electronics

Article

EmSM: Ensemble Mixed Sampling Method for Classifying Imbalanced Intrusion Detection Data

Ilok Jung [1,*], Jaewon Ji [2] and Changseob Cho [2]

[1] Graduate School of Information Security, Korea University, Seoul 02841, Korea
[2] Cyber Security Research Laboratory, IGLOOSECURITY, Seoul 05836, Korea; jaewon.ji@igloosec.com (J.J.); changseob.cho@igloosec.com (C.C.)
* Correspondence: okkida@korea.ac.kr

Abstract: Research on the application of machine learning to the field of intrusion detection is attracting great interest. However, depending on the application, it is difficult to collect the data needed for training and testing, as the least frequent data type reflects the most serious threats, resulting in imbalanced data, which leads to overfitting and hinders precise classification. To solve this problem, in this study, we propose a mixed resampling method using a hybrid synthetic minority oversampling technique with an edited neural network that increases the minority class and removes noisy data to generate a balanced dataset. A bagging ensemble algorithm is then used to optimize the model with the new data. We performed verification using two public intrusion detection datasets: PKDD2007 (balanced) and CSIC2012 (imbalanced). The proposed technique yields improved performance over state-of-the-art techniques. Furthermore, the proposed technique enables improved true positive identification and classification of serious threats that rarely occur, representing a major functional innovation.

Keywords: imbalanced data; intrusion detection; machine learning; sampling method

Citation: Jung, I.; Ji, J.; Cho, C. EmSM: Ensemble Mixed Sampling Method for Classifying Imbalanced Intrusion Detection Data. *Electronics* **2022**, *11*, 1346. https://doi.org/10.3390/electronics11091346

Academic Editor: Vijayakumar Varadarajan

Received: 30 March 2022
Accepted: 21 April 2022
Published: 23 April 2022

1. Introduction

Rapid advancements in information technology have resulted in more sophisticated and frequent cyberattacks. Hence, traditional intrusion detection security measures have reached their limits in terms of detecting, analyzing, and responding to threats [1,2]. To address this problem, machine learning techniques have been used in several areas to improve cyber security [3]. Based on current advancements, supervised machine learning techniques that learn and classify network behaviors have achieved higher true-positive and lower false-positive rates than existing signature-based approaches [4,5].

When considering machine learning intrusion detection approaches, there are two important considerations regarding the available labeled datasets. First, normal non-threatening cyber activities outweigh malicious behaviors 100,000:1 (see Table 1), which overall presents heavily imbalanced testing and training scenarios. Second, several types of cyberattacks are known to exist, and the most serious threats are naturally the least frequent. Hence, when limiting the set of cyber activities only to intrusions, the data remains quite imbalanced, limiting the ability of machine learning algorithms to provide true positive intrusion classifications [5,6].

Table 1. Imbalance ratios of intrusion detection datasets.

Dataset	Imbalance Ratio
DARPA/KDD Cup 99 [7]	36,725
CICIDS2017 [8]	112,287
CSIC2012 [9]	1160
NSL-KDD [10]	648
PKDD2007 [11]	18

As is evident from Table 1, with respect to the imbalance ratios (IRs) of KDD Cup 99 [7], CICIDS2017 [8], and CSIC2012 [9], there is a vast gap among the data classes which also affects the efficiency of the machine learning (ML) [5]. In contrast, NSL-KDD [10] and PKDD2007 [11], which were built to minimize imbalanced and redundant data types, have smaller ratios. The IR reflects the weight of the majority class as compared with the minority class. In Equation (1), $max\_i\{C\_i\}$ and $min\_i\{C\_i\}$ denote the majority and minority classes, respectively. The larger the IR, the less reliable the dataset is for machine learning intrusion detection [12].

