Strengthening intrusion detection system for adversarial attacks

Pimsarn, Chutipon; Boongoen, Tossapon; Iam-On, Natthakan; Naik, Nitin; Yang, Longzhi

Published in:
Complex & Intelligent Systems

DOI:
10.1007/s40747-022-00739-0

Publication date:
2022

Citation for published version (APA):
Pimsarn, C., Boongoen, T., Iam-On, N., Naik, N., & Yang, L. (2022). Strengthening intrusion detection system for adversarial attacks: Improved handling of imbalance classification problem. *Complex & Intelligent Systems*, *8*(6), 4863-4880. https://doi.org/10.1007/s40747-022-00739-0

Complex & Intelligent Systems (2022) 8:4863–4880
https://doi.org/10.1007/s40747-022-00739-0

ORIGINAL ARTICLE

Strengthening intrusion detection system for adversarial attacks: improved handling of imbalance classification problem

Chutipon Pimsarn[1] · Tossapon Boongoen[1,2] · Natthakan Iam-On[2] · Nitin Naik[3] · Longzhi Yang[4]

Received: 3 August 2021 / Accepted: 8 April 2022 / Published online: 25 April 2022

Abstract

Most defence mechanisms such as a network-based intrusion detection system (NIDS) are often sub-optimal for the detection of an unseen malicious pattern. In response, a number of studies attempt to empower a machine-learning-based NIDS to improve the ability to recognize adversarial attacks. Along this line of research, the present work focuses on non-payload connections at the TCP stack level, which is generalized and applicable to different network applications. As a compliment to the recently published investigation that searches for the most informative feature space for classifying obfuscated connections, the problem of class imbalance is examined herein. In particular, a multiple-clustering-based undersampling framework is proposed to determine the set of cluster centroids that best represent the majority class, whose size is reduced to be on par with that of the minority. Initially, a pool of centroids is created using the concept of ensemble clustering that aims to obtain a collection of accurate and diverse clusterings. From that, the final set of representatives is selected from this pool. Three different objective functions are formed for this optimization driven process, thus leading to three variants of FF-Majority, FF-Minority and FF-Overall. Based on the thorough evaluation of a published dataset, four classification models and different settings, these new methods often exhibit better predictive performance than its baseline, the single-clustering undersampling counterpart and state-of-the-art techniques. Parameter analysis and implication for analyzing an extreme case are also provided as a guideline for future applications.

Keywords Intrusion detection system · Adversarial attack · Machine learning · Imbalance classification · Data clustering

Introduction

As the world becomes more interconnected, web-based applications like personalized online banking [2], industrial control systems [8], Internet of things [18] and wireless sensor networks [41] are subject to various security vulnerabilities, thus raising an urgent need for effective network and information security measures [35,39]. For this, a number of attempts have developed different defence mechanisms such as a hardware/software firewall and an intrusion detection system (IDS) to safeguard assets as well as system/device control in a cyberspace [6,22,67]. Making use of unpatched services is a common instance of intrusive attacks that remain one of the crucial threats to both individuals and organizations [73]. This is highly critical to the Internet-based monitoring of engineering systems in general, especially in the case of critical national infrastructure such as healthcare, manufacturing, power grid, gas and oil refineries [9]. Recently, a similar approach is proposed to detect an intrusive attack to an in-vehicle communication system, the controller-area-network bus protocol in particular [42].

As a response to this challenge, a network-based intrusion detection system (NIDS) has been continuously invented and improved to provide security by monitoring network traffics and identifying malicious connections [44]. To support a timely intervention and useful information for human operators, an NIDS is required to be resilient to new breeds of threat like polymorphism, which allows an exploit-code to

✉ Tossapon Boongoen
tossapon.boo@mfu.ac.th

1 Center of Excellence in AI and Emerging Technologies, School of Information Technology, Mae Fah Luang University, Chiang Rai 57100, Thailand

2 Department of Computer Science, Aberystwyth University, Aberystwyth, UK

3 School of Informatics and Digital Engineering, Aston University, Birmingham, UK

4 Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, UK

avoid positive signature matching. At large, initial signature-based systems are ineffective for the detection of mutated patterns and zero-day attacks, with a common problem of high false positive rate [10,61]. This lacking leads to a family of machine-learning-based NIDS that has proven to be more effective provided up-to-date connection samples and corresponding expert-directed labels [52,58]. Over the years, different classification models have been investigated for this task that is sometimes referred to as an anomaly-based approach [36]. These include support vector machine or SVM [75], k-means clustering [76], artificial neural networks or ANN [63] and classifier ensemble [23]. Despite their successes reported in the literature, the aforementioned methods are capable of detecting some unknown intrusions that partly resemble those known to the NIDS, but they are still prone to adversarial attacks that exploit a variety of new obfuscation techniques [17,30]. These attacks are branded as adversarial machine learning, which explores blind spots of a machine learning-based system [8]. In particular, slight perturbations are introduced to new connections such that the pre-trained model may incorrectly justify a decision boundary, thus a predicted class [68].

Without the knowledge of obfuscation methods, one has to develop NIDS that is more flexible to recognize modified traffic patterns. One way to achieve this is to constantly update the underlying classifier with new training instances, while the other focuses on learning more from the attack class that is commonly a minority in a training data collection [7,20]. Specific to the latter, a number of studies have made use of techniques developed within the data science community to handle this issue, called class imbalance [36]. In general, a conventional oversampling technique like synthetic minority oversampling or SMOTE [15] and its variants have been a common choice to cope with this problem for different classification algorithms [23,63,75]. However, the resulting model shows signs of overfitting as it is not generalized to unseen data [45]. To this extent, an undersampling strategy has proven more robust to overfitting across different domain problems [11,40]. Following the work of [46] that introduces an initial model of clustering-based undersampling, the multiple clustering-based approach proposed in this paper aims to explore a pool of representative candidates, which is richer than that obtained from one clustering. As such, it may better preserve useful data patterns presented in the majority class, hence the improved predictive performance [65].

Assumption Following the initial research of [30], this paper focuses on classification-based NIDS, which does not analyze payload data [17]. It considers objects of TCP connections, not at a lower level of individual packets. It is assumed that an adversary possesses knowledge about the design of the victim's system, but can only mutate inputs to this system such that they still conform with the TCP/IP protocol specification. Based on the report of [28], this is achieved by designing non-payload-based obfuscation techniques to work at network and transport layers. These will mutate samples of known intrusions in an exploit-independent way, thus making attacks looking similar to a legitimate one. This approach is on par with evasions of the measurement phase of IDS, defined in the taxonomy of adversarial attacks against IDS [16].

Problem and Scope: Given the data collection generated within the study of [30], the current work aims to develop an effective clustering-based undersampling method, which is novel as well as more accurate than the baseline [46], other well-known methods found in the literature [60,66] and the standard oversampling counterpart [15]. This new approach selects representative samples from multiple clusterings that are generated with various settings to promote the diversity between them. This follows the concept and success reported for the field of consensus clustering [31,54]. In particular, this selection is designed as an optimization problem, where the best alternative is added to the target sampling set iteratively. It is applied to reduce the cardinality of instances belonging to the majority class of the binary classification problem, i.e., legitimate and attack connections are majority and minority classes. Following that, a classification model developed from the balanced training set is employed to categorize unseen and obfuscated attacks. Intuitively, the chance of recognizing these modified patterns may be improved, as the classifier is able to learn more from the smaller class, while information loss due to undersampling the majority class is minimized.

Contributions: Main contributions of this paper are summarized below.

- This work extends the previous research of [30] to develop an accurate classification-based NIDS that is robust to adversarial attacks. It presents a new multiple clusterings-based method to handle imbalance classification through undersampling the majority class. The proposed framework allows any classification algorithm to learn better with the minority class that represents attack connections. As such, the resulting model becomes more effective in recognizing obfuscated intrusions, whose appearances partly overlap with those known direct attacks. In addition, the resulting framework is generalized to a wider range of non-payload adversarial attacks.
- The proposed undersampling technique is based on an iterative and greedy-optimization process of selecting the best alternative from a pool of centroids that represent different clustering results or data partitions. This is inspired by the concept of ensemble or consensus clustering [31,34], which usually provides a more accurate

data partition than a single clustering. Hence, this idea is novel and may enhance the effectiveness of an initial work of [46] that makes use of only one clustering to guide the sampling procedure. For the aforementioned selection, it is designed as an optimization problem that maximizes three different objective functions of: (i) the maximum distance from the centroid under the question to other k nearest centroids in the pool, (ii) the maximum distance from the examined centroid to k nearest instances from the minority class, and (iii) the average between functions (i) and (ii), respectively.

- The proposed model is evaluated against the single-clustering alternative of [46], RUS (Random UnderSampling [60]) as the conventional model of undersampling, other well-known ensemble-level techniques: RUSBoost [60] and IRUS (Inverse Random UnderSampling [66]), the oversampling technique of [15] and its recent extension [70]. To generalize this experimental investigation, a range of basic classifiers and classifier ensemble method are employed. Specific to the line of IDS research, a state-of-the-art model is also included as a comparison method. Furthermore, parameter analysis specific to the proposed technique is also included such that the relation between predictive performance and algorithmic variables is illustrated and discussed. Anyone tries to apply this to a future problem would find it helpful as to tailor the model for good performance.

The rest of this paper is organized as follows. Section 2 emphasizes related works and problem definition, with details of the dataset exploited herein. Besides, Sect. 3 presents the proposed undersampling framework, including the underlying optimization process. The performance evaluation of new and compared methods are included and discussed in Sect. 4. At the end, the conclusion with directions of future research are given in Sect. 5. In addition, Table 1 provides a description of abbreviations used in this paper.

Table 1 List of abbreviations used throughout this paper

Abbreviation	Description
AdaBoost	AdaBoost technique for boosting-based ensemble generation
ANN	Artificial neural networks
ASNM	Advanced security network metrics
C4.5	A decision tree algorithm
DNN	Deep neural networks
DT	Decision tree
HTTP	Hypertext transfer protocol
IDS	Intrusion detection system
IP	Internet protocol
IRUS	Inverse random undersampling
KNN	K nearest neighbours
LDA	Linear discriminant analysis
LR	Logistic regression
MTU	Maximum transmission unit
NB	Naive Bayes
NIDS	Network-based intrusion detection system
PCA	Principal component analysis
RF	Random forest
RUS	Random undersampling
RUSBoost	Random undersampling with boosting-based ensemble generation
SMOTE	Synthetic minority oversampling
SVM	Support vector machine
TCP	Transmission control protocol
TCP FIN	TCP finish flag
TCP URG	TCP urgent flag

Related works

In this section, background and related works of the proposed research regarding classification-based NIDS, traffic object of interest and non-payload-based obfuscations is elaborated to set a scene for the rest of paper. Then, the definition of NIDS as a classification problem is given as a basis for the proposed method, which is fully explained in Sect. 3.

