_{max}(y) = 0$. Otherwise, it belongs to the class c_i if $P(c_i|y) = P_{max}(y)$.

MNBK Step 4: "unknown" Traffic Detection

This step is to detect "unknown" traffic that wrongly predicted to class c_i in MNBK Step 3 and to reclassify them to the "unknown" class. Figure 7 depicts the differences between the traditional classifiers and the proposed MNBK classifier on traffic classification and "unknown" traffic detection.

In Figure 7, the traditional supervised classifiers have no difference with work reported in the existing literature, i.e., using labelled training data as input and a machine learning algorithm to build the learning model. Each testing packet is assigned/predicted to a traffic class based on the learning model.

The MNBK classifier adds an extra procedure to the traditional supervised classifiers at the end, i.e., "unknown" traffic detection. The procedure for the "unknown" traffic detection is to investigate the predicted packets from MNBK Step 3, identify exceptional packets that are wrongly predicted to an existing traffic class, and reclassify them as "unknown" traffic class. The "unknown" traffic detection is based on comparing the difference of the testing packet's kernel densities under the predicted results and the training data respectively. Other unidentified testing packets stay with the original predicated traffic class.

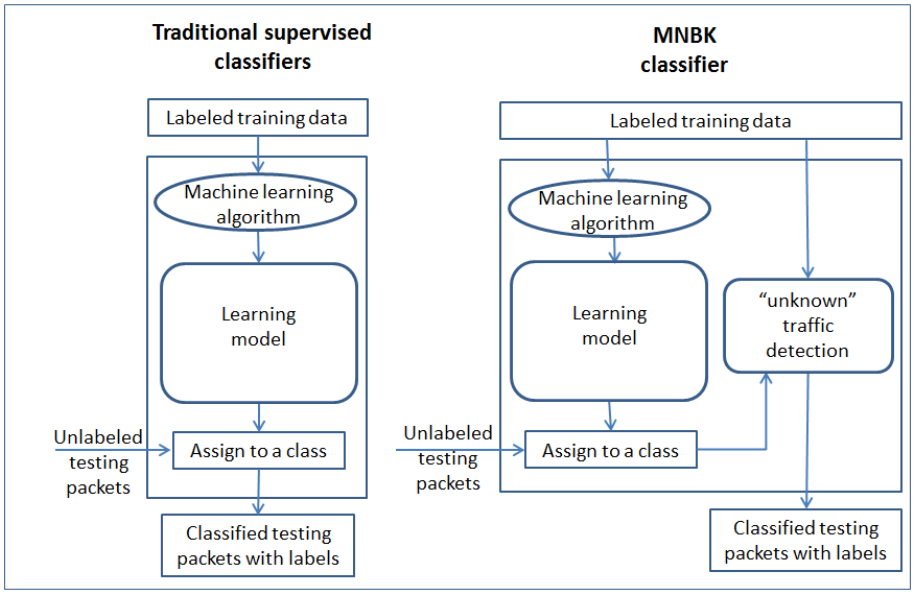


Figure 7: *The different procedures between the traditional supervised classifiers and the proposed MNBK classifier on traffic classification.*

Assume $X'_i = \{x'_{i1}, x'_{i2}, \dots, x'_{iN'_{c_i}}\}$ are the packets predicted to class c_i in the testing dataset and $L'_i = \{l'_{i1}, l'_{i2}, \dots, l'_{im'}\}$ are all different packet lengths in X'_i with their weights $w(l'_{ij})$ arranged in descending order.

Calculate the kernel density of each packet with length l'_{ij} related to class c_i based on the testing data using the following equation

$$f'_{te}(l'_{ij}|c_i) = \frac{1}{N'_{c_i}} \sum_{z=1}^{N'_{c_i}} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{-\frac{(l'_{ij} - x'_{iz})^2}{2t_i^2}} \right) \quad \text{for } j = 1,2, \dots m' \tag{20}$$

where t_i is the OWB kernel bandwidth selected based on the training data in Section 3.2.

Equation (20) gives us the kernel densities $\{f'_{te}(l'_{i1}|c_i), f'_{te}(l'_{i2}|c_i), \dots, f'_{te}(l'_{im'}|c_i)\}$ related to each packet length in $\{l'_{i1}, l'_{i2}, \dots, l'_{im'}\}$ for the testing dataset.

As we know that the packet with length l_{il} has the highest weight for class c_i in the training dataset, we want to use this packet as a baseline to find the major exceptional packets[10] in the predicted class c_i. Therefore, we need to calculate the kernel density of the packet with length l_{il} in the testing dataset using the following equation

[10] Here major exceptional packets are the packets with high relative frequency and wrongly predicted to the class c_i.

$$f_{te}(l_{i1}|c_i) = \frac{1}{N'_{c_i}} \sum_{z=1}^{N'_{c_i}} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{-\frac{(l_{i1}-x'_{iz})^2}{2t_i^2}} \right) \tag{21}$$

There are many ways to check the major exceptional packets in the predicted class c_i. Here we use the ratio and threshold metrics. With the packet l_{i1} as baseline, the kernel density ratio is calculated for each packet l'_{ij} in X'_i based on the following equation.

$$r'_{ij} = \frac{f'_{te}(l'_{ij}|c_i)}{f_{te}(l_{i1}|c_i)} \qquad \text{for } j = 1, 2, \dots, m' \tag{22}$$

A threshold r_{th} can be set to filter the packets we want to investigate. In this work, we set $r_{th} = 0.5$ for two reasons. The packets whose kernel density ratio is over 1 are definitely required for further investigation since their densities in class c_i are over the density of the peak packet which is identified in the training data. Based on the traffic pattern of the class c_i, these packets might be wrongly predicted to the class c_i. In addition, we may miss some major exceptional packets with the ratio just less than 1 if we set the threshold to 1. With the threshold set to 0.5, we only miss some minor exceptional packets.[11] If the threshold r_{th} is set too low, it may increase false negatives and cause extra computational effort. Further research is required to determine the best value of the threshold r_{th}, but we defer that to possible future work. In this work we did not find any exceptional packets whose ratio r'_{ij} is less than 1 under our scenarios, i.e., the traffic generated and collected based on BreakingPoint traffic generator (see details in Section 4).

Assume $\{l'_{i1}, l'_{i2}, \dots, l'_{iv}\}$ are selected packet lengths for further investigation. To estimate whether these selected packets are exceptional to class c_i or not, we compare their density ratios in the training dataset. The density ratio r_{ij} of each selected packet l'_{ij} based on the training data can be calculated with the following equation

$$r_{ij} = \frac{f'_{tr}(l'_{ij}|c_i)}{f_{tr}(l_{i1}|c_i)} \qquad \text{for } j = 1, 2, \dots, v \tag{23}$$

where v ≤ m'

$$f'_{tr}(l'_{ij}|c_i) = \frac{1}{N_{c_i}} \sum_{z=1}^{N_{c_i}} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{-\frac{(l'_{ij}-x_{iz})^2}{2t_i^2}} \right) \quad \text{for } j = 1, 2, \dots, v \tag{24}$$

and

[11] Minor exceptional packets are the packets with low relative frequency and wrongly predicted to the class c_i.

$$f_{tr}(l_{i1}|c_i) = \frac{1}{N_{c_i}} \sum_{z=1}^{N_{c_i}} \left(\frac{1}{\sqrt{2\pi t_i^2}} e^{-\frac{(l_{i1}-x_{iz})^2}{2t_i^2}} \right) \tag{25}$$

For the selected packets $\{l'_{i1}, l'_{i2}, \dots, l'_{iv}\}$, two different density ratios $\{r'_{i1}, r'_{i2}, \dots, r'_{iv}\}$ and $\{r_{i1}, r_{i2}, \dots, r_{iv}\}$ are calculated based on the predicted class in the testing dataset and real class in the training dataset respectively. We reclassify the packets with length l'_{ij} to "unknown" class if their density ratio r'_{ij} in the predicted class is much bigger than the density ratio r_{ij} in the class from the training data. Therefore, we need another threshold R_{th} for reclassification. The reclassification for the packets with length in $\{l'_{i1}, l'_{i2}, \dots, l'_{iv}\}$ is based on the following equation

$$C(l'_{ij}) = \begin{cases} Unknown_{class} & if \ R_{ij} \geq R_{th} \\ c_i & otherwise \end{cases} \quad \text{for } j = 1, 2, \dots, v \tag{26}$$

where

$$R_{ij} = \frac{r'_{ij}}{r_{ij}} \quad \text{for } j = 1, 2, \dots, v \tag{27}$$

For the reclassification threshold R_{th}, if it is too small, it may cause more false negatives since it can wrongly reclassify the correct class c_i packets to "unknown" class. If it is too big, it may still reduce a certain number of false positives but it may miss many "unknown" class packets and retain them in the class c_i. Therefore, finding the best threshold for R_{th} is another challenge and needs further research. In this work, we set R_{th} to 40. Based on our testing, we found that there is not much difference when setting R_{th} between 10 and 40 under our scenarios, i.e., using the traffic generated and collected based on BreakingPoint traffic generator in Section 4.

4 Performance of modified naïve Bayes classifier

In this section, we set up a testbed to generate and collect several encrypted traffic types transmitted over wireless channels in tactical edge networks, and to demonstrate the performance of our modified naïve Bayes kernel classifier.

4.1 Encrypted traffic data collection

To evaluate the performance of our proposed modified naïve Bayes kernel classifier, we set up a testbed as shown in Figure 8.