$$IR = max\_i\{C\_i\}/min\_i\{C\_i\} \quad (1)$$

In previous studies, we applied resampling techniques to balance the training data, developed or adapted machine learning algorithms (e.g., cost-sensitive learning technique), and employed ensemble approaches [13]. This routine of incremental improvement is commonplace in machine learning fields, especially those dealing with imbalanced data types. With respect to cyber security data, several studies have applied the synthetic minority oversampling technique (SMOTE, Ma and Shi) [14]. Others have adapted traditional methods (e.g., random forest (RF), multilayer perceptron (MLP), and support vector machine (SVM)), deep learning methods (i.e., convolutional neural network (CNN) and deep neural network (DNN) Tripathi and Makwana) [15], and boosting techniques for all of these. Nevertheless, all remain inadequate to ensure high classification accuracy for minority cyberattack data types, and no suitable indicators are available to assess the imbalanced datasets. Notably, the edited nearest neighbor (ENN) approach finds the K nearest neighbors (KNNs) of each data type and checks whether the majority class from all neighbors is the same as the observation class, which effectively cleans the database by removing samples close to the decision boundary [6].

Accordingly, in this study, we propose a new ensemble mixed sampling method (EmSM) that combines resampling with ensemble learning. First, SMOTE is combined with an ENN to produce a superior intrusion detection dataset. Second, a bagging ensemble model is applied to resolve the imbalanced dataset by generating multiple models of the same size and performing predictions to improve the accuracy for the minority class while maintaining the accuracy for the majority class.

This is accomplished by considering two factors. First, the PKDD2007 and CSIC2012 datasets are leveraged as they include basic cyberattack payloads limited to web-service intrusions. Notably, they present various classes per field, which is good for calculating IR. Second, binary and multi-classification sampling techniques from the Python Toolbox [12] are leveraged to compare our results to extant sampling techniques for a variety of label formats.

We make several contributions with this study:

- A hybrid resampling and bagging ensemble technique that improves the prediction accuracy of the minority class while maintaining that of the majority class (Section 3);
- A method for measuring improvements to the machine learning performance on imbalanced datasets using binary and multi-classification (Section 4);
- A first-of-its-kind imbalanced data handling method that accurately identifies rare types of intrusion detection (Section 4);
- Identification of the effects of the data distribution of extracted features, applied algorithm, and class ratio on the efficiency of handling imbalanced datasets (Section 4).

The remainder of this paper is organized as follows: in Section 2, we discuss previous work related to this research; in Section 3, we describe the proposed ensemble mixed sampling method; in Section 4, we present our experiments and the results; and in Section 5, we present the conclusions.

2. Related Work

This section provides the background and justification for this study. First, we summarize the current knowledge on intrusion detection data, then, we cover related work in which sampling techniques were examined for imbalanced class handling.

2.1. Intrusion Detection Data

Over time, system administrators using dedicated hardware and software devices have accumulated many examples of anomalous patterns, transactions, policy changes, and user behaviors indicative of network- and web-based intrusions. Notably, it takes a long time to accumulate sufficient examples to construct a sizable database, as most network traffic is mundane and harmless. Hence, only a minority of network traffic is considered malicious. This imbalanced data frequency makes it difficult to effectively train a machine learning application to recognize serious threats. Usually, data dedicated to network packet analysis and protection are difficult to manage and are rarely made public owing to confidentiality issues. However, several datasets are publicly available for benchmarking and scholarly pursuits. For example, the Defense Advanced Research Projects Agency (DARPA) continues to release up-to-date Knowledge Discovery in Databases (KDD) datasets [7], and the Network Security Laboratory's (NSL) KDD version [10] is widely used. Table 2 provides a brief comparison of the most well-known intrusion detection datasets.

Table 2. Comparison of intrusion detection datasets.

Dataset	Public	HTTP	Labeled	Payload	Class	Metadata	Year	Balance	IR *
DARPA/KDD Cup 99 [7]	O	-	O	X	4	X	1998	X	36,725
NSL-KDD [10]	O	-	O	X	4	X	2009	X	648
CICIDS2017 [8]	O	△	O	△	6	O	2017	X	112,287
PKDD2007 [11]	O	O	O	O	8	O	2007	O	18
CSIC2010 [9]	O	O	O	O	2	O	2010	X	1160
CSIC2012 [9]	O	O	O	O	10	O	2012	X	1160

* IR, imbalance ratio.