Classification-based NIDS and handling of imbalance class problem

With an increasing demand for NIDS along with the development of Internet-of-things and network and wireless sensor networks [18,25], a series of intelligent agents to cope with intrusive traffics have been the center of many research works. In general, these systems rely on either signature matching to determine misuse and attack patterns that are largely different from signatures of legal activities [3], or a pattern recognition modeling normally referred to as anomaly detection-based intrusion detection [47]. To get over the problem with a low recall of unseen attacks, most of NIDS instances have turned to exploit a predictive model learned from an up-to-date data collection [24,71]. However, the major limitation is with a high false-positive rate caused by the difficulty in determining boundary between normal and abnormal cases [57]. Note that other approaches to NIDS such as watchdogs and trust models are also presented in the literature, but are beyond the scope of the current work that

Table 2 Summarization of related works, with respect to classification-based NIDS & treatment of class imbalance

Proposed work	Exploited technique	Treatment of class imbalance
Online oversampling PCA for anomaly detection [43]	PCA	Oversampling
Adaptive ensemble learning model [23]	DT, RF, KNN & DNN	n/a
Handling of imbalance class in IDS [1]	RF and DNN	Oversampling & undersampling
Hybrid data mining approach [55]	Clustering, feature selection & KNN	Oversampling
Information-Gain based feature selection [69]	RF & feature selection	Oversampling
Filter-based attribute reduction [14]	K-means & SVM	n/a
Handling of imbalance class in IDS [36]	DT, RF, KNN & LDA	Oversampling
Determining informative features [19]	RF & feature selection	n/a
Recursive feature elimination [62]	SVM, DT & RF	n/a
Feature reduced IDS [4]	ANN	n/a
Artificial bee colony for improved classifier ensemble [48]	DT & AdaBoost	n/a
Forwarding feature selection [30]	DT, NB, LR & SVM	n/a

concentrates solely on machine learning-based implementations. Within this particular subject, a classification model has been an effective alternative that blends advantages of both signature-based and anomaly-based methods.

For instance, in the study of [36], a number of machine learning algorithms and the oversampling technique of SMOTE [15] are used to improve the detection rate for minority attack classes. This recent investigation is among some in the NIDS field that pay attention to the problem of class imbalance. It inspires the current research, with the summarization of related works being presented in Table 2 (please refer to reviews of [41,49] for a boarder scope of NIDS research).

These representative methods point out major directions taken to improve the classification performance with the presence of class imbalance. Some such as the studies of [1,43] deal with this problem directly by using conventional oversampling and undersampling methods like SMOTE [15] and RUS [60] to obtain a desired training dataset. On the other hand, many techniques get over this burden implicitly through the determination of feature space [4,14,19,30,62], and the ensemble methodology that learns from multiple data perturbations [23,48]. The ability to handling the problem of class imbalance becomes even more critical as a classification-based NIDS encounters adversarial attacks, where known intrusive patterns are mutated to avoid a positive recognition. In fact, those methods listed in Table 2 are capable of detecting some unknown intrusions, but it is still prone to evasion by obfuscation techniques [17,30]. The next section provides further details on this attack type, especially the case of non-payload-based obfuscation that is the focus of the present study.

Adversarial attack and non-payload-based obfuscation

Based on the taxonomy of evasion against machine learning models [10], two major categories of evasive adversarial attacks against IDS can be highlighted, payload-based and non-payload-based approaches. As an initial investigation into the former, the Whisker tool is developed to mutate HTTP requests such that IDS becomes confused and inaccurate [17]. Similar works have been introduced to bypass the detection of IDS through changing the payload [16,72], using obfuscation techniques such as malware morphism [50,51,78]. However, those methods that are able to evade payload-based IDS mainly by morphing the payload may not be efficient for non-payload-based counterpart. Provided this, it is also necessary to recover defected patterns associating to the attack morphing at network and transport layers of the TCP/IP stack [30]. As non-payload-based IDS is usually more cost-effective as well as generalized to network and application settings than the other, it motivates a large number of research works, such as Protocol Scrubbing [74] and AGENT [59], for instance. Recently, another series of investigations [28,30] on non-payload-based intrusion detection and obfuscation-based adversarial evasion has been reported. These assess the robustness of different ML models, based on experiments with the dataset that implements obfuscation techniques to simulate adversarial non-payload attacks. Note that a thorough feature engineering process, i.e., a wrapper method [13], is designed to deliver a set of informative features to train a classifier with both original samples and mutated ones. Despite the good result reported therein, the

issue of imbalance classification has not been clarified and handled.

As mentioned earlier, the current work investigates instances of TCP connection, not network packets, which represent application data exchanges between client and server on the TCP/IP stack up to the transport layer. Of course, these are subject to connection-oriented protocol TCP at Layer 4, Internet protocol IP at Layer 3 and Ethernet protocol at Layer 2, respectively. In particular, a TCP connection γ can be presented by start and end timestamps, ports/IP addresses of the client and the server, sets of packets interchanged between the two ends. Given this assumption, a connection γ can be explained with its network connection features, thus allowing the following extraction function to map γ to the feature space Λ of d dimensions [30].

$$f(\gamma) \mapsto \Lambda,\ \Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_d) \tag{1}$$

Each function $f_i(\gamma)$ maps the connection to the i^{th} dimension as follows.

$$f_i(\gamma) \mapsto \lambda_i,\ i = \{1, 2, \ldots, d\}, \tag{2}$$

According to [28,30], examples of these features include the standard deviation of outbound (client to server) packet sizes, modus of TCP header lengths in all traffic, the number of TCP URG flags occurred in inbound traffic, and the number of TCP FIN flags occurred in inbound traffic.

Having specified those, the next part describes the concept of non-payload-based obfuscation, which aims to modify connection characteristics or features for a remote attack. For a connection γ_a representing a remote attack without any obfuscation, it can be represented by the following equation.

$$f(\gamma_a) \mapsto \Lambda^a,\ \Lambda^a = (\lambda_1^a, \lambda_2^a, \ldots, \lambda_d^a) \tag{3}$$

Then, let turn to the connection $\gamma_{a'}$ that corresponds to an intrusive communication γ_a to which non-payload-based obfuscations are applied. These modifications make changes to packet sets of the original connection γ_a by insertion, removal and transformation of the packets. As such, the previous feature space Λ^a is transformed to $\Lambda^{a'}$.

$$f(\gamma_{a'}) \mapsto \Lambda^{a'},\ \Lambda^{a'} = (\lambda_1^{a'}, \lambda_2^{a'}, \ldots, \lambda_d^{a'}) \tag{4}$$

As a result, a classifier trained without knowledge of obfuscated or modified patterns may not perform as well as it does against intrusive connections with original features. According to the study of [29], a set of techniques have been initiated as part of developing an obfuscation tool in the Unix environment. Examples of functions $f(\gamma_{a'})$ are listed below.

- Spread out packets in time: constant delay of 1 and 8 s., as well as the normal distribution of delay with 5 s. mean with 2.5 s. standard deviation (25% correlation)
- Packets loss: 25% of packets
- Unreliable network channel simulation: 25% of packets damaged, 35% of packets damaged, and 35% of packets damaged with 25% correlation
- Packets duplication: 5% of packets
- Packets order modifications: reordering of 25% and 50% packets; reordered packets are sent with 10 ms. delay and 50% correlation
- Fragmentation: MTU 1000, MTU 750, MTU 500 and MTU 250
- Combinations of the aforementioned techniques

In accordance with previous studies of classification-based NIDS [63,75,76], let a training dataset $X = V \times Y$ be the space of labeled connection instances, where V denotes the feature space of n instances ($V \in R^{n \times d}$), Y represents the corresponding label space of size $n \times 1$, and each entry $x_i \in V$ is classified as $y_i \in Y$ with the value of y_i being drawn from the domain of class D_X. For a classifier that is trained on the dataset X using the algorithm α, the resulting model CF_X^α estimates the class $y_o \in D_X$ of a new instance $x_o \in R^{1 \times d}$, i.e., $CF_X^\alpha(x_o) = y_o$.

Specific to the context of NIDS as a binary classification problem, the predicted class y_{γ_a} of a connection γ_a whose feature vector is defined as $f(\gamma_a)$, can be defined by the following [30].

$$y_{\gamma_a} = CF_X^\alpha\ (f(\gamma_a)) \tag{5}$$

where $y_{\gamma_a} \in \{\text{Intrusion, Legitimate}\}$. Now with a connection $y_{\gamma_{a'}}$ whose features are modified through obfuscation functions, the quality of prediction $y_{\gamma_{a'}}$ is the subject worth investigating.

$$y_{\gamma_{a'}} = CF_X^\alpha\ (f(\gamma_{a'})) \tag{6}$$

Without any prior knowledge regarding mutated patterns or model adjustment, any classifier CF_X^α can often be sub-optimal. To this extent, the original work of [30] overcomes this difficulty through a feature selection approach, which iteratively add highly informative features to the desired set. This leads to an improvement towards a robust classification of adversarial intrusions but lacks an appropriate handling of the class imbalance problem. Henceforth, a new undersampling method is proposed in the next section to attend to this issue.

Proposed method

The proposed undersampling framework can be considered as an extension to the initial work of [46], which employs a single clustering to determine representative instances of original samples belonging to the majority class. At the same time, it demonstrates an organic application of consensus clustering concept [31,32] to provide a pool of multiple clustering results from which better cluster-wise representatives can be extracted. This section provides details of different stages developed for this new method, including the creation of multiple clustering results, the selection of representatives from those data partitions, and the use of data after undersampling to train a classifier.

Generating a Pool of Multiple Clusterings. Provided the dataset X, let X_0 and X_1 (where $X_0 \cup X_1 = X$) be the set of samples belonging to the majority class and that of the minority class, respectively. An undersampling technique χ is applied to X_0 to extract the set of representative samples $\chi(X_0) = X_0^*$, where $|X_0^*| \ll |X_0|$ and $|B|$ denotes the size of set B. Then, the target data X^* that can be formed as $X^* = X_1 \cup X_0^*$ is used to create a classifier $CF_{X^*}^{\alpha}$ using the classification algorithm α (previously, this is CF_X^{α} without the undersampling process). Figure 1 illustrates the overall process of undersampling the training data set. Based on the study presented by [46], the function χ is simply represented by a set of centroids Z obtained from a clustering of X_0, where the k-means technique is a common alternative for this analysis. In other words, representative samples are centroids $z_1, z_2, \ldots, z_\rho$ from clustering X_0 using k-means and the preferred number of clusters as ρ. Note that ρ is set to the desired size of the majority class, which normally is the same as that of the other, i.e., $\rho = |X_1|$.