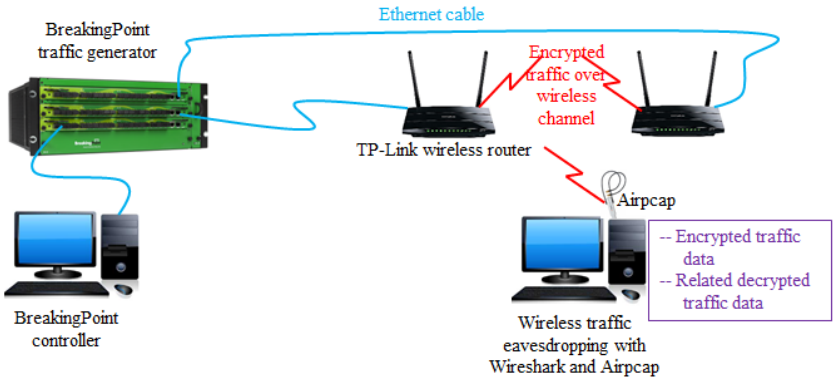


Figure 8: *A testbed to generate and collect encrypted traffic through wireless channel.*

The testbed uses the BreakingPoint [20] traffic generator to generate different traffic types consisting of single application traffic such as voice call, or mixed application traffic such as voice, video, chat, etc. The BreakingPoint controller is used to configure the traffic generation. The generated traffic will be transmitted over the wireless channel between two wireless routers, which apply encryption to the on-air data. A wireless sniffer is set up in a standalone computer with Airpcap and Wireshark for wireless traffic eavesdropping. The traffic collected in the eavesdropping computer is the encrypted traffic data but it can be decrypted with the encryption key for performance evaluation.

In this work, four traffic applications are considered in our evaluation. They are JabberChat, voice call, video call, and file transfer, which represent similar major traffic applications such as instant messaging and file sharing in tactical edge networks. To complete communications for each application with the BreakingPoint traffic generator, other supported protocols may be involved during communication as well. For example, for JabberChat traffic, other protocols such as DNS, TCP, TLS, and XMPP are used as well to support its communication. The BreakingPoint traffic generator can deliver a dynamic and realistic pattern for each application. This is the main reason we use it in our work.

4.2 Training data labelling and testing data prediction

In the training phase, each training traffic packet is collected with its encrypted packet length and is labelled with a traffic class based on its decrypted packet. Therefore, all training data packets are labelled with correct traffic class and are recorded with encrypted packet length.

In the testing phase, each testing traffic packet is collected with its encrypted packet length as well for traffic classification. There is no need to label the testing traffic packets for the traffic classification. However, to evaluate the performance of the traffic classification such as predictive accuracy, we need to know its correct traffic class for each encrypted testing packet. This correct traffic class information for each encrypted testing packet can be retrieved from its decrypted packet.

4.3 Traffic classification performance metrics

The most important metric to differentiate (or evaluate) different traffic classification techniques is the predictive accuracy value, i.e., the percentage of the correct decisions made by the classification techniques on testing traffic packets. There are many metrics for evaluating a traffic classifier's accuracy from different angles, for instance, precision, recall, true positive, false positive, true negative, false negative, etc. In this work, we use three popular metrics—accuracy, precision, and recall for our performance evaluation. In addition, we extend the traditional binary-class classification metrics (see Table 1) to multiple-class classification metrics (see Table 2) for metric calculation.

Table 1: *The traditional binary-class classification metrics for accuracy, precision, and recall.*

		True Condition		
		Positive	**Negative**	
Predicted Condition	**Positive**	*True Positive (tp)*	*False Positive (fp)*	$\boldsymbol{Precision=\frac{tp}{tp+fp}}$
	Negative	*False Negative (fn)*	*True Negative (tn)*	
		$\boldsymbol{Recall=\frac{tp}{tp+fn}}$		$\boldsymbol{Accuracy=\frac{tp+tn}{tp+tn+fp+fn}}$

For multiple classes, the true positive tp_{c_i} for class c_i could be the true negative tn_{c_j} for class c_j. To simplify the table, we use only the true positive for each class. Similarly, the false positive $fp_{c_{ji}}$ for the predicted class c_i under the true class c_j, could be the false negative $fn_{c_{ij}}$ for the predicted class c_j under the true class c_i. To simplify the table, we use only the false positive for all classes. The traffic classification metrics for multiple-class classification are described below.

Table 2: *The multiple-class classification metrics for accuracy, precision, and recall calculation.*

		True Condition Classes				
		c_1	c_2	…	c_k	
Predicted Condition Classes	c_1	tp_{c_1} $(tn_{c_{j:j\neq 1}})$	$fp_{c_{21}}$ $(fn_{c_{12}})$		$fp_{c_{k1}}$ $(fn_{c_{1k}})$	$\textbf{Precision}=\frac{tp_{c_1}}{tp_{c_1}+\sum_i fp_{c_{i1}}}$
	c_2	$fp_{c_{12}}$ $(fn_{c_{21}})$	tp_{c_2} $(tn_{c_{j:j\neq 2}})$		$fp_{c_{k2}}$ $(fn_{c_{2k}})$	$\textbf{Precision}=\frac{tp_{c_2}}{tp_{c_2}+\sum_i fp_{c_{i2}}}$
	…					
	c_k	$fp_{c_{1k}}$ $(fn_{c_{k1}})$	$fp_{c_{2k}}$ $(fn_{c_{k2}})$		tp_{c_k} $(tn_{c_{j:j\neq k}})$	$\textbf{Precision}=\frac{tp_{c_k}}{tp_{c_k}+\sum_i fp_{c_{ik}}}$
		$\textbf{Recall}=\frac{tp_{c_1}}{tp_{c_1}+\sum_i fp_{c_{1i}}}$	$\textbf{Recall}=\frac{tp_{c_2}}{tp_{c_2}+\sum_i fp_{c_{2i}}}$		$\textbf{Recall}=\frac{tp_{c_k}}{tp_{c_k}+\sum_i fp_{c_{ki}}}$	$\textbf{Accuracy}=\frac{\sum_{i=1}^{k} tp_{c_i}}{\sum_{i=1}^{k} tp_{c_i}+\sum_{ij} fp_{c_{ij}}}$

- **Accuracy:** The percentage of all true positive traffic packets predicted to each class (i.e., all correctly classified packets) among the total number of traffic packets examined.
- **Precision for Class c_i:** The percentage of the true positive traffic packets predicted to class c_i (i.e., correctly classified packets to class c_i) among the total number of traffic packets predicted to class c_i.
- **Recall for Class c_i:** The percentage of the traffic packets correctly classified to class c_i among the total number of packets belonging to class c_i.

4.4 Performance of modified naïve Bayes kernel classifier

In this section, we generate and collect single application traffic as training data[12] and multiple application traffic as testing data[13] to evaluate and compare the classification performance of the proposed modified naïve Bayes kernel classifier (MNBK) with the traditional naïve Bayes kernel classifiers. The traditional naïve Bayes kernel classifiers use different kernel bandwidth selections such as ROTs, ROTm, and STE. These classifiers are named as NBK-ROTs, NBK-ROTm, and NBK-STE classifier respectively.

[12] Note that in this report, single application traffic contains the application's data packets and its related supporting protocol packets, i.e., the single application traffic contains multiple classes as well but with one application class only. For example, the single JabberChat application traffic contains the JabberChat data packets and its related supporting protocol packets such as DNS, TCP, TLS, and XMPP. So, the traffic contains five classes: JabberChat, DNS, TCP, TLS, and XMPP.

[13] Note that here multiple application traffic contains the multiple application data packets and their related supporting protocol packets.

To evaluate the performance of the four classifiers, the single JabberChat application traffic is generated in the training phase. The encrypted JabberChat packets and related supporting protocol traffic are collected and labelled as training data. In the testing phase, the mixed applications such as JabberChat, voice, video, and file transfer are generated. Their encrypted packets and related supporting protocol traffic are collected as testing data for traffic classification. The performance of the traffic classification is based on the comparison of the predicted results and the correct traffic class. As the training data only contains the JabberChat data and related supporting protocol traffic packets, the classifiers could predict other application data such as voice, video, and file transfer and their related supporting protocols which are not labelled in the training dataset as "unknown" traffic.

Figure 9 depicts the classification accuracy of the four different classifiers. The classification accuracy is calculated for all packets correctly classified to JabberChat, its related supporting protocols, i.e., DNS, TCP, TLS, and XMPP, and "unknown" traffic, i.e., voice, video, and file transfer packets.