Specific to the current work, the concept ensemble or consensus clustering [31] is exploited to create a pool of multiple clusterings, from which a collection of ρ clusters are selected to generate X_0^*. Let $V_0 = \{x_1, x_2, \ldots, x_{n_0}\}$ be the matrix in the normalized domain $[0, 1]^{n_0 \times d}$ of n_0 legitimate connection instances with respect to d features. It is noteworthy that the label space Y_0 from a training dataset $X_0 = V_0 \times Y_0$ will not be exploited here as a clustering process is an unsupervised model, which develop a data partition (i.e., a set of clusters) without the knowledge of class information. In addition, each sample $x_i \in V_0$ is represented as a vector of d feature dimensions or $x_i = (x_{i1}, \ldots, x_{id}), \forall i \in \{1, 2, \ldots, n_0\}$. Also let $\Pi = \{\pi_1, \pi_2, \ldots, \pi_M\}$ be a cluster ensemble with M base clusterings, i.e., a clustering result or ensemble member. In particular, the g^{th} member delivers a collection of clusters $\pi_g = \{C_1^g, C_2^g, \ldots, C_{k_g}^g\}$, where $\bigcup_{t=1}^{k_g} C_t^g = X_0$, $\bigcap_{t=1}^{k_g} C_t^g = \emptyset$, and k_g is the number of clusters in partition π_g. To ensure the diversity within Π, the following ensemble generation strategies are used together.

- Random-k method: these clusterings are generated using k-means with a cluster number that is randomly chosen from the range $\{2, \ldots \sqrt{|X_0|}\}$ or $\{2, \ldots 50\}$ when $\sqrt{|X_0|} > 50$ (see the report of [32] for relevant details).
- Random-subspace method: each base clustering can be generated from a random feature subspace $V_0' \in [0, 1]^{n_0 \times d'}$ of the feature space V_0. Each of the data

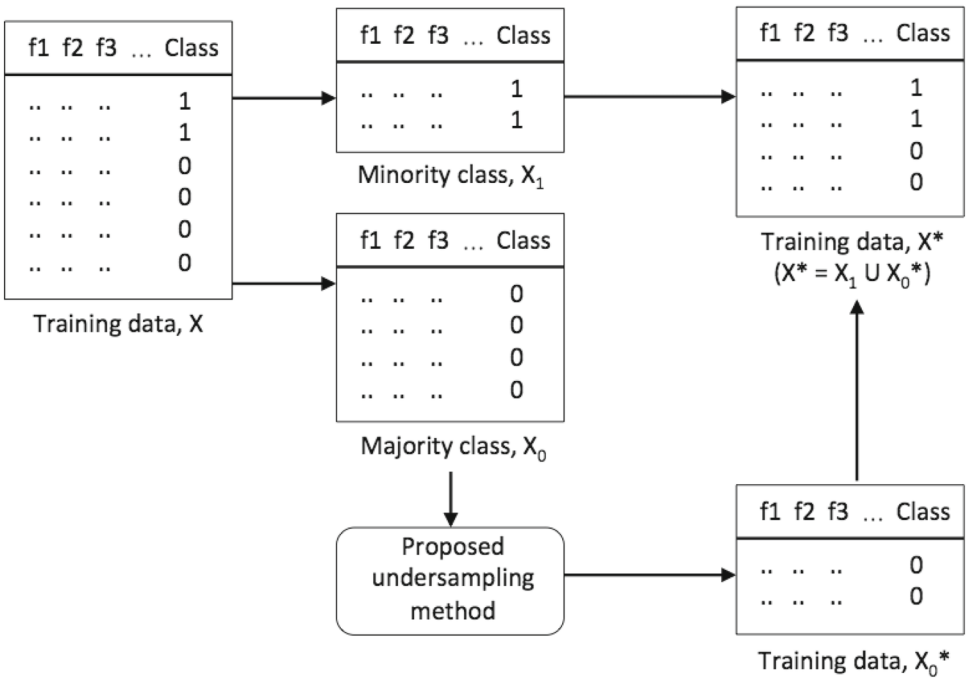

Fig. 1 Overview of the undersampling process

subspaces is created with respect to the following interval d'.

$$d' = d'_{min} + \lfloor \epsilon (d'_{max} - d'_{min}) \rfloor, \tag{7}$$

provided that $\epsilon \in [0, 1]$ is a uniform random variable, while d'_{min} and d'_{max} denote the lower and upper bounds of the generated subspace V'_0. Following the initial work of [33], d'_{min} and d'_{max} are set to $0.75d$ and $0.85d$, respectively. With this being decided, one of d features is picked up at a time to form the desired subspace of d' non-duplicated features is obtained. For that, the index of each randomly selected feature is determined by the following.

$$h = \lfloor 1 + \eta D \rfloor, \tag{8}$$

where h denotes the h^{th} feature in the pool of d attributes and $\eta \in [0, 1)$ is another uniform random variable. As a working example, Fig. 2 summarizes the process of generating a set of representatives from a clustering of two clusters, while Fig. 3 extends this procedure to a selection of representatives from a pool of centroids generated by multiple clusterings. Note that the underlying selection process will be explained next.

Selecting Cluster-Wise Representatives. Having obtained the ensemble Π, a pool of centroids Z_Π is created such that each centroid $z_a \in Z_\Pi$ represents a particular cluster C_a in the set $\{C^1_1, C^1_2, \ldots, C^1_{k_1}\} \cup \{C^2_1, C^2_2, \ldots, C^2_{k_2}\} \cup \ldots \{C^M_1, C^M_2, \ldots, C^M_{k_M}\}$. Note that every centroid in this collection is represented by the original space of d features, i.e., $z_a \in [0, 1]^{1 \times d}, \forall z_a \in Z_\Pi$. This processing stage is to select ρ of these centroids to form the target set of representative samples $Z = \{z_1, z_2, \ldots, z_\rho\}$. It can be summarized by the following steps.

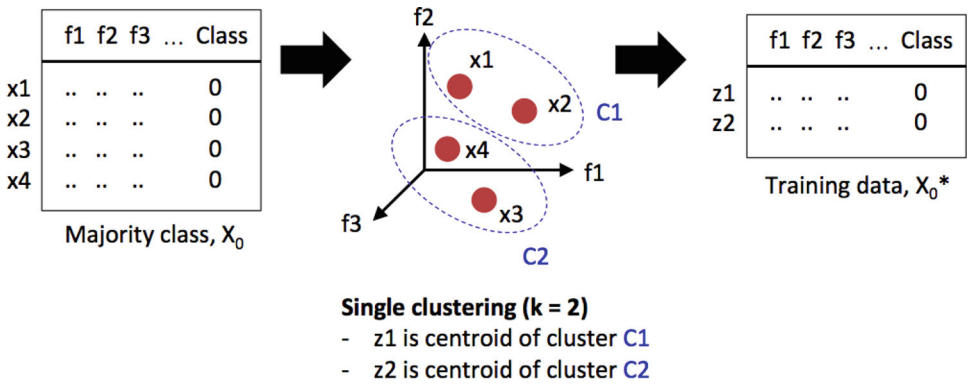

Fig. 2 The process of generating a set of representative centroids $z1$ and $z2$ from a single clustering with $k = 2$ and $X_0 = x1, x2, x3, x4$

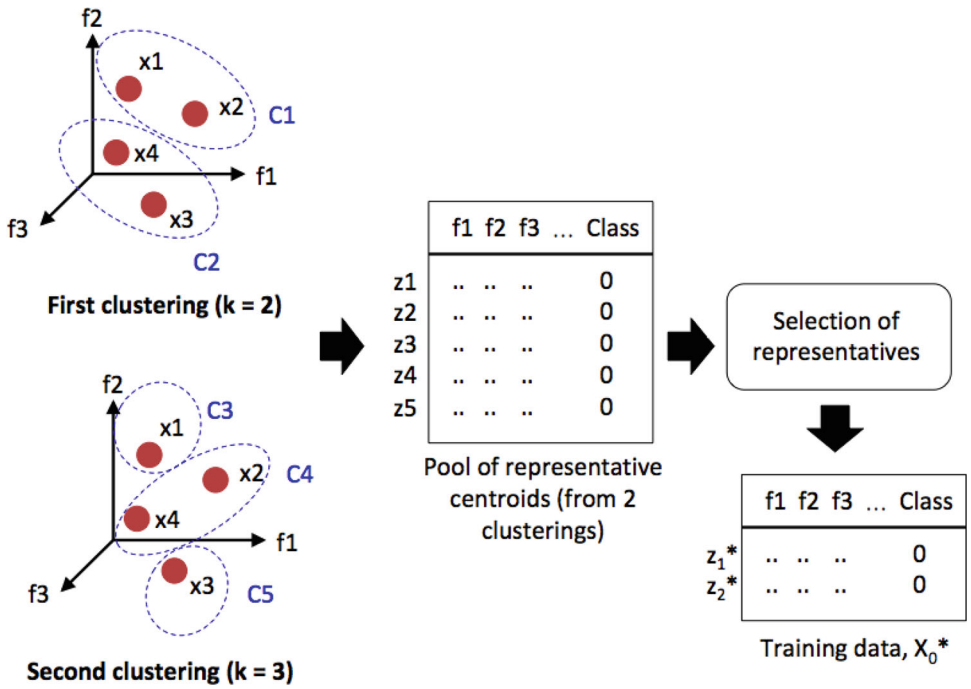

Fig. 3 The process of generating a set of representative centroids from two clusterings with $k = 2$ and $k = 3$

- *Step 1:* To start with, the first member z_1 of Z is selected from the Z_Π, provided that z_1 maximizes the average distance to other centroids in Z_Π.

$$z_1 = \underset{\forall z_a \in Z_\Pi}{\operatorname{argmax}} \frac{\sum_{\forall z_b \in Z_\Pi, z_b \neq z_a} d(z_a, z_b)}{|Z_\Pi| - 1} \tag{9}$$

  where $d(x_i, x_j)$ denotes the distance between two samples x_i and x_j. Note that the Euclidean metric is employed in this research to estimate the distance measurement. In addition, $|A|$ represents the size of set A. At the end of this step, $|Z|$ is 1, while that of Z_Π is reduced by 1 as well, i.e., $Z_\Pi = Z_\Pi - z_1$.
- *Step 2:* Next, a new member is iteratively chosen from Z_Π and moved to Z. To be exact, in each iteration, a greedy optimization is exploited to determine the best centroid $z_c \in Z_\Pi$ using one of three different objective functions defined below. As a result, the two sets are updated by $Z = Z \cup z_c$ and $Z_\Pi = Z_\Pi - z_c$, respectively. This is repeated until all ρ members are obtained, i.e., $|Z| = \rho$. The following equation describes the first objective function, called *Furthest-First-Majority* or shortly as *FF-Majority*(Z, Z_Π). This is to find a centroid representing a unique feature space that is minimally overlapping with those of others in Z_Π. It is leveraged with a similar assessment to existing members of Z, which is the first term in this function.