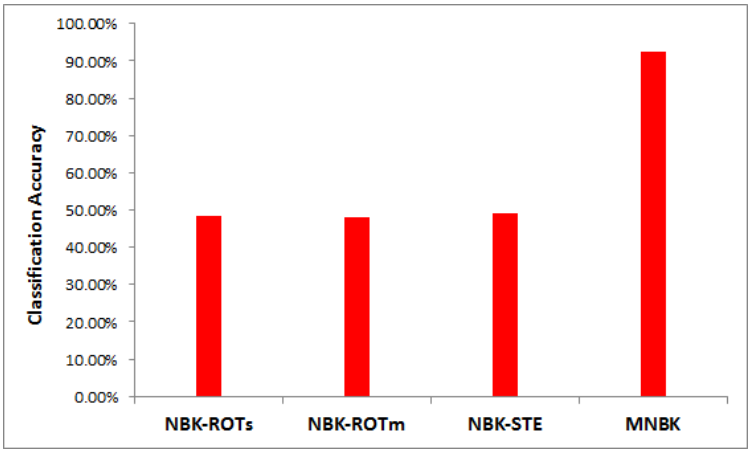


Figure 9: *The classification accuracy of the proposed MNBK classifier compared with other traditional NBK classifiers.*

The figure shows that the proposed MNBK classifier increases the classification accuracy by over 40% compared with the traditional NBK kernel classifiers. Even using the single attribute, i.e., packet length, the MNBK classifier can reach 93% accuracy but the traditional NBK classifiers can only reach less than 50% accuracy. This result shows that the proposed MNBK classifier can correctly detect most "unknown" traffic. The traditional NBK classifiers predict most of the "unknown" packets to the classes contained in the training dataset, e.g., the NBK-ROTs and NBK-ROTm classifiers predict most "unknown" traffic to JabberChat and the NBK-STE classifier predicts most "unknown" traffic to XMPP.

Detailed information about these classifications is shown in Figure 10.

Traffic classification with MNBK								
Accuracy	93%							
	True DNS	True Jabber	True TCP	True TLS	True XMPP	True UnKnown	total	Precision
Predicted DNS	223						223	100%
Predicted Jabber	18	10632		61	2682	440	13833	77%
Predicted TCP			18386			853	19239	96%
Predicted TLS		62		66			128	52%
Predicted XMPP					150	0	150	100%
Predicted UnKnown		27		24	340	26544	26935	99%
Total	241	10721	18386	151	3172	27837		
Recall	93%	99%	100%	44%	5%	95%		

(a) Traffic classification with MNBK.

Traffic classification with NBK-ROTs								
Accuracy	48%							
	True DNS	True Jabber	True TCP	True TLS	True XMPP	True UnKnown	total	Precision
Predicted DNS	148						148	100%
Predicted Jabber	93	10704		113	3005	26984	40899	26%
Predicted TCP			18386			853	19239	96%
Predicted TLS		17		38	167		222	17%
Predicted XMPP					0		0	0%
Predicted UnKnown						0	0	0%
Total	241	10721	18386	151	3172	27837		
Recall	61%	100%	100%	25%	0%	0%		

(b) Traffic classification with NBK-ROTs.

Traffic classification with NBK-ROTm								
Accuracy	48%							
	True DNS	True Jabber	True TCP	True TLS	True XMPP	True UnKnown	total	Precision
Predicted DNS	111					178	289	38%
Predicted Jabber	130	10454		113	2702	22174	35573	29%
Predicted TCP			18386			675	19061	96%
Predicted TLS		228		38	340	4576	5182	1%
Predicted XMPP		39			130	234	403	32%
UnKnown						0	0	0%
Total	241	10721	18386	151	3172	27837		
Recall	46%	98%	100%	25%	4%	0%		

(c) Traffic classification with NBK-ROTm.

Traffic classification with NBK-STE								
Accuracy	49%							
	True DNS	True Jabber	True TCP	True TLS	True XMPP	True UnKnown	total	Precision
Predicted DNS	74						74	100%
Predicted Jabber		5928		13	977	3553	10471	57%
Predicted TCP			18386			675	19061	96%
Predicted TLS		52		0			52	0%
Predicted XMPP	167	54		62	767	19019	20069	4%
Predicted UnKnown		4687		76	1428	4590	10781	43%
Total	241	10721	18386	151	3172	27837		
Recall	31%	55%	100%	0%	24%	16%		

(d) Traffic classification with NBK-STE.

Figure 10: *The traffic classification detail information of the proposed MNBK classifier compared with other traditional NBK classifiers.*

In the figure, the MNBK classifier detects "unknown" traffic with much better recall and precision compared with the other three classifiers. For recall on "unknown" traffic detection, the MNBK classifier can correctly detect 95% of the "unknown" traffic, the NBK-STE classifier only detects 16% of the "unknown" traffic, both the NBK-ROTs and NBK-ROTm classifiers do not detect any "unknown" traffic. For precision on "unknown" traffic prediction, the MNBK classifier makes 1% mistakes on predicted "unknown" traffic, the NBK-STE classifier makes 57% mistakes on predicted "unknown" traffic. Both the ROTs and ROTm classifiers wrongly predict all "unknown" traffic to others.

Figure 11 depicts the classification precisions of the four classifiers on testing data, where DNS, TCP, TLS, and XMPP are the supporting protocols for the JabberChat communications.

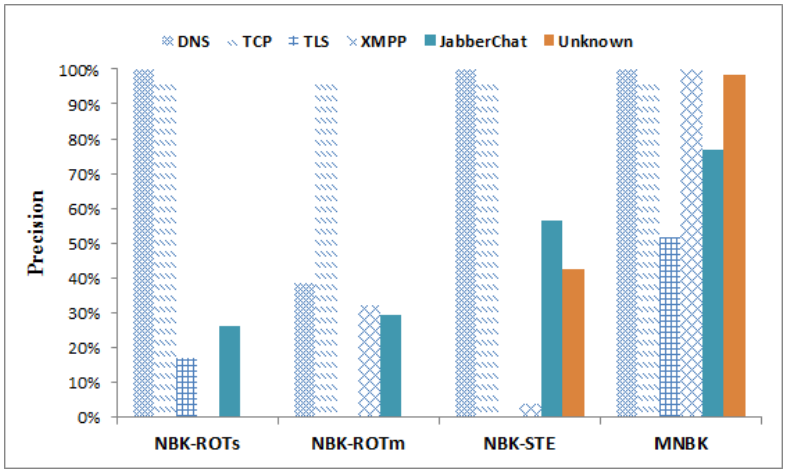


Figure 11: *The classification precisions of the four classifiers on the mixed multiple application traffic as testing data.*

Figure 11 shows that the traditional NBK classifiers with ROTs and ROTm kernel bandwidths cannot detect any "unknown" traffic. Both methods have very low precisions on the JabberChat data traffic (less than 30%), i.e., they cause very high false positives on the JabberChat packet prediction. The traditional NBK classifier with the STE kernel bandwidth has better precision compared with NBK-ROTs and NBK-ROTm. It detects certain "unknown" traffic, though with low precision (around 43%), and increases the prediction precision to 57% on the JabberChat data traffic. The proposed MNBK classifier is the best among the four classifiers. It increases the prediction precision to 77% on the JabberChat data traffic. For the "unknown" traffic, it reaches 99% precision, i.e., very low false positive on "unknown" traffic detection. The reason for this result is that the traditional NBK classifiers wrongly predict many "unknown" and XMPP packets as JabberChat packets (see Figure 10), resulting in very low precision on JabberChat packet prediction. The proposed MNBK classifier can effectively differentiate them, especially the JabberChat and "unknown" packets, and classify them to the correct classes.

For the related supporting protocol traffic prediction, all traditional NBK classifiers have very low precision on TLS and XMPP traffic, i.e., resulting in high false positive rate on these traffic packets. This low precision result is caused by these three traffic classes (JabberChat, TLS, XMPP) having overlay packets with the same or very close packet lengths. When the packets under different classes overlay each other or are too close, classifiers with only one attribute (e.g., length) may classify them to one class and make mistakes (i.e., false positive). A good classifier with only one attribute can improve the classification precision on closely related packets that do not have an overlay (e.g., MNBK). A bad classifier cannot differentiate these close packets, resulting in classifying them to one class and producing higher false positive rate (e.g., NBK-ROTs, NBK-ROTm, and NBK-STE). In addition, any classifier with only one attribute cannot differentiate the packets with overlay, resulting in predicting them to one class with false positives. This is the reason that the proposed MNBK still makes mistakes on some traffic packet prediction. A better solution is to research multiple attributes classification techniques, which are not discussed in this report.

Figure 12 depicts the classification recall of the four classifiers on testing data. In this figure, MNBK, NBK-ROTs, and NBK-ROTm have very high recall (over 98%) on the JabberChat data traffic, i.e., they have very low false negative rate on the JabberChat packet prediction. However, both NBK-ROTs and NBK-ROTm have zero recall on the "unknown" traffic, i.e., they have 100% false negative rate on the "unknown" traffic prediction. The NBK-STE classifier has around 16% recall on the "unknown" traffic which is better than NBK-ROTs and NBK-ROTm but it reduces the recall on the JabberChat data packet prediction to 55% as a trade-off. The proposed MNBK classifier not only has over 99% recall on the JabberChat traffic data but also increases the recall to over 95% on the "unknown" traffic, i.e., it results in very low false negative rate on both the JabberChat and "unknown" traffic packets. This is because the traditional NBK classifiers predict the majority "unknown" traffic packets to other existing classes contained in the training dataset, resulting in very low recall on "unknown" traffic packet prediction. The proposed MNBK classifier can effectively differentiate "unknown" traffic from the existing classes and correctly detect "unknown" traffic packets.