$$z_c = \underset{\forall z_a \in Z_\Pi}{\operatorname{argmax}} \left(\frac{\sum_{\forall z_e \in Z} d(z_a, z_e)}{|Z|} + \frac{\sum_{\forall z_b \in Z_\Pi, z_b \neq z_a} d(z_a, z_b)}{|Z_\Pi| - 1} \right) \tag{10}$$

  The second objective function, called *Furthest-First-Minority* or shortly as *FF-Minority*(Z, Z_Π, X_1). It finds a centroid in Z_Π that is largely different from samples of the minority class X_1. This can be formally specified as follows.

$$z_c = \underset{\forall z_a \in Z_\Pi}{\operatorname{argmax}} \left(\frac{\sum_{\forall z_e \in Z} d(z_a, z_e)}{|Z|} + \frac{\sum_{\forall x_i \in X_1} d(z_a, x_i)}{|X_1|} \right) \tag{11}$$

  And the third alternative of objective function, called *Furthest-First-Overall* or shortly as *FF-Overall*(Z, Z_Π, X_1), combines the two previous functions to gain the overall justification based on samples belong to both majority and minority classes. In particular, z_c is any $z_a \in Z_\Pi$ that miximize the following measurement.

$$\left(\frac{\sum_{\forall z_e \in Z} d(z_a, z_e)}{|Z|} + \frac{1}{2} \left(\frac{\sum_{\forall z_b \in Z_\Pi, z_b \neq z_a} d(z_a, z_b)}{|Z_\Pi| - 1} + \frac{\sum_{\forall x_i \in X_1} d(z_a, x_i)}{|X_1|} \right) \right) \tag{12}$$

Application to Training a Classifier. The set of sample representatives Z acquired from the last phase corresponds to the features space $V_0^* \in [0, 1]^{\rho \times d}$ of the majority class $Y_0^* = \{1\}^{\rho \times 1}$, i.e., $X_0^* = V_0^* \times Y_0^*$. The resulting training set is to aggregate this with the set of samples assigned to the minority class, or $X^* = X_1 \cup X_0^*$. Consequently, a classifier $CF_{X^*}^{\alpha}$ can be trained with the balanced data X^* using the choice of classification algorithm α. As such, the prediction $y_{\gamma_{a'}}$ of a connection instance whose features are altered by different obfuscation techniques is determined by the following definition.

$$y_{\gamma_{a'}} = CF_{X^*}^{\alpha}(f(\gamma_{a'})) \tag{13}$$

It is noteworthy that other methods to handle the class imbalance problem can also be used in this way to generate the data X^*. These will be included in the empirical study reported next.

Performance evaluation

Having defined the problem under examination and the proposed method in the past two sections, this paper continues with an empirical study in which the new undersampling framework is assessed against the results published in the original work and other methods to handle the problem of imbalance class. In particular, details of the dataset, experimental design, results and discussion are included in this section.

Investigated dataset and experimental design

The dataset under investigation is acquired from the original study of [30], where it is used for evaluating the robustness of machine-learning-based NIDS to obfuscated attacks. In order to obtain this data collection, connections representing both legitimate and intrusive attacks are created across vulnerable network services. Note that obfuscated intrusions are also generated by applying obfuscation techniques to some direct attacks, such that their appearances partly resemble the originals. Then, the TCP-level feature extractor called ASNM: Advanced Security Network Metrics [26] is deployed to generate the corresponding feature space. After normalizing value domains, the resulting dataset $X = V \times Y$ consists of a normalized feature space $V \in [0, 1]^{11,445 \times 194}$, where

a class of an instance $v_i \in V$ is drawn from the domain of 3 possible connection categories, i.e., $y_i \in Y$, $y_i \in$ {Legitimate, Direct attack, Obfuscated attack}. In Table 3, details of this dataset are presented with respect to number of class-specific samples that are generated using different network services. In addition, Table 4 illustrates ASNM feature categories and those 14 ones that have been selected in the report of [30] for classifying instances of legitimate connections and direct attacks (please consult [26] for a full list of 194 features). The current research focuses on this selection as not to bias a predictive model with features that are highly associated with obfuscated cases, thus reducing the feature space to $V \in [0, 1]^{11,445\times 14}$. Please consult the original data publication [27] for details of simulation and related settings.

For a thorough evaluation, a rich collection of compared methods are included in this empirical study, in addition to the proposed framework with three different objective functions: FF-Majority, FF-Minority and FF-Overall, respectively. The 'Baseline' of these new models is that initially introduced by [30], i.e., a classifier is developed from the original dataset X with the presence of imbalance class problem. Another basic competitor is the single clustering technique proposed by [46], which will be referred to as 'SingleClus' hereafter. Also, other undersampling algorithms are investigated, including

Table 3 Details of connections collected from different vulnerable network services

Network service	Legitimate	Direct attack	Obfuscated attack	Total
Apache Tomcat	809	61	163	1033
DistCC	100	12	23	135
MSSQL	532	31	103	666
PostgreSQL	737	13	45	795
Samba	4641	19	44	4704
Server	3339	26	100	3465
Other legitimate traffics	647	n/a	n/a	647
All services	10,805	162	478	11,445

Table 4 ASNM feature categories (with a full list available in the work of [26]) and those 14 features chosen by the study of [30]. Note that FFT denotes Fast Fourier Transformation

Category (total number: selected number) & *Feature name*; Description

Statistical (total 77: selected 4)

SigPktLenOut; standard deviation of outbound packet lengths

MeanPktLenIn; mean of packet sizes in inbound traffic of a connection

ConTcpFinCntIn; number of inbound packets of a connection with FIN flag set

ConTcpPshCntIn; number of inbound packets of a connection with PSH flag set

Localization (total 8: selected 0)

Distributed (total 34: selected 0)

Dynamic (total 32: selected 0)

Behavioral (total 43: selected 10)

CntOfOldFlows; no. of mutual flows between client/server hosts of analyzed connection 5 mins before it started

CntOfNewFlows; no. of mutual flows between client/server hosts of analyzed connection 5 mins after it finished

FourGonModulIn[1]; the module of 2nd coefficient of the FFT in goniometric representation

FourGonModulOut[1]; same as the previous one, but for outbound traffic

FourGonAngleN[9]; the angle of 10th coefficient of the FFT in goniometric representation

FourGonModulN[0]; same as the previous, but it represents the module of 1st coefficient of FFT

PolyInd3ordOut[3]; same as the previous, but it represents 4th coefficient of the approximation

GaussProds8Out[7]; same as the previous, but computed above outbound packets and represents a product of 8th slice of packets with Gaussian function that fits to interval of packets' slice

OutPktLen32s10i[3]; same as the previous, but computed above the first 32 secs of a connection.It is totaled outbound packet lengths of 4th interval

OutPktLen4s10i[2]; same as the previous, but computed above the first 4 secs of a connection.It is totaled outbound packet lengths of 3rd interval

RUS or Random UnderSampling [60] and its extensions to an ensemble approach. These include RUSBoost [60] and IRUS or Inverse Random UnderSampling [66], with their hyper-parameters being set according to the comparative study of [53]. Note that the target size of the majority class after an undersampling process is the number of samples belonging to the minority one. Besides those undersampling and ensemble models, this assessment also explores the oversampling counterpart that increase the cardinality of X_1 such that $|X_1| = |X_0|$. Specific to this work, the benchmark SMOTE technique [15] and its recent variant named 'Outlier-SMOTE' are employed, using the parameter setting recommended in the original report of [70]. This extension is picked up as it employs a similar intuition to FF-Majority, where samples that are largely different from others will be considered more important. Other settings are summarized as follows.

- For the proposed framework, the target size of multiple-clustering pool or M is set to 150, and each specific setting of these models is repeated for 30 trials to achieve a reliable conclusion from non-deterministic processes, based on averages across multiple runs.
- For the classification algorithm α, four techniques are included here. These include decision tree (C4.5) with the maximum depth of 15, Naive Bayes (NB) with the Gaussian kernel function, support vector machine (SVM) with the Radial kernel function, and Logistic Regression (LR), respectively. These are employed with the dataset generated by those filter and wrapper techniques to create a classifier.
- Since the current research focuses on robustness of machine-learning-based NIDS to obfuscated intrusions, a classifier is to be trained with instances representing legitimate connections and direct attacks only. Without the knowledge of those obfuscated instances, it is the aim of this study to see how well different methods recognize unseen threats. As such, the stratified 10-fold cross validation is firstly applied to the task of classifying legitimate connections and direct attacks, i.e., the examined data is the combination of instances belonging to these two classes only. The corresponding results illustrate the quality of classifiers to capture usual patterns. On top of that, the classifier trained in each fold will be used to predict 478 obfuscated intrusions, as either a legitimate or attack one. The latter experiment leads to the comparison of robustness as a classifier encounters truly new intrusive connections. At last, metrics used to assess predictive performance are TPR (True Positive Rate), FPR (False Positive Rate) and F1, respectively. Note that results with respect to Precision and Recall metrics are also provided in the Supplementary.[1] However, TPR is the only appropriate alternative for the second experiment, as all examined samples belong to the same class, i.e., obfuscated intrusion.

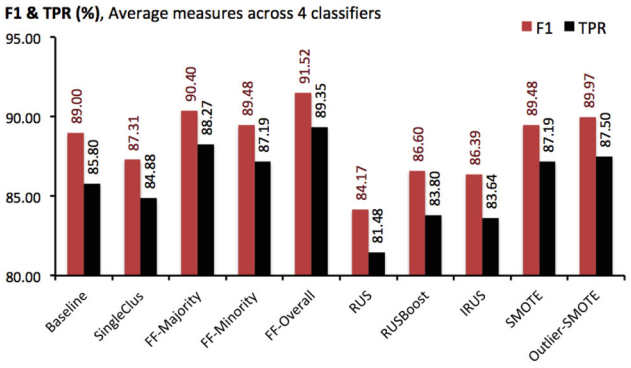

Fig. 4 Comparison of F1 and TPR scores obtained by examined methods, as averages across classification models and 30 trials of 10-fold cross validation

Experimental results

To start with, the overview of results obtained for the classification of legitimate and direct attack instances are provided in Figs. 4 and 5. In particular, the former presents F1 and TPR scores for different methods investigated herein. Note that these are averages across 30 trials of tenfold cross validation and four classification algorithms. This figure shows that the proposed methods of FF-Majority, FF-Minority and FF-Overall are able to improve the F1 metric of 89.00% achieved by the Baseline, with scores of 90.40%, 89.48% and 91.52%, respectively. Of course, these outperform the main competitor of SingleClus that receives a lower F1 measure of 87.31%. This suggests the more effective use of multiple clusterings for undersampling the majority class than the initial model that exploits only a single clustering. Nonetheless, SingleClus delivers more accurate classifiers than the random selection of representative samples, with RUS and its best ensemble extension getting the score of 84.17% and 86.60%. In addition to these compared methods that belong to the undersampling category, other oversampling alternatives of SMOTE and Outlier-SMOTE are also included in this experiment, where their F1 values are higher than that of the Baseline model and comparable to FF-Majority and FF-Minority. Only the FF-Overall technique performs better than these, where the highest score of 89.97% is obtained by Outlier-SMOTE. A similar trend can be observed with TPR scores acquired by those techniques, with the highest and

[1] https://drive.google.com/drive/folders/1V2wp7gReowaX3ZS6M8wGfA7uGnFjwSep?usp=sharing.

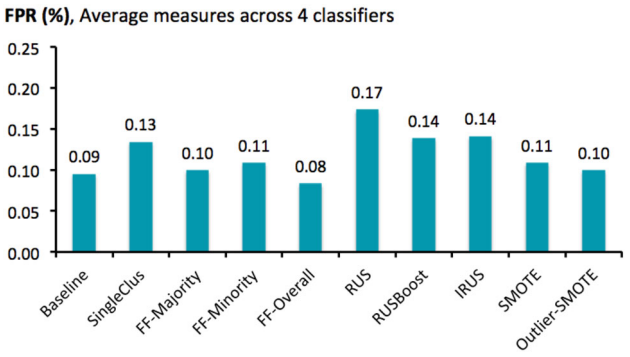

Fig. 5 Comparison of FPR scores obtained by examined methods, as averages across classification models and 30 trials of 10-fold cross validation

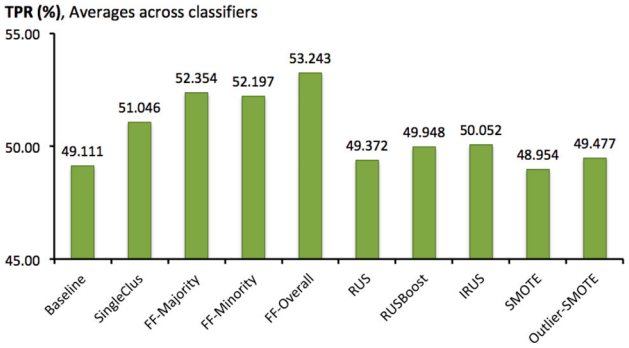

Fig. 6 Comparison of TPR scores obtained by different methods for classifying obfuscated instances. These are averages across classification models and 30 trials of tenfold cross validation

the lowest scores of 89.35% and 81.48% are reported with FF-Overall and RUS.

In Fig. 5, FPR measures of Baseline, proposed methods and those two oversampling models appear to be comparable, with the two lowest scores of 0.08% and 0.09% being seen with FF-Overall and Baseline, respectively. Despite this positive observation, oversampling algorithms may lead to overfitting, which can be witnessed later as a classification model is applied to unseen cases of obfuscated attacks. Similar to the previous results of F1 and TPR, RUS leads to the worst FPR score of 0.17%, where an improvement can be made by its extensions of RUSBoost and IRUS, i.e., lower measures of 0.14%. It is noteworthy that all three new methods of FF-Majority, FF-Minority and FF-Overall perform better than SingleClus that the score of 0.13%. Given the results depicted in both Figs. 4 and 5, the proposed framework can support a development of an accurate classifier, in addition to the application of feature subset selection employed by Baseline (further results with Precision and Recall metrics can be found in the Supplementary). However, the result discussed thus far is based on classify only legitimate connections and direct attacks, without the knowledge of obfuscated intrusions. Next, it is crucial to see predictive performance of these classification models to recognize obfuscated attacks, whose patterns are only partly represented in training samples of direct attacks.

As mentioned above, Fig. 6 provides the report of TPR scores each of which is acquired by one of investigated methods for classifying 478 obfuscated intrusions (as either legitimate or direct attack). These measures are presented as averages across classifiers and 30 runs of 10-fold cross validation. The purpose of this empirical study is to investigate predictive performance of different couplings of methods to handle the imbalance class problem and benchmark classification algorithms, with respect to the amount of unseen obfuscated attacks each of the resulting models can identify as direct attacks. Ideally, the target TPR score expected from those should be higher than 49.111% that is seen with Baseline. In particular, the clustering-based undersampling approach is apparently more effective than the random counterpart. This is concluded from the 50.0% score achieved by SingleClus, which is higher than those of RUS, RUSBoost and IRUS, i.e., 48.326%, 48.092% and 49.006%. Along this trend of improvement, all three new methods perform better than the single-clustering competitor, where average TPR measures summarized across different classifiers are 51.360%, 51.151% and 52.197% for FF-Majority, FF-Minority and FF-Overall, respectively. Note that the best result is somewhat 3.086% higher than the benchmark set by Baseline. Besides these reports, it is important to point out that oversampling techniques of SMOTE and Outlier-SMOTE become less effective to classify unseen samples than Baseline, with TPR scores of 47.908% and 48.431%. These are in par with the speculation regarding the overfitting effect caused by learning from an oversampled data set. Further details are presented in Table 5, in which TPR scores have been recorded for each coupling of classifiers and those examined techniques. According to this, the three variations of proposed undersampling framework usually exhibit predictive performance superior than Baseline, SingleClus and other compared methods. In particular, the best accurate alternative among these combinations is the exploitation of FF-Overall with NB classifier, which delivers an averaged TPR score of 85.146%. This tendency is similarly observed across four classification techniques examined herein. These results suggest that the robustness of classifiers to obfuscated intrusions can well be enhanced using the clustering based undersampling scheme, especially the multiple-clustering approach introduced in this work. This observation is due to other data-level methods included in this empirical study either add new samples of the minority class or reduce those of the majority class, using the general concept of nearest neighbors. Provided that a clear border exists between classes, its application would be rather effective. However, an inter-class overlap may dampen the quality of this local

Table 5 TPR scores as averages across from 30 trials of 10-fold cross validation, categorized by a combination of classifier and examined method. Note that corresponding values of standard deviation are given in (brackets)

Examined Method	NB	C4.5	SVM	LR
Baseline	81.172 (4.182)	36.402 (2.891)	15.690 (3.446)	63.180 (3.852)
SingleClus	82.636 (5.376)	38.494 (3.784)	17.782 (4.103)	65.272 (3.448)
FF-Majority	83.891 (3.842)	39.540 (3.019)	19.038 (3.211)	66.946 (2.562)
FF-Minority	84.100 (2.871)	39.540 (2.995)	18.619 (3.103)	66.527 (2.383)
FF-Overall	85.146 (3.004)	40.377 (2.981)	19.665 (3.164)	67.782 (2.410)
RUS	80.753 (5.122)	35.983 (4.721)	16.946 (5.673)	63.808 (6.714)
RUSBoost	81.172 (4.219)	36.402 (4.016)	17.573 (4.862)	64.644 (4.673)
IRUS	81.381 (5.134)	36.611 (3.673)	17.782 (4.523)	64.435 (4.381)
SMOTE	79.079 (4.873)	35.565 (4.822)	16.318 (3.749)	64.854 (4.006)
Outlier-SMOTE	79.498 (4.027)	36.192 (3.760)	16.946 (3.662)	65.272 (3.885)

approach, as compared to the clustering-oriented technique such as SingleClus. The same problem is also witnessed for the task of imputing missing values, where clustering information can be exploited to improve the accuracy of estimates of those missing ones [37,38]. Nonetheless, the use of a single clustering seen with SingleClus may overlook patterns exhibited in data under examination. Intuitively, multiple clusterings might fill in this gap well, which is observed here as well as within the line of research called ensemble clustering [12].

For the interpretation of experimental results thus far, averages across multiple trials are exploited for simplicity. This initial assessment follows the central limit theorem suggesting that the observed statistics in a controlled experiment may well be justified to the normal distribution. However, to obtain a more robust comparison between proposed models and other compared methods, the number of times that one is 'significantly better' and 'significantly worse' (of 95% confidence level) than the others are investigated next. Following the work of [33], let $\mu(i,t)$ be the average of TPR scores, across the t-th run of n-fold cross validation (n is 10 for the current research) for a model $i \in TC$ (TC consists of proposed and compared methods). Note that the TPR metric is the most appropriate with this second task of classifying a set of samples belonging to the same class of obfuscated attack. Formally, $\mu(i,t)$ can be defined as follows:

$$\mu(i,t) = \frac{1}{n}\sum_{\eta=1}^{n} TPR_{\eta}(i,t), \tag{14}$$

where $TPR_{\eta}(i,t)$ denotes the TPR score obtained from the η-th fold within the t-th run of method i. The comparison of means obtained from a single trial of cross validation may be misleading, as the difference between means may not be statistically significant at times. As such, it is more reliable to make a decision based on the 95% confidence interval for the mean $\mu(i,t)$. Such an interval is defined by the following.

$$\left[\mu(i,t) - 1.96\frac{Std(i,t)}{\sqrt{n}},\quad \mu(i,t) + 1.96\frac{Std(i,t)}{\sqrt{n}}\right], \tag{15}$$

where $Std(i,t)$ denotes the standard deviation of TPR measures across n-folds cross validation of the t-th trial, for a technique i. The statistical significance of the difference between any two methods $i, i' \in TC$ is found if there is no intersection between their confidence intervals of $\mu(i,t)$ and $\mu(i',t)$. In particular, a model i is significantly *better* than the other model i' when

$$\left(\mu(i,t) - 1.96\frac{Std(i,t)}{\sqrt{n}}\right) > \left(\mu(i',t) + 1.96\frac{Std(i',t)}{\sqrt{n}}\right) \tag{16}$$

Following that, the frequency that one method $i \in TC$ is significantly *better* than others across all experimented trials, i.e., $B(i)$, is calculated by the next equation.