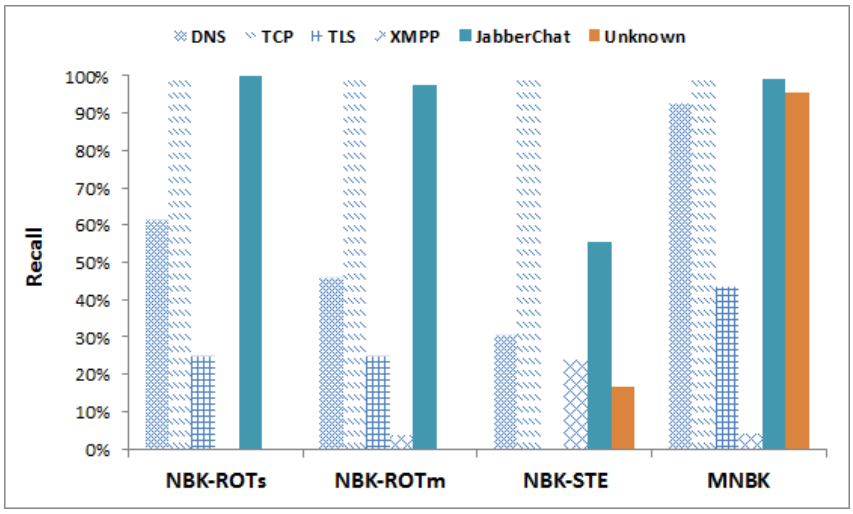
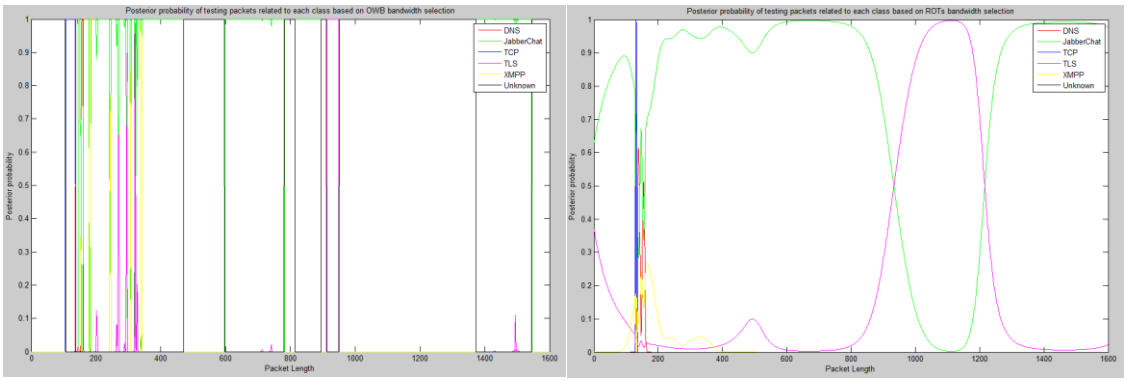


Figure 12: *The classification recall of the four classifiers on the mixed multiple application traffic as testing data.*

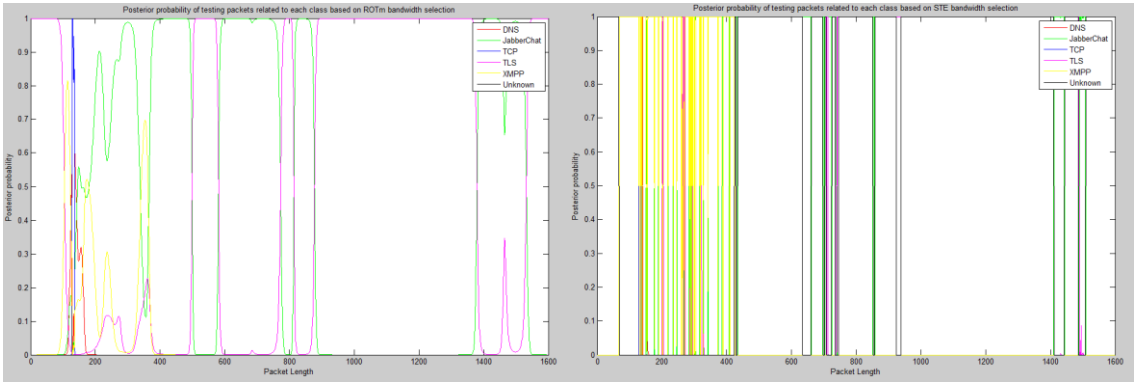
For the related supporting protocol traffic prediction, all four classifiers have very low recall rates on TLS and XMPP traffic, resulting in high false negative rate on these traffic predictions. The main reason for such performance is the same as that explained in the classification precision part, i.e., many TLS and XMPP packets' lengths approximately equal or are close to those of the JabberChat packets, resulting in wrongly predicting to the JabberChat class. Since they are the application supporting protocols and there are not many in the traffic compared with the JabberChat data packets, it does not have much effect on the classification recall of the JabberChat data packet prediction.

There are two main reasons for the MNBK classifier having better classification performance than other when considering all three major evaluation metrics. The first one is that, as discussed in Section 3, the learning model constructed by the MNBK classifier is closer to the real distribution of the training data. The second reason is that the MNBK classifier considers the packet distribution in the testing data as well to inspect exceptional packets and detect "unknown" packets during traffic prediction (see the difference between MNBK and the traditional supervised classifiers in Figure 7).

In addition, the different learning models can make a big difference on the posterior probability calculation for testing traffic packets, which affects the classification accuracy, precision, and recall. Generally speaking, a better learning model can give a more accurate posterior probability for each testing traffic packet during traffic classification. Figure 13 depicts the posterior probabilities for each testing traffic packet predicted under different classes and learning models. Figure 13 shows the background of Figures 9–12 for each packet prediction. We can see that the posterior probability related to "unknown" traffic is zero for each packet length under both NBK-ROTs and ROTm classifiers, insulting in both classifiers missing the "unknown" traffic detection. NBK-STE predicts certain "unknown" traffic but makes more mistakes on the JabberChat traffic prediction as a trade-off. The proposed MNBK classifier produces good classification results on both of the JabberChat and "unknown" traffic packets.



(a) Posterior probability of testing packets with MNBK. (b) Posterior probability of testing packets with NBK-ROTs.



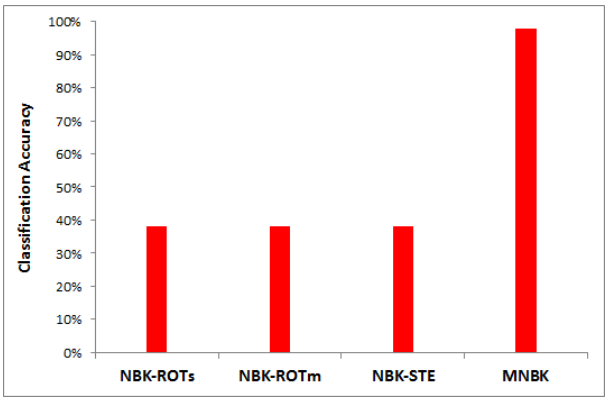
(c) Posterior probability for testing packets with NBK-ROTm. (d) Posterior probability for testing packets with NBK-STE.

Figure 13: *The posterior probability of testing traffic packet under different classes and predicted by the four different classifiers.*

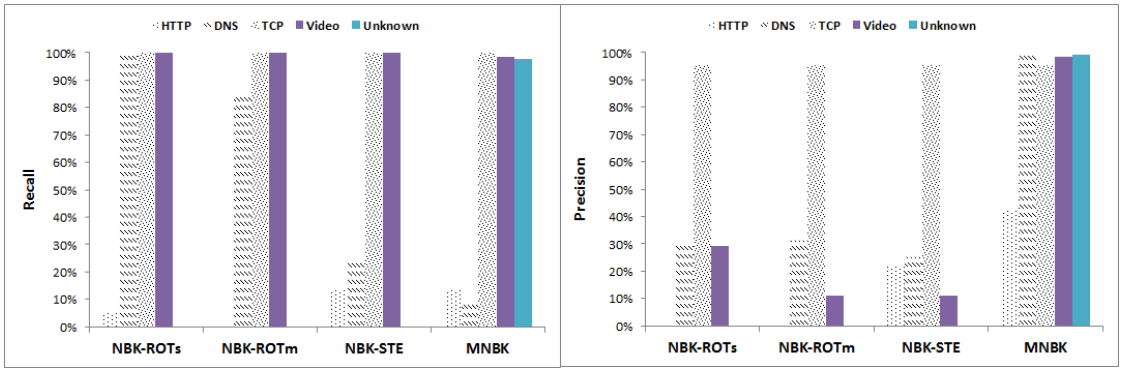
To demonstrate the classification performance of the proposed MNBK technique with different training data, we use single video application traffic as the training data[14] and the same multiple application traffic used in Figure 9–12 as the testing data. We then compare the classification results with other traditional classifiers. Figures 14 (a), (b), and (c) depict the classification accuracy, precision, and recall of the four classifiers.

Figure 14 shows that the proposed MNBK classifier has even better performance on classification accuracy, precision, and recall when using the single video application traffic as the training data compared with the results from Figures 9–12. Under this scenario, all three traditional classifiers miss the "unknown" traffic prediction. In addition, they result in huge false positive rates (i.e., very low precision) on the video traffic prediction as well though their recall is very good. The MNBK classifier increases the classification accuracy by over 50% compared with others (less than 40%). It predicts and classifies both the video and "unknown" traffic packets by over 98% in accuracy, precision, and recall.

[14] Note that here single video application traffic contains the video data packets and its related supporting protocol packets: DNS, TCP, and HTTP.



(a) Accuracy



(b) Precision (c) Recall

Figure 14: *The classification accuracy, precision, and recall of the four classifiers on the same mixed multiple application traffic used in Figure 9–12 as testing data but using the single video application traffic as training data.*

Generally speaking, the proposed MNBK classifier has much better performance in predicting and detecting unknown traffic packets that do not belong to the classes contained in the training data, and it increases the classification accuracy, precision, and recall of the existing classes as well.