$$B(i) = \sum_{\forall t=1\ldots30}\ \sum_{\forall i' \in TC, i' \neq i} better(i,i',t), \tag{17}$$

where

$$better(i,i',t) = \begin{cases} 1 \text{ if } \left(\mu(i,t) - 1.96\frac{Std(i,t)}{\sqrt{n}}\right) > \left(\mu(i',t) + 1.96\frac{Std(i',t)}{\sqrt{n}}\right) \\ 0 \text{ otherwise} \end{cases} \tag{18}$$

Likewise, the frequency that one technique $i \in TC$ is significantly *worse* than others, i.e., $W(i)$, is estimated as follows.

$$W(i) = \sum_{\forall t=1\ldots30}\ \sum_{\forall i' \in TC, i' \neq i} worse(i,i',t), \tag{19}$$

Table 6 Statistical assessment of TPR scores reported in Table 5 for the classification of unseen obfuscated instances. Note that, both better and worse metrics are reported for all combinations of compared methods and classification algorithms

Examined Method	NB better/worse	C4.5 better/worse	SVM better/worse	LR better/worse
Baseline	24/45	38/43	8/70	13/82
SingleClus	50/31	46/37	41/31	47/38
FF-Majority	73/23	70/30	55/18	74/28
FF-Minority	87/20	69/28	49/23	67/32
FF-Overall	101/12	93/19	80/14	90/13
RUS	20/54	13/64	21/52	20/69
RUSBoost	26/41	36/42	36/36	34/52
IRUS	30/37	38/40	39/29	31/58
SMOTE	8/96	5/69	16/58	36/49
Outlier-SMOTE	16/76	19/55	32/46	49/40

where

$$worse_j(i, i', t) = \begin{cases} 1 \text{ if } \left(\mu(i,t) + 1.96\frac{Std(i,t)}{\sqrt{n}}\right) < \left(\mu(i',t) - 1.96\frac{Std(i',t)}{\sqrt{n}}\right) \\ 0 \text{ otherwise} \end{cases} \tag{20}$$

Given this assessment framework, the results reported in Table 5 is further analyzed, with the corresponding statistics being shown in Table 6. These suggest that the three variations of proposed method appear to be more accurate than other compared methods. In particular, they possess higher 'better' and lower 'worse' measures than the baseline and the state-of-the-art oversampling model of Outlier-SMOTE. Hence, the new framework can be a competitive alternative to previous works that focus on handling the class imbalance problem through data sampling and feature engineering. In addition to the evaluation against Outlier-SMOTE, two other recent methods identified earlier in the related work are also investigated, i.e., Recursive feature elimination [62] and Adaptive ensemble learning [23]. The next comparison shown in Fig. 7 depicts TPR scores of the three best models reported in Table 5, against those of the abovementioned state-of-the-art techniques. Based on this illustration, the predictive performance of proposed models are comparable to those of both Recursive feature elimination and Adaptive ensemble learning, thus putting the new method as another benchmark for developing a classifier to recognize unseen and mutated intrusive attacks.

Discussion and implication of proposed framework

In addition to the comparative assessment presented previously, this section continues with a discussion regarding parameter analysis specific to the proposed framework and its implication as a general method for handling the class imbalance problem. At first, there are two algorithmic variables that may influence predictive performance of all three variations of FF-Majority, FF-Minority and FF-Overall. One is the size of clustering results M, which determines the number of centroids in the pool and has been initially set to 150 for the results reported thus far. To disclose the association between this particular parameter and the accuracy of classifying those obfuscated intrusions, the experiment explained above is repeated for different values of $M \in \{100, 150, 200, 250, 300, 350\}$. Provided that Fig. 8A summarizes the corresponding results across four different classifiers, a bigger M usually leads to improved TPR scores witnessed with all three new techniques. In particular to FF-Overall, the highest measures around 54% is obtained as the ensemble size grows above 250. In other words, adding more clustering results to the pool will not yield any further significant improvement. This is similarly seen with the other two models, where FF-Majority performs slightly better FF-Minority across different values of M. Besides, the

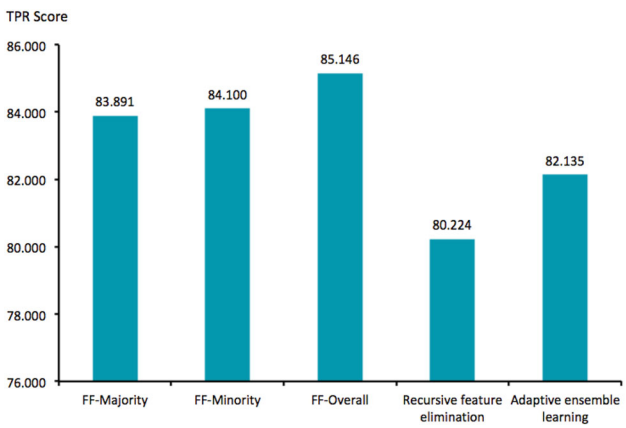

Fig. 7 TPR scores obtained by FF-Majority, FF-Minority and FF-Overall for the classification of unseen obfuscated instances using the NB model. These are compared with those achieved by Recursive feature elimination [62] and Adaptive ensemble learning [23], where parameter settings are based on original studies

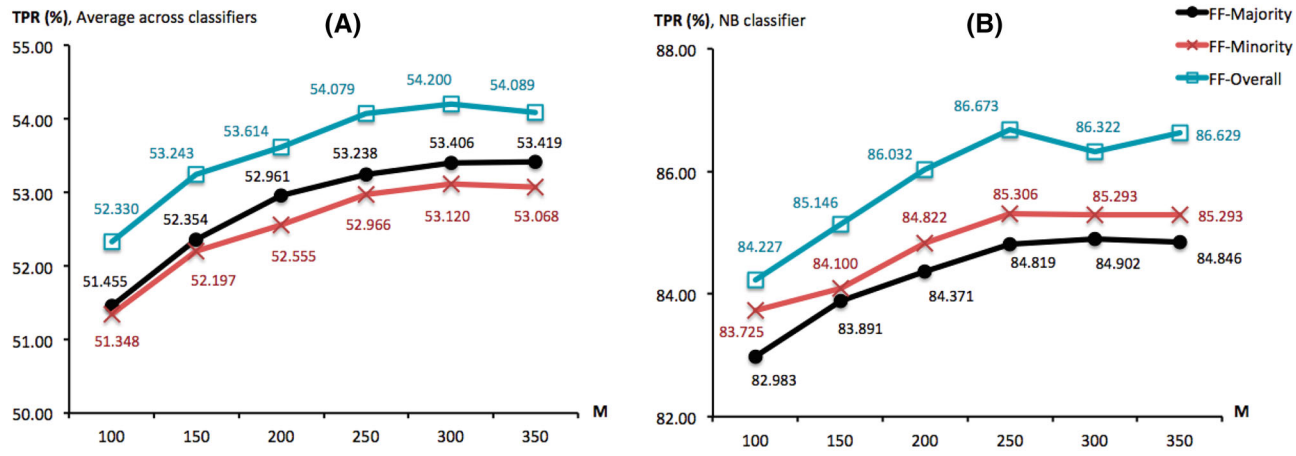

Fig. 8 TPR scores obtained by FF-Majority, FF-Minority and FF-Overall, with different ensemble sizes of $M \in \{100, 150, 200, 250, 300, 350\}$. These are summarized from 30 trials of 10-fold cross validation: **A** across four classifiers and **B** specific to NB

same comparison specific to the NB classifier is illustrated in Fig. 8B, where the score of FF-Overall increases from 85.146 to 86.673% by enlarging the ensemble size from 150 to 250. Alike trends are also observed for both FF-Majority and FF-Minority. Despite these, a tradeoff between gain in classification performance and a higher complexity is a real concern for resource-constrained applications. It is noteworthy that the complexity of generating a single base clustering is approximated to $O(n)$ and $O(nM)$ for an ensemble of M members. In fact, the overall complexity of a new method is the combination of $O(nkM)$ and $O(\rho^2)$ for the generation of multiple clusterings and the selection of representatives, where ρ denotes the final size of the majority class and k is the averaged number of clusters in a clustering result. Since ρ tends to be much smaller than n, this complexity may usually converges to $O(n)$. However, it is more expensive than (to the a factor of M) the baseline of clustering whose complexity is commonly $O(nk)$. It is noteworthy that the complexity analyzed herein focuses on the data preprocessing phase as a prior to training a classification model. As such, a prediction time would not be affected by the proposed procedure but depending on a choice of algorithm used to develop the target classifier. The resulting module can be embedded in a software solution to perform an automated NIDS. A timely response may be expected from such a setting; however, its efficiency is up to hardware support. Of course, to gain a rigorous body of knowledge, matching the new undersampling framework to big training data may well require a great deal of time. In such a case, the exploitation of distributed computing might be sensible.

As mentioned above, another important parameter to be studied is the size ρ of reduced majority class X_0, where it is set initially as the same as that of the minority counterpart X_1. With the process of tenfold cross validation, it is automatically configured to be around 146 samples that will be referred to as Q hereafter. The next investigation is to discover the relation between overall as well as classifier-specific TPR scores and different values of Q: $Q, 2Q$ and $3Q$ that correspond to target numbers of representative samples of the majority class being 146, 292 and 438, respectively. Based on the experiment of classifying obfuscated connections using $M = 250$, the results summarized from all four classifiers are depicted in Fig. 9A, where the case of $2Q$ generally delivers a more accurate classification model. For instance, TPR scores of FF-Overall inclines from 54.079% to 54.813%, as the size of reduced majority class grows twice larger from Q to $2Q$. This suggests a loss of majority-class information encountered along with the procedure of selecting a small number of representative samples. Nonetheless, having too many of these in the case of $3Q$ may not be as effective since the cause of imbalance problem has been re-visited. A similar tendency can be seen in Fig. 9B that illustrates the result specific to NB. In particular, FF-Overall a higher TPR of 87.246% with $2Q$, compared to 86.673% with the initial setting of Q.

The issue to be clarified next regards the implication of proposed methods to an extreme case of class imbalance, in which the size of minority class gets extremely small. For this experiment, the whole training data collection, i.e., sizes of majority and minority classes are 10.805 and 162, is exploited to determine classes of those unseen obfuscated instance. Note that the experiment is repeated from 30 trials with different sizes of $|X_1| \in \{162, 122, 81, 41\}$, while M is 250 and the target size of majority class is $2Q$, i.e., twice that of the other class. According to Fig. 10A that shows TPR scores obtained by the NB classifier, the tendency of predictive performance constantly declines as $|X_1|$ drops from 162 to 41. This is commonly observed with different methods examined here, with

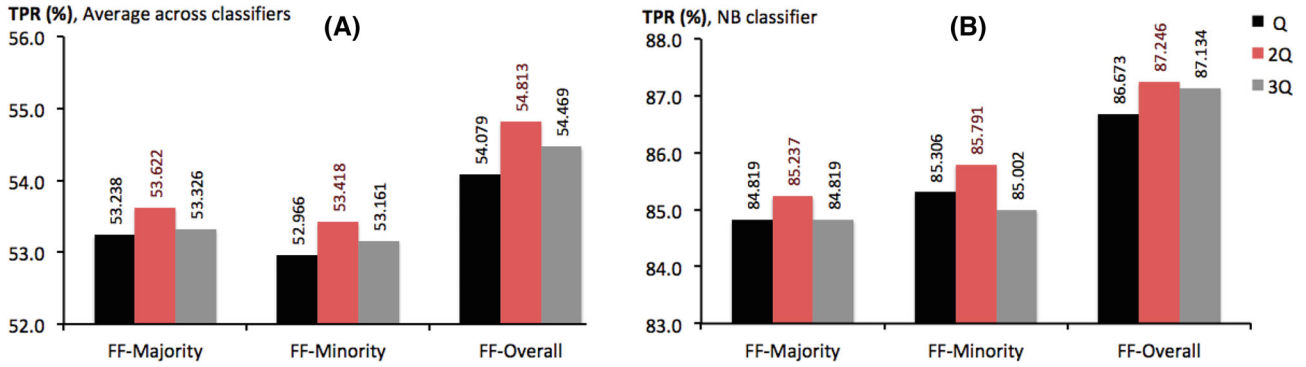

Fig. 9 TPR scores obtained by FF-Majority, FF-Minority and FF-Overall, with different sizes of the majority class, Q, $2Q$ and $3Q$. These are summarized from 30 trials of tenfold cross validation: **A** across four classifiers and **B** specific to NB

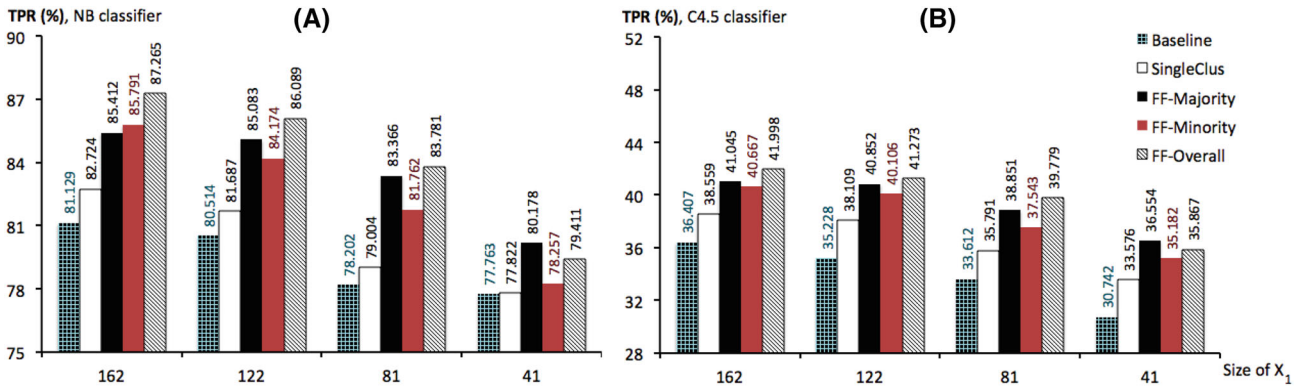

Fig. 10 TPR scores obtained by FF-Majority, FF-Minority and FF-Overall, with different sizes of the minority class $X_1 \in \{162, 122, 81, 41\}$. These are summarized from 30 trials for two specific classifiers: **A** NB and **B** C4.5

FF-Majority, FF-Minority and FF-Overall providing the best set of measures and outperforming both Baseline and SingleClus. This is also observed with the C4.5 technique whose results are given in Fig. 10B. With these, it is possible to infer that the proposed framework can be a robust alternative to develop a classification model for a highly imbalance problem found in many real-world domains, e.g., fraud detection and cancer diagnosis. Furthermore, the application of FF-Minority tends to be less effective as $|X_1|$ decreases, which is sensible given the fact that the underlying objective function makes use of distances from a centroid of interest to members of X_1. As such, the performance of FF-Overall is also affected by this estimation, where its TPR scores become lower than those of FF-Majority as $|X_1|$ getting to the smallest size of 41. For such a case, FF-Majority is preferred to the others, but it should be noted that accuracy of the resulting model will still be largely lower than that achieved with a bigger $|X_1|$. To this end, it may be better to collect additional samples of the minority classes.

Conclusion

This paper has presented a new approach to handle the problem of imbalance class for machine-learning-based NIDS with adversarial attacks, where legitimate, direct attacks and obfuscated intrusions can be more accurately classified. As a compliment to an initial attempt of seeking a feature subset that is optimal for this classification task, the current work focuses on solving the aforementioned difficulty by extending the concept of clustering based undersampling. Instead of making a reference out of a single data partition, the target set of representative centroids is selected from a pool of multiple clustering results, which has been generated using the concept of ensemble clustering. This ensures the diversity of data partitions, hence the resulting centroids, from which some can well be chosen to represent the original set of majority class. Three variations of the proposed framework are introduced based on different objective functions used for

this iterative and greedy process: FF-Majority, FF-Minority and FF-Overall.

Based on the evaluation with four different classifiers and the published dataset, those new methods usually outperform the baseline (i.e., the use of selected feature subset without handling the class imbalance problem) and the single-clustering-based undersampling. They also demonstrate the predictive performance that is in par with state-of-the-art methods included in this study. In fact, the proposed framework is generalized such that it can be coupled with any common classification model. Likewise, this applies to the exploitation of other objective functions to differentiate the goodness of a centroid as a representative of the majority class. However, the proposed models may be less effective if a more sophisticated obfuscation technique is exploited to mutate known patterns. Yet, they are more expensive than their baseline, which may not be appropriate for analyzing big data. Enriching a training set with modified instances [77] using new obfuscation methods can partly resolve the former, while applying the concept of federated learning may scale the application of proposed models up to a big security data collection [5]. Besides, another future work is to explore optimization techniques found in the literature [56] that may lead to a better selection of centroids from a pool of multiple clusterings. This is similar to the attempt to improve a greedy optimization to discretization of feature domains [64]. Finally, an introduction of fuzzy sets and vocabularies is able to support the explainability of prediction process [21]. This final remark draws a great deal of attention provided the emerging trend of explainable AI for modern applications.

Acknowledgements This research work is partly supported by Mae Fah Luang University, Newton IAPP 2017 (Royal Academy of Engineering and Thailand Research Fund), and Newton Institutional Links 2020-21 project (British Council and National Research Council of Thailand).

Declarations

Conflict of interest The authors declare no conflict of interest.

CRediT author statement Chutipon Pimsarn: Data curation, Visualization, Investigation, Software, Writing - Original draft preparation Tossapon Boongoen: Conceptualization, Methodology, Validation, Writing - Original draft preparation, Supervision Natthakan Iam-On: Methodology, Validation, Writing - Reviewing and Editing, Supervision Nitin Naik: Writing - Reviewing and Editing Longzhi Yang: Writing - Reviewing and Editing

References

1. Abdulhammed R, Faezipour M, Abuzneid A, Abumallouh A (2019) Deep and machine learning approaches for anomaly-based intrusion detection of imbalanced network traffic. IEEE Sens Lett 3(1):1–4
2. Agarwal N, Hussain SZ (2018) A closer look at intrusion detection system for web applications. Secur Commun Netw 2018(9601357):1–27
3. Ahmim A, Derdour M, Ferrag M (2018) An intrusion detection system based on combining probability predictions of a tree of classifiers. Int J Commun Syst 31(9):e3547
4. Akashdeep, Manzoor I, Kumar N (2017) A feature reduced intrusion detection system using ANN classifier. Expert Syst Appl 88:249–257
5. Alazab M, MSPR, MP, Reddy P, Gadekallu TR, Pham QV (2022) Federated learning for cybersecurity: Concepts, challenges and future directions. IEEE Trans Ind Inf. https://doi.org/10.1109/TII.2021.3119038
6. Alcaraz C (2018) Cloud-assisted dynamic resilience for cyber-physical control systems. IEEE Wirel Commun 25(1):76–82
7. Aljanabi M, Ismail MA, Ali AH (2021) Intrusion detection systems, issues, challenges, and needs. Int J Comput Intell Syst 14(1):560–571
8. Anthi E, Williams L, Rhode M, Burnap P, Wedgbury A (2021) Adversarial attacks on machine learning cybersecurity defences in industrial control systems. J Inf Secur Appl 58(102717):1–9
9. Ashibani Y, Mahmoud QH (2017) Cyber physical systems security: Analysis, challenges and solutions. Computer Security 8:81–97
10. Barreno M, Nelson B, Joseph A, Tygar J (2010) The security of machine learning. Mach Learn 81(2):121–148
11. Blaszczynski J, Stefanowski J (2015) Neighborhood sampling in bagging for imbalanced data. Neurocomputing 150:529–542
12. Boongoen T, Iam-On N (2022) Using link-based consensus clustering for mixed-type data analysis. CMC 70(1):1993–2011
13. Boongoen T, Shang C, Iam-On N, Shen Q (2011) Extending data reliability measure to a filter approach for soft subspace clustering. IEEE Transactions on Systems, Man and Cybernetics, Part B 41(6):1705–1714
14. Chandra A, Khatri SK, Simon R (2019) Filter-based attribute selection approach for intrusion detection using k-means clustering and sequential minimal optimization technique. In: Proceedings of Amity International Conference on Artificial Intelligence, pp. 740–745
15. Chawla N, Bowyer K, Hall L, Kegelmeyer W (2002) SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16:321–357
16. Corona I, Giacinto G, Roli F (2013) Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues. Inf Sci 239:201–225
17. Dka C, Papa J, Lisboa C, Munoz R, Dvhc A (2019) Internet of things: a survey on machine learning-based intrusion detection approaches. Comput Networks 151:147–157
18. Farhan BI, Jasim AD (2022) A survey of intrusion detection using deep learning in internet of things. Iraqi Journal For Computer Science and Mathematics 3(1):83–93
19. Farnaaz N, Jabbar MA (2016) Random forest modeling for network intrusion detection system. Procedia Computer Science 89:213–217