5 Real time and continuous traffic classification

In this section, we discuss how to apply the proposed MNBK classifier to real time and continuous traffic classification [4] and compare its classification performance with other traditional classifiers.

5.1 Real time traffic classification

It is straightforward to apply the traditional supervised classifiers such as NBK-ROTs, NBK-ROTm, and NBK-STE to real time traffic classification with the packet length attribute. First, unlike other attributes such as timing, packet lengths are discrete integers and have limited sizes. The posterior probability $P(c_i|y)$ of each testing packet with length y related to each class c_i can be pre-calculated with Equation (4) based on the training data. Therefore, each testing packet with length y can be pre-classified to a class based on Section 2.3 and a predefined classification table[15] can be constructed for all testing packet lengths based on learning model and training data (see Figure 15). For real time traffic classification with these traditional supervised classifiers, once a real time traffic packet is received, it can be assigned to a class immediately based on the packet length and the predefined classification table.

It is a little different for applying the proposed MNBK classifier to real time traffic since MNBK considers the distribution features of the ongoing testing data and detects some exceptional packets as "unknown" traffic. This means the MNBK classifier requires the continuous computation for "unknown" traffic detection/reclassification based on real time traffic. To make the continuous computation more efficient, we can set a reclassification investigation point[16] to limit the number of reclassification computations, where the time interval of two adjacent investigation points should be longer than the reclassification computation time. In addition, based on Section 3.4, Step 5, only attributes based on Equations (20), (21), and (22) are required for recalculation since they use the testing data. Other attributes based on Equations (23), (24), and (25) can be pre-calculated since they are based on the training data. The procedure to use the MNBK classifier in real time traffic is described below.

Assume $\{T_1, T_2, T_3, \ldots\}$ are reclassification investigation points with time interval T_{intvl}. As in the traditional supervised classifiers, MNBK first constructs a predefined classification table for each testing packet based on the training data with MNBK Step 1–4 in Section 3.4. The MNBK classifier assigns each arriving packet to a class in real time based on the predefined classification table as well. The difference is that at each reclassification investigation point, the MNBK classifier recalculates the probability density of each packet based on all received real time traffic (i.e., testing data), reclassifies some exceptional packets, and updates the predefined classification table. The newly arrived real time traffic is classified based on the updated classification table. Figure 15 depicts the differences between the traditional supervised classifiers and the proposed MNBK classifier on real time traffic classification.

In Figure 15, the predefined classification table is created based on learning model and packet length, and added to both the traditional supervised classifiers and the MNBK classifier for real time traffic classification. The predefined classification table can predict a real time received packet to a traffic class

[15] Predefined classification table is a table which matches a packet to a class based on its length. The predefined classification table is created based on the learning model and training data.

[16] Reclassification investigation point is a time point to investigate the already received and predicted real time traffic packets, to find wrongly predicted packets, and to update and correct the predefined classification table used for predicting ongoing future real time packets.

immediately without much delay since there is no more computation required for the prediction. For the real time MNBK classifier, different to Figure 7, it needs to periodically update the predefined classification table at each reclassification investigation point.

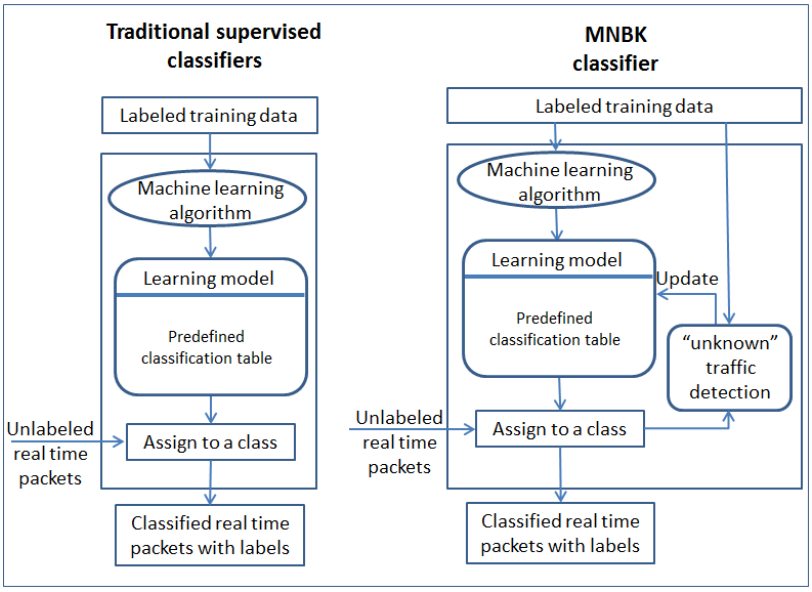


Figure 15: *The different procedures between the traditional supervised classifiers and the proposed MNBK classifier on real time traffic classification.*

The basic rule of this real time MNBK classification model is that the learning model never changes. The reclassification is based on the comparison of the cumulated real time traffic at each reclassification investigation point to the original training dataset. Therefore, the real time traffic does not introduce or build extra error into the learning model. The real time traffic might introduce errors into the predefined classification table and affect the classification performance due to wrong prediction of the exceptional packets. Along with more and more cumulated real time traffic available, i.e., more accurate information retrieved from the traffic, this error will become smaller and smaller.

Figure 16 depicts the average classification accuracy[17] of real time traffic with T_{intvl} = 10s under different classifiers, where the training data is the single JabberChat application traffic (i.e., JabberChat data plus its supporting protocols). The real time traffic is the mixed multiple application traffic (i.e., multiple application data plus their supporting protocols), and the reclassification investigation point for MNBK is set to every ten seconds. Figure 16 shows that without considering the testing data (i.e., ongoing real time traffic) the MNBK classifier is just marginally better than other traditional classifiers, e.g., the average classification accuracy from zero to ten seconds in Figure 16. However, MNBK improves a lot once it updates the predefined classification table based on the received real time traffic at the first reclassification investigation point.

[17] Average classification accuracy is the classification accuracy of all real time traffic received within a time interval. We use this classification accuracy value as the average classification accuracy within that time interval since time is a continuous variable and we cannot calculate classification accuracy at all time points. In addition, we want to investigate the classification performance under different time intervals as well.

In addition, we can see that the average classification accuracy may change under each ten second time interval for each classifier. This is understandable since the traffic is different under each ten second interval. For example, at the last ten second interval in Figure 16, i.e., from 40s to 50s, the classification accuracy of NBK-ROTs, NBK-ROTm, and NBK-STE drop significantly. This drop is caused by the "unknown" traffic occupying the majority traffic at the last ten second interval. The traditional NBK classifiers wrongly predict them to the existing classes contained in the training dataset, which cause lower classification accuracy during this time interval compared with the classification accuracies during other time intervals. In contrast, the classification accuracy of the proposed real time MNBK classifier increases in the last ten second time interval. This result is caused by that there are fewer traffic types, i.e., JabberChat, TCP, and "unknown," in the last ten second time interval, and these traffic packets have fewer overlay or too close packets compared with traffic among JabberChat, TLS, and XMPP contained in the other time intervals. Therefore, the proposed MNBK classifier can differentiate the traffic appearing in the last ten second time interval very well and make fewer mistakes on prediction compared with other time intervals which contain TLS and XMPP traffic.

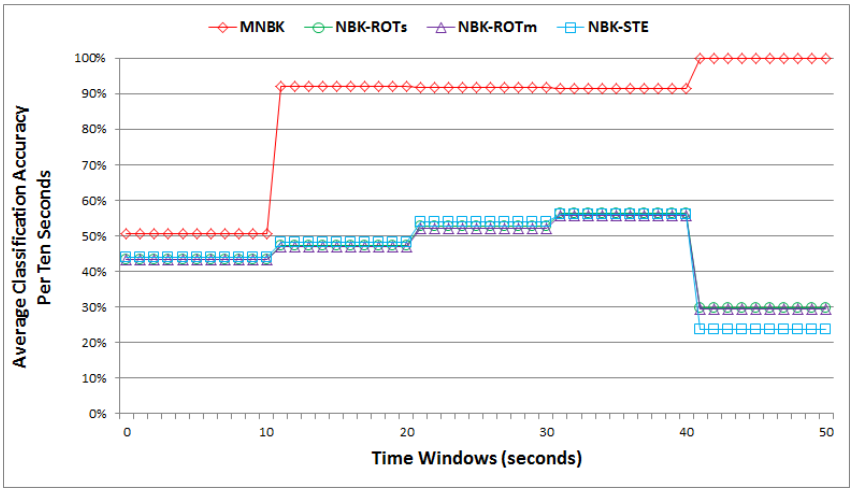


Figure 16: *The average classification accuracy per ten second on the real time mixed multiple application traffic as testing data under different traffic classifiers.*

Figure 17 depicts the average precision of the real time JabberChat data and "unknown" data traffic over each ten second time interval under different classifiers. Figure 17 shows that MNBK has much better precision in predicting "unknown" traffic compared with other traditional classifiers. It does not do a good job on JabberChat traffic in the first ten second time interval, i.e., without considering the testing data. However, it improves a lot once it updates its classification table based on the received real time traffic at the first reclassification investigation point.

The average classification precision changes significantly in the last ten second time interval compared with other time intervals, especially the NBK-STE classifier. As we mentioned above, there are only three traffic types, i.e., JabberChat, TCP, and "unknown," during the last time interval. For the traditional NBK classifiers, the prediction errors occur in JabberChat and "unknown" traffic in the last time interval compared with errors occurring for more traffic types in other time intervals. For the NBK-STE classifier, it correctly predicts the JabberChat traffic but wrongly predicts all "unknown" traffic to the XMPP class, resulting in very high precision on JabberChat packets but very low precision on "unknown" traffic compared with its classification precision in other time intervals. NBK-ROTs and NBK-ROTm classifiers

wrongly predict all "unknown" traffic to JabberChat class in the last time interval, resulting in lower precision on JabberChat compared with the precision in the other time intervals since "unknown" is the majority traffic in the last time interval. The proposed MNBK classifier increases the precision on JabberChat in the last time interval because there are no TLS and XMPP packets in the last time interval; these packets appeared in the previous time intervals and were mainly predicted to JabberChat, resulting in lower precision.

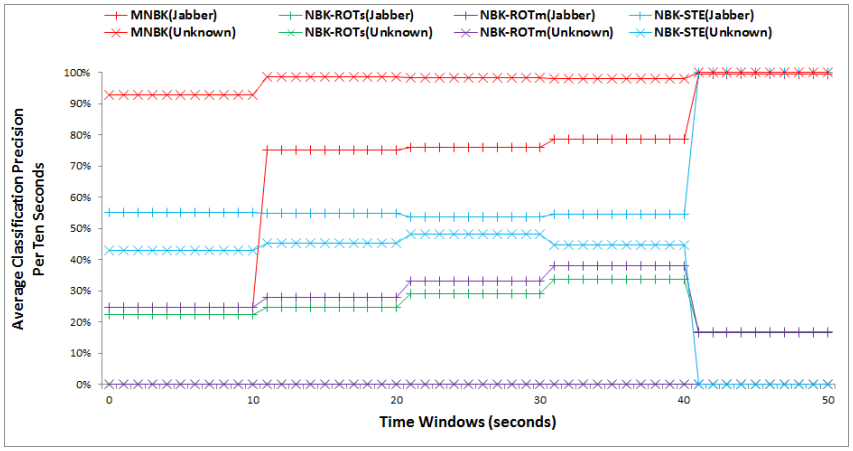


Figure 17: *The average classification precision per ten second on the real time JabberChat data and "unknown" data traffic under different traffic classifiers.*

Figure 18 depicts the average classification recall of the real time JabberChat data and "unknown" data traffic over each ten second time interval under different classifiers. Figure 18 shows that without considering the testing data all classifiers have low recall on "unknown" traffic. However, the MNBK classifier has a huge improvement after updating its classification table at the first reclassification investigation point based on the received real time traffic. The reason of this improvement is that MNBK detects the exceptional packets at the first reclassification investigation point based on the received real time traffic.

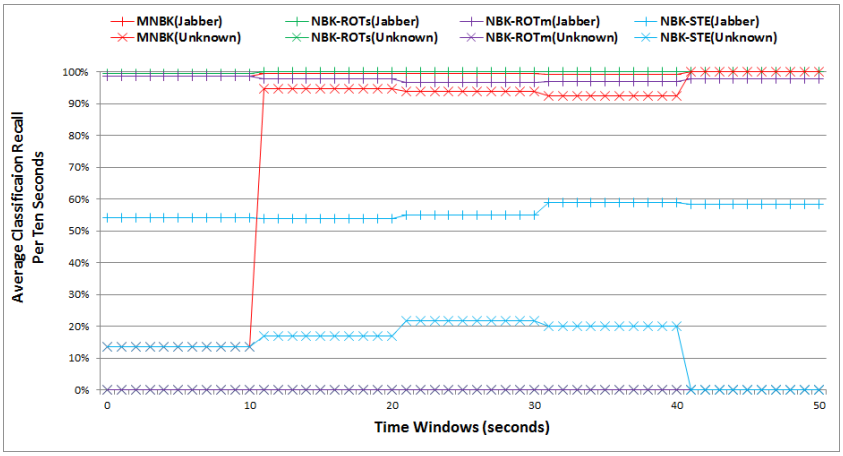


Figure 18: *The average classification recall per ten second on the real time JabberChat data and "unknown" data traffic under different traffic classifiers.*

To demonstrate the classification performance of the different classifiers under different training data, we choose another single application traffic—video and its related supporting protocol traffic—as the training data. Figure 19 depicts the average classification accuracy of the different classifiers on the real time mixed multiple application traffic based on the single video application traffic as the training data. Figure 19 shows that the MNBK classifier has much better accuracy performance than other traditional classifiers even during the first ten second time interval. In addition, the classification accuracy of the traditional NBK classifiers decreases a lot in the last time interval, i.e., from 40s to 50s. This is because there is no video traffic during the last time interval and the traditional NBK classifiers wrongly predict all "unknown" traffic to the existing classes contained in the training dataset, resulting in lower classification accuracy compared with the previous time intervals that contained video traffic. This demonstrates that with less knowledge about the testing traffic, the classification performance of the traditional classifiers decreases a lot. However, this will not have much effect on the classification performance of the proposed MNBK classifier.

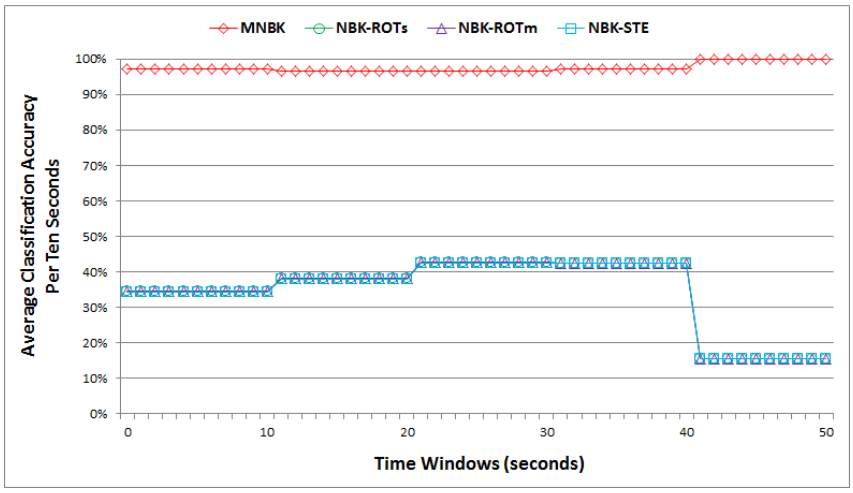


Figure 19: *The average classification accuracy per ten seconds on the same real time mixed multiple application traffic as testing data under different traffic classifiers but using the single video application traffic as training data.*

Figure 20 depicts the average classification precision of the real time video data and "unknown" data traffic over each ten second time interval under different classifiers. Figure 20 shows that MNBK has much better precision performance on predicting both the video and "unknown" traffic compared with other traditional classifiers. In addition, the classification precision of the traditional NBK classifiers on video traffic reduces to zero at the last time interval. This classification precision drop is caused by that the traditional NBK classifiers predict certain (part or all, depending on the classifier) "unknown" traffic to video traffic but there is no video traffic in the last time interval.

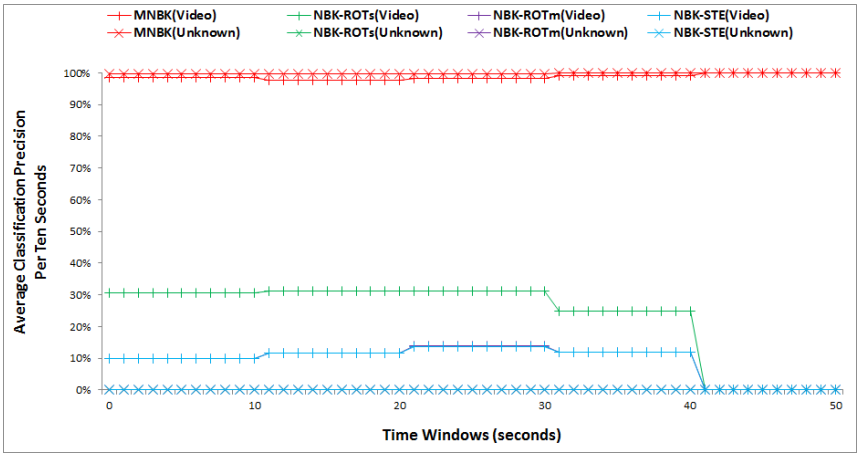


Figure 20: *The average classification precision per ten seconds on the real time video data and "unknown" data traffic under different traffic classifiers.*

Figure 21 depicts the average classification recall performance of the real time video data and "unknown" data traffic over each ten second time interval under different classifiers. Figure 21 shows that the MNBK classifier has very good recall performance on both the video and "unknown" traffic. Other traditional classifiers have very poor recall performance on "unknown" traffic even though they have pretty good recall on the video traffic. This poor performance on recall is caused by that the traditional NBK classifiers predict all "unknown" traffic to other existing classes contained in the training dataset resulting in 0% recall on the "unknown" traffic.