20. Fossaceca JM, Mazzuchi TA, Sarkani S (2015) MARK-ELM: application of a novel multiple kernel learning framework for improving the robustness of network intrusion detection. Expert Syst Appl 42:4062–4080
21. Fu X, Boongoen T, Shen Q (2010) Evidence directed generation of plausible crime scenarios with identity resolution. Appl Artif Intell 24(4):253–276
22. Gao L, Shen W, Li X (2019) New trends in intelligent manufacturing. Engineering 5(4):11–20
23. Gao X, Shan C, Hu C, Niu Z, Liu Z (2019) An adaptive ensemble machine learning model for intrusion detection. IEEE Access 7:82512–82521
24. Guo C, Ping Y, Liu N, Luo S (2016) A two-level hybrid approach for intrusion detection. Neurocomputing 214:391–400
25. Haseeb K, Almogren A, Islam N, Ud-Din I, Jan Z (2019) An energy-efficient and secure routing protocol for intrusion avoidance in iot-based wsn. Energies 12(21):4174
26. Homoliak I, Barabas M, Chmelar P, Drozd M, Hanacek P (2013) ASNM: Advanced security network metrics for attack vector description. In: Conference on Security and Management, pp. 350–358
27. Homoliak I, Malinka K, Hanacek P (2020) ASNM Datasets: A collection of network attacks for testing of adversarial classifiers and intrusion detectors. IEEE Access 8:112427–112453
28. Homoliak I, Ovsonka D, Gregr M, Hanacek P (2014) NBA of obfuscated network vulnerabilities exploitation hidden into HTTPS traffic. In: International Conference for Internet Technology and Secured Transactions, pp. 311–318
29. Homoliak I, Teknos M, Barabas M, Hanacek P (2016) Exploitation of netem utility for non-payload-based obfuscation techniques improving network anomaly detection. In: International Conference on Security and Privacy in Communication Systems, pp. 770–773
30. Homoliak I, Teknos M, Ochoa M, Breitenbacher D, Hosseini S, Hanacek P (2018) Improving network intrusion detection classifiers by non-payload-based exploit-independent obfuscations: An adversarial approach. EAI Endorsed Transactions on Security and Safety 5(17):e4
31. Iam-On N (2020) Clustering data with the presence of attribute noise: a study of noise completely at random and ensemble of multiple k-means clusterings. Int J Mach Learn Cybern 11(3):491–509
32. Iam-On N, Boongoen T (2015) Diversity-driven generation of link-based cluster ensemble and application to data classification. Expert Syst Appl 42(21):8259–8273
33. Iam-On N, Boongoen T, Garrett S, Price C (2011) A link-based approach to the cluster ensemble problem. IEEE Trans Pattern Anal Mach Intell 33(12):2396–2409
34. Iam-On N, Boongoen T (2017) Improved student dropout prediction in thai university using ensemble of mixed-type data clusterings. Int J Mach Learn Cybern 8(2):497–510
35. Jia Y, Qi Y, Shang H, Jiang R, Li A (2018) A practical approach to constructing a knowledge graph for cybersecurity. Engineering 4(1):53–60
36. Karatas G, Demir O, Sahingoz O (2020) Increasing the performance of machine learning-based idss on an imbalanced and up-to-date dataset. IEEE Access 8:32150–32162
37. Keerin P, Boongoen T (2022) Estimation of missing values in astronomical survey data: An improved local approach using cluster directed neighbor selection. Inf Process Manage 59(2):102881
38. Keerin P, Boongoen T (2022) Improved knn imputation for missing values in gene expression data. CMC-Computers, Materials and Continua 70(2):4009–4025
39. Kravchik M, Shabtai A (2018) Detecting cyber attacks in industrial control systems using convolutional neural networks. In: ACM International Workshop on Cyber-Physical Systems Security and Privacy, pp. 72–83
40. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5:231–232
41. Kumar DP, Amgoth T, Annavarapu CSR (2019) Machine learning algorithms for wireless sensor networks: A survey. Information Fusion 49:1–25
42. Kumar DP, Amgoth T, Annavarapu CSR (2021) CANintelliIDS: Detecting In-Vehicle Intrusion Attacks on a Controller Area Network Using CNN and Attention-Based GRU. IEEE Transactions on Network Science and Engineering 8(2):1456–1466
43. Lee YJ, Yeh YR, Wang YCF (2013) Anomaly detection via online oversampling principal component analysis. IEEE Trans Knowl Data Eng 25(7):1460–1470
44. Li J, Qu Y, Chao F, Shum H, Ho E, Yang L (2019) Machine learning algorithms for network intrusion detection. In: AI in Cybersecurity, pp. 151–179. NY: Springer
45. Lin CT et al (2018) Minority oversampling in kernel adaptive subspaces for class imbalanced datasets. IEEE Trans Knowl Data Eng 30(5):950–962
46. Lin WC, Tsai CF, Hu YH, Jhang JS (2017) Clustering-based undersampling in class-imbalanced data. Inf Sci 409–410:17–26
47. Ma W (2020) Analysis of anomaly detection method for internet of things based on deep learning. Transactions on Emerging Telecommunications Technologies p. e3893
48. Mazini M, Shirazi B, Mahdavi I (2019) Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and adaboost algorithms. Journal of King Saud University - Computer and Information Sciences 31(4):541–553
49. Molina-Coronado B, Mori U, Mendiburu A, Miguel-Alonso J (2020) Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. IEEE Trans Netw Serv Manage 17(4):2451–2479
50. Naik N, Jenkins P, Savage N, Yang L, Boongoen T, Iam-On N (2021) Fuzzy-import hashing: A static analysis technique for malware detection. Forensic Science International: Digital Investigation 37:301139
51. Naik N, Jenkins P, Savage N, Yang L, Boongoen T, Iam-On N, Naik K, Song J (2021) Embedded YARA rules: strengthening YARA rules utilising fuzzy hashing and fuzzy rules for malware analysis. Complex and Intelligent Systems 7:687–702
52. Najafabadi M, Villanustre F, Khoshgoftaar T, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. Journal of Big Data 2(1):1
53. Nanni L, Fantozzi C, Lazzarini N (2015) Coupling different methods for overcoming the class imbalance problem. Neurocomputing 158:48–61
54. Panwong P, Boongoen T, Iam-On N (2020) Improving consensus clustering with noise-induced ensemble generation. Expert Syst Appl 146:113–138
55. Parsaei MR, Rostami SM, Javidan R (2016) A hybrid data mining approach for intrusion detection on imbalanced nsl-kdd dataset. Int J Adv Comput Sci Appl 7(6):20–25
56. Pervaiz S, Ul-Qayyum Z, Bangyal WH, Gao L, Ahmad J (2021) A systematic literature review on particle swarm optimization techniques for medical diseases detection. Comput Math Methods Med 2021(5990999):1–10
57. Pierazzi F, Cristalli S, Bruschi D, Colajanni M, Marchetti M, Lanzi GA (2020) Glyph: Efficient ML-based detection of heap spraying attacks. IEEE Trans Inf Forensics Secur 16:740–755
58. Prasad R, Rohokale V (2020) Artificial intelligence and machine learning in cyber security. In: Cyber Security: The Lifeline of Information and Communication Technology, pp. 231–247. NY: Springer

59. Rubin S, Jha S, Miller B (2004) Automatic generation and analysis of NIDS attacks. In: Annual Computer Security Applications Conference, pp. 28–38
60. Seiffert C, Khoshgoftaar T, Hulse JV, Napolitano A (2010) Rusboost: a hybrid approach to alleviating class imbalance. IEEE Transactions on System, Man and Cybernertics, Part A 40(1):185–197
61. Sethi TS, Kantardzic M (2018) When good machine learning leads to bad security. Ubiquity May(1):1–14
62. Sharma NV, Yadav NS (2021) An optimal intrusion detection system using recursive feature elimination and ensemble of classifiers. Microprocess Microsyst 85:104293
63. Shen Y, Zheng K, Wu C, Zhang M, Niu X, Yang Y (2018) An ensemble method based on selection using bat algorithm for intrusion detection. Comput J 61(4):526–538
64. Sriwanna K, Boongoen T, Iam-On N (2017) Graph clustering-based discretization of splitting and merging methods (graphs and graphm). HCIS 7(1):1–39
65. Sun Z, Song Q, Zhu X, Sun H, Xu B, Zhou Y (2015) A novel ensemble method for classifying imbalanced data. Pattern Recogn 48:1623–1637
66. Tahir M, Kittler J, Yan F (2012) Inverse random under sampling for class imbalance problem and its application to multi-label classification. Pattern Recogn 45(10):3738–3750
67. Tarter A (2017) Importance of cyber security. In: Community Policing-A European Perspective: Strategies, Best Practices and Guidelines, pp. 213–230. NY: Springer
68. Teixeira MA, Salman T, Zolanvari M, Jain R, Meskin N, Samaka M (2018) SCADA system testbed for cybersecurity research using machine learning approach. Future Internet 10(8):76
69. Tesfahunand A, Bhaskari DL (2013) Intrusion detection using random forests classifier with SMOTE and feature reduction. In: Proceedings of International Conference on Cloud Ubiquitous Computing and Emerging Technology, pp. 127–132
70. Turlapati VPK, Prusty MR (2020) Outlier-SMOTE: A refined oversampling technique for improved detection of COVID-19. Intelligence-Based Medicine 3–4:100023
71. Uddin M, Rahman A, Uddin N, Memon J, Alsaqour R, Kazi S (2013) Signature-based multi-layer distributed intrusion detection system using mobile agents. International Journal of Network Security 15(2):97–105
72. Vigna G, Robertson W, Balzarotti D (2004) Testing network-based intrusion detection signatures using mutant exploits. In: ACM conference on Computer and Communications Security, pp. 21–30
73. Wang D, abd Y, Zhang XW, Jin L (2019) Detection of power grid disturbances and cyber-attacks based on machine learning. Journal of Information Security and Applications 46:42–52
74. Watson D, Smart M, Malan G, Jahanian F (2004) Protocol scrubbing: Network security through transparent flow modification. IEEE/ACM Trans Networking 12(2):261–273
75. Yan B, Han G (2018) Effective feature extraction via stacked sparse autoencoder to improve intrusion detection system. IEEE 6:41238–41248
76. Yao H, Fu D, Zhang P, Li M, Liu Y (2018) MSML: a novel multilevel semi-supervised machine learning framework for intrusion detection system. IEEE Internet Things J 6(2):1949–1959
77. Yin C, Zhu Y, Liu S, Fei J, Zhang H (2020) Enhancing network intrusion detection classifiers using supervised adversarial training. Journal of Supercomputing 76:6690–6719
78. You I, Yim K (2010) Malware obfuscation techniques: A brief survey. In: International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 297–300