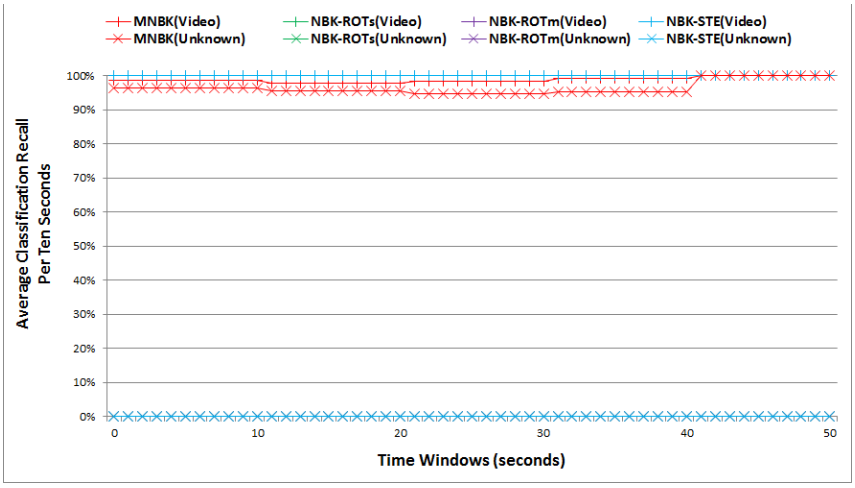


Figure 21: *The average classification recall per ten seconds on the real video data and "unknown" data traffic under different traffic classifiers.*

Generally speaking, both the training data and testing data can affect the classification performance of traffic classifiers, especially the traditional classifiers. With less knowledge of the testing data contained in the training data, the classification performance of the traditional classifiers decreases a lot since they are not good on detecting new "unknown" traffic. The proposed MNBK classifier has much better overall classification performance on accuracy, precision, and recall compared with other traditional classifiers, especially on detecting and predicting "unknown" traffic types.

5.2 Continuous traffic classification

Continuous traffic classification uses a continuous machine learning algorithm to update the classification model based on the continuing arrival of real time traffic. Generally speaking, it should improve the classification performance in terms of accuracy, precision, and recall compared with the traditional traffic classification techniques without considering the continuous ongoing traffic. Figure 22 illustrates a continuous traffic classification model with the proposed MNBK classifier. The continuous traffic classification can be used on real time traffic classification as well. In contrast to the real time classification model described in Figure 15, the continuous traffic classification will consider the distribution of the real time traffic received during previous time interval, use the result to update the learning model, and further update the predefined classification table based on the updated learning model. The newly arrived real time traffic will be classified based on the newly updated classification table.

The advantage of this classification model is that it introduces and integrates the distribution pattern of the "unknown" traffic into the learning model. However, the continuous traffic classification might introduce errors into the original learning model. The error could be accumulated over time with more and more received real time traffic and go out of control. Therefore, two important technologies will be left for further research. The first one is error measurement and control, i.e., how to measure and control the introduced error in each updating of the learning model. The second one is efficiency, i.e., how to efficiently integrate the new distribution feature of the real time traffic into the learning model with minimum computation resources and less classification error.

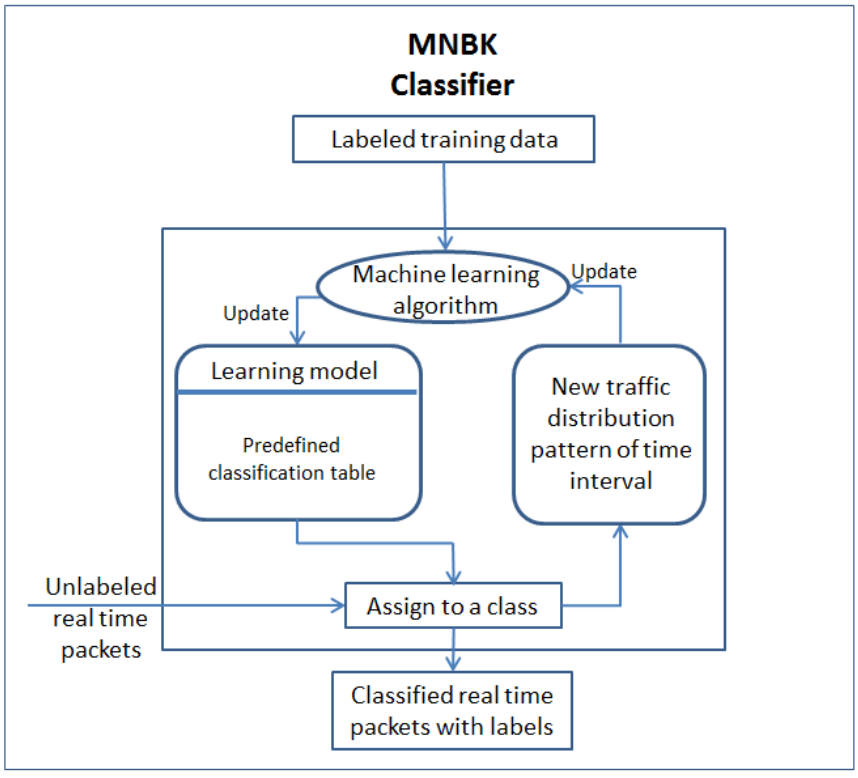


Figure 22: *The continuous traffic classification model based on the proposed MNBK classifier.*

6 Conclusions and future research

Traffic analysis plays a very important role in cyber and electronic warfare. It has been used to provide valuable intelligence about targets and to support decision making in the battlefield such as target selection and cyber effects. Therefore, the accuracy of information prediction in traffic analysis becomes a key factor in mission success.

This report discussed the limitations of the existing well-known supervised traffic classification techniques when there is limited knowledge about the testing data. Such a problem is practical in the battlefield where it is almost impossible to have full knowledge of the adversary's traffic, especially traffic protected with encryption. To improve the traffic classification performance under the above scenarios and to detect unknown traffic that does not belong to the classes contained in the training data, we presented a modified naïve Bayes kernel classifier (MNBK) based on an optimal weight-based (OWB) kernel bandwidth selection algorithm. The classification performance of the proposed MNBK classifier was demonstrated with the traffic generated by the BreakingPoint traffic generator and outperformed traditional classifiers it was compared against. The MNBK classifier not only improves the classification accuracy, precision, and recall significantly but also detects unknown traffic with very high precision and recall performance. In addition, the real time and continuous classification have been discussed by applying our approach to real time traffic.

The following topics have not been addressed in this report and will be left for possible future research. Several optimal thresholds selection related to "unknown" traffic investigation and detection are not discussed in Section 3.4, in particular: zero conditional probability threshold for $f(y|c_i)$, exceptional packet selection threshold r_{th} for "unknown" traffic investigation, and reclassification threshold R_{th} for "unknown" traffic detection. Several topics related to continuous classification need further research, such as error measurement and control for learning model updating, efficient learning model integration with real time traffic, etc.

References

[1] O. Castillo, L. Xu, and S. I. Ao, Trends in Intelligent Systems and Computer Engineering*, Lecture Notes in Electrical Engineering*, Volume 6, Springer US, April 2, 2008.

[2] A. Madhukar and C. Williamson, A Longitudinal Study of P2P Traffic Classification, In Proc. of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Monterey, USA, September 11–14, 2006.

[3] A. Moore and K. Papagiannaki, Toward the Accurate Identification of Network Applications, In Proc. of the Passive and Active Measurement Workshop (PAM 2005), Boston, USA, March 31–April 1, 2005.

[4] T. T. T. Nguyen and G. Armitage, A Survey of Techniques for Internet Traffic Classification Using Machine Learning, *IEEE Communications Surveys & Tutorials*, Vol. 10, No. 4, Fourth Quarter, 2008.

[5] S. Sen, O. Spatscheck, and D. Wang, Accurate, Scalable in Network Identification of P2P Traffic Using Application Signatures, In Proc. of the 13th International Conference on World Wide Web (WWW 2004), New York, USA, May 17–20, 2004.

[6] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann Publisher (An Imprint of Elsevier), June 9, 2011.

[7] J. Erman, A. Mahanti, M. Arlitt, and C. Williamson, Identifying and Discriminating between Web and Peer-to-Peer Traffic in the Network Core, In Proc. of the 16th International Conference on World Wide Web (WWW 2007), Banff, Canada, May 8–12, 2007.

[8] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, Flow Clustering Using Machine Learning Techniques, In Proc. of the Passive and Active Measurement Workshop (PAM 2004), Antibes Juan-les-pins, France, April 19–20, 2004.

[9] S. Zander, T. Nguyen, and G. Armitage, Automated Traffic Classification and Application Identification Using Machine Learning, In Proc. of the 30th IEEE Conference on Local Computer Networks, Sydney, Australia, November 15–17, 2005.

[10] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, Traffic Classification on the Fly, *ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review*, Vol. 36, No. 2, 2006.

[11] P. Domingos and M. Pazzani, Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier, In Proc. of the 13th International Conference on Machine Learning, Bari, Italy, July 3–6, 1996.

[12] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2010.

[13] A. Moore and D. Zuev, Internet Traffic Classification Using Bayesian Analysis Techniques, In Proc. of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Banff, Alberta, Canada, June 6–10, 2005.

[14] T. Nguyen and G. Armitage, Training on Multiple Sub-Flows to Optimize the Use of Machine Learning Classification in Real World IP Networks, In Proc. of IEEE 31st Conference on Local Computer Networks, Tampa, Florida, USA, November 2006.

[15] T. Nguyen and G. Armitage, Synthetic Sub-Flow Pairs for Timely and Stable IP Traffic Identification, In Proc. of Australian Telecommunication Networks and Application Conference, Melbourne, Australia, December 2006.

[16] T. Auld, A. Moore, and S. F. Gull, Bayesian Neural Networks for Internet Traffic Classification, *IEEE Transactions on Neural Networks*, Vol. 18, No. 1, pp. 223–239, January 2007.

[17] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, Kernel Density Estimation via Diffusion, *The Annals of Statistics*, Vol. 38, No. 5, pp. 2916–2957, 2010.

[18] B. W. Silverman, Density Estimation for Statistics and Data Analysis, London, Chapman and Hall/CRC, 1986.

[19] J. Engel, E. Herrmann, and T. Gasser, An Iterative Bandwidth Selector for Kernel Estimation of Densities and Their Derivatives, *Journal of Nonparametric Statistics*, Vol. 4, pp. 21–34, 1994.

[20] BreakingPoint Systems. https://www.ixiacom.com/products/network-security-testing-breakingpoint. Access Date: August 14, 2017.

[21] The Origination and Evolution of Radio Traffic Analysis: The World War I Era, *NSA Declassified Cryptologic Quarterly Articles*, Vol. 6, No. 1, Spring 1987.

[22] The Origination and Evolution of Radio Traffic Analysis: The World War II, *NSA Declassified Cryptologic Quarterly Articles*, Vol. 7, No. 4, Winter 1989.

[23] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, Network Traffic Analysis and Intrusion Detection Using Packet Sniffer, In Proc. of the 2nd IEEE International Conference on Communication Software and Networks (ICCSN 2010), Singapore, February 26–28, 2010.

[24] I. Butun, S. D. Morgera, and R. Sankar, A Survey of Intrusion Detection Systems in Wireless Sensor Networks, *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 1, First Quarter, 2014.

[25] B. Panneton and J. Adamets, High-Bandwidth Tactical-Network Data Analysis in a High-Performance-Computing (HPC) Environment: Packet-Level Analysis, US Army Research Laboratory, ARL-CR-0779, September 2015.

[26] Z. Lu, C. Wang, and M. Wei, On Detection and Concealment of Critical Roles in Tactical Wireless Networks, In Proc. of the 2015 IEEE Military Communication Conference (MILCOM 2015), Tampa, USA, October 26–28, 2015.

[27] Y. Liu, D. R. Bild, R. P. Dick, Z. M. Mao, and D. S. Wallach, The Mason Test: A Defense Against Sybil Attacks in Wireless Networks Without Trusted Authorities, *IEEE Transactions on Mobile Computing*, Vol. 14, No. 11, pp. 2376–2391, 2015.

[28] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, Machine Learning Applications in Cancer Prognosis and Prediction, *Computational and Structural Biotechnology Journal, Elsevier*, Vol. 13, pp. 8–17, 2015.

[29] K. R. Foster, R. Koprowski, and J. D. Skufca, Machine Learning, Medical Diagnosis, and Biomedical Engineering Research – Commentary, BioMedical Engineering OnLine, July 5, 2014.

[30] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour, Dropout Improves Recurrent Neural Networks for Handwriting Recognition, In Proc. of the 14th IEEE International Conference on Frontiers in Handwriting Recognition, Heraklion, Greece, September 1–4, 2014.

[31] A. Graves, A. Mohamed, and G. Hinton, Speech Recognition with Deep Recurrent Neural Networks, In Proc. of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, Canada, May 26–31, 2013.

[32] J. Wu, K. M. Williams, H. Chen, M. Khabsa, C. Caragea, S. Tuarob, A. G. Ororbia, D. Jordan, P. Mitra, and C. L. Giles, CiteSeerX: AI in a Digital Library Search Engine, *AI Magazine*, Vol. 36, No. 3, 2015.

[33] K. Williams, L. Li, M. Khabsa, J. Wu, P. Shih, and C. L. Giles, A Web Service for Scholarly Big Data Information Extraction, In Proc. of the 21st IEEE International Conference on Web Services, Anchorage, USA, June 27–July 2, 2014.

[34] R. Kohavi, Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid, in Proc. of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, Oregon, August 2–4, 1996.

[35] R. Bouckaert, Bayesian Network Classifiers in Weka, Technical Report, Department of Computer Science, Waikato University, Hamilton, NZ, 2005.

[36] R. Kohavi and J. R. Quinlan, Data Mining Tasks and Methods: Classification: Decision-Tree Discovery, In Handbook of Data Mining and Knowledge Discovery, Editors: W. Klosgen and J. M. Zytkow, Oxford University Press, pp. 267–276, 2002.

[37] A. Gut, An Intermediate Course in Probability, 2nd Edition, Springer-Verlag, New York, 2009.

[38] I. H. Witten, E. Frank, M. Hall, and C. Pal, Data Mining: Practical Machine Learning Tools and Techniques, 4th Edition, Morgan Kaufmann Publisher (An Imprint of Elsevier), November 17, 2016.

[39] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proceedings of IEEE*, Vol. 90, Issue 7, 2002.

[40] P. Janssen, J. S. Marron, N. Veraverbeke, and W. Sarle, Scale Measures for Bandwidth Selection, *Journal of Nonparametric Statistics*, Vol. 5, Issue 4, pp. 359–380, 1995.

[41] P. V. Kerm, Adaptive Kernel Density Estimation, *Stata Journal*, Vol. 3, No. 4, pp. 148–156, 2003.

[42] R. J. Karunamuni and S. Zhang, Some Improvements on a Boundary Corrected Kernel Density Estimator, *Statistics Probability Letter*, Vol. 78, pp. 499–507, 2008.

List of symbols/abbreviations/acronyms/initialisms

DNS	Domain name system
HTTP	Hypertext transfer protocol
KDE	Gaussian kernel density estimation
MNBK	Modified naïve Bayes kernel classifier
NBK	Naïve Bayes kernel estimation
OWB	Optimal weight-base kernel bandwidth selection
ROT	Rules-of-thumb
ROTm	Rules-of-thumb bandwidth selection with median absolute deviation
ROTs	Rules-of-thumb bandwidth selection with standard deviation
STE	Solve-the-equation
TCP	Transmission control protocol
TLS	Transport layer security
XMPP	Extensible messaging and presence protocol

DOCUMENT CONTROL DATA

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)

DRDC – Ottawa Research Centre
Defence Research and Development Canada
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
Canada

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

Traffic analysis on encrypted traffic over wireless channels: Traffic classification based on partial knowledge

4. AUTHORS (Last name, followed by initials – ranks, titles, etc., not to be used)

Song, R.; Willink, T.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)	6b. NO. OF REFS (Total references cited.)
November 2018	46	42

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

Scientific Report

8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)

DRDC – Ottawa Research Centre
Defence Research and Development Canada
3701 Carling Avenue
Ottawa, Ontario K1A 0Z4
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
TNO - Tactical Network Operations	

10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2018-R196	

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

Traffic Analysis; Traffic Classification; Machine Learning; Gaussian Kernel Density; Bayes Theorem; Naïve Bayes Classifier; Naïve Bayes Kernel Estimation

13. ABSTRACT (When available in the document, the French version of the abstract must be included here.)

This Scientific Report investigates the limitations of traditional supervised traffic classification techniques on classification performance such as accuracy, precision, and recall when there is only limited knowledge available about the traffic, especially an adversary's encrypted traffic. To improve the classification performance under the above scenario, a new modified naïve Bayes kernel (MNBK) classifier is proposed based on optimal weight-based (OWB) kernel bandwidth selection. The proposed OWB kernel bandwidth selection algorithm can make a more accurate learning model for traffic classification compared with the traditional classifiers. By generating several possible major traffic types in tactical edge networks, we demonstrate that the proposed MNBK classifier not only improves the classification performance on the existing classes significantly, but also detects unknown traffic with very high accuracy, precision, and recall compared with the traditional classifiers. In addition, a learning classification model is proposed based on MNBK, that processes received ongoing real time traffic and updates the classification table periodically. Generally speaking, with more and more accurate information retrieved from received real time traffic, the proposed real time classification model should improve the classification performance over time compared with the traditional classifiers that do not consider the ongoing received traffic. This has been demonstrated with our classification performance evaluation.

Le présent rapport traite des limites au rendement des techniques traditionnelles de classification du trafic supervisé en ce qui a trait à l'exactitude, à la précision et au rappel lorsqu'on dispose de peu d'information sur le trafic, particulièrement sur le trafic crypté d'un adversaire. Le rapport présente également une nouvelle méthode de classification naïve bayésienne modifiée basée sur la sélection de la taille du noyau selon une pondération optimale afin d'améliorer le rendement en matière de classification dans un tel cas. Le nouvel algorithme de sélection de la taille du noyau nous permettra d'instaurer un modèle d'apprentissage pour la classification du trafic qui sera plus précis que les modèles traditionnels. En générant plusieurs grands types de trafic pour les réseaux tactiques en périphérie, la méthode proposée permettra non seulement d'améliorer considérablement le rendement en matière de classification à l'aide des classes existantes, mais également de détecter le trafic inconnu avec beaucoup plus d'exactitude, de précision et de facilité de rappel que les méthodes traditionnelles. Un nouveau modèle de classification par apprentissage est proposé. Celui-ci est fondé sur une méthode naïve bayésienne modifiée qui permettra de traiter le trafic en temps réel et de mettre à jour périodiquement le tableau de classification. Globalement, au fur et à mesure que l'on recueillera de l'information de plus en plus exacte provenant du trafic en temps réel le modèle proposé permettra avec le temps d'améliorer le rendement de la classification par rapport aux modèles traditionnels qui ne tiennent pas compte du trafic continu, comme l'a démontré notre évaluation du rendement en matière de classification.