The utilization of these datasets requires an understanding of their purposes and the environments in which they were built, as clarified in Table 2. It is necessary to determine which part of each dataset was created in a real environment (Table 2, public) and which was generated arbitrarily and then refined (Table 2, metadata). Moreover, it is important to know whether the labeled classes were classified using binary or multiple classification (Table 2, class). Thus, the presence or absence of labeled attacks must be assessed (Table 2, labeled), and the attack type is also classified (Table 2, class). As seen in Table 2, public datasets are often imbalanced (Table 2, balance).

2.2. Sampling Techniques for Handling Imbalanced Classes

An imbalanced dataset is one in which the number of samples belonging to each class is unequal. Methods designed to address this imbalance can be divided into data- and algorithmic-level solutions [16].

2.2.1. Data-Level Methods

Several data-level approaches, including general sampling methods, are known. These include over-, undersampling, and hybrid sampling methods. Undersampling gives rise to the concern that useful information in the dataset will be reduced. For example, Tomek et al. [17] proposed the Tomek link method (TOMEK), which minimized the distance between pairs of nearest neighbors of opposite classes. The appearance of two samples in a single Tomek link means that they are either noisy or close to the border and can be used for undersampling to remove samples belonging to the majority class.

Oversampling can create a decision region in which less learning takes place and becomes more specific; thus, the results are overfitted. SMOTE [14] interpolates minority

class instances that are close to each other to create synthetic minority class instances. However, SMOTE encounters the problem of overgeneralization, where the majority class is not accounted for, and the minority class region is generalized unconditionally. To resolve this problem, improved sampling methods have been proposed.

An extended SMOTE routine uses an ensemble iterative partitioning filter [18] that overcomes problems caused by noise and borderline instances in imbalanced class sets. Dong et al. [19] proposed Random-SMOTE to expand decision regions. It differs from SMOTE in that it synthesizes new samples along the boundary between the two samples. B-SMOTE, proposed by Han et al. [20], oversamples and strengthens only borderline minority samples, classifying them into three groups: safe, dangerous, and noisy.

A sample is considered dangerous when more than half of the nearest neighbors of the minority sample, m, are majority samples. In the case of a safe sample, more than half of the m are minority samples. In the case of a noisy sample, m neighbors are majority samples. Borderline instances of the minority class are more often misclassified than instances far from the borderline. B-SMOTE outperforms SMOTE, but it oversamples only dangerous points and does not use information from safe samples. Haibo et al. [21] proposed an adaptive synthetic (ADASYN) sampling approach for imbalanced learning. ADASYN uses weight-value distributions in minority class instances that differ according to learning difficulty. More synthetic data are generated for minority class instances, which are difficult to learn, than for minority class instances, which are easier to learn.

2.2.2. Algorithm-Level Methods

Chawla et al. [22] proposed a new algorithm-level approach to solving imbalanced class sets based on a combination of SMOTE and a boosting process. Unlike standard boosting, which assigns the same weight values to all examples of improper classifications, SMOTE indirectly changes the updated weight values and compensates for skewed distributions by generating synthetic examples from rare or minority classes. AdaBoost, a traditional algorithm proposed by Freund and Schapire [23], is used to reduce errors that generate persistent classifiers. Its performance is slightly better than random guessing. Based on AdaBoost, Fan et al. [24] proposed a transformation algorithm that used misclassification costs to update the training distribution in continuous boosting rounds. Joshi et al. [25] proposed an improved algorithm that provided additional features to enhance the balance between recall and precision in data mining. Wu et al. [26] proposed a class boundary alignment algorithm to be used in combination with SVMs to handle the problem of imbalanced training data related to image and video sequences.

2.3. Studies on Class Imbalances in the Field of Intrusion Detection

Intrusion detection data are naturally imbalanced as they are produced by classifying malicious network traffic amid the much higher volume normal traffic. Intrusion attack types that present a significant threat have a ripple effect, but they do not often occur. Hence, their detection accuracy is unavoidably low. The studies listed in Table 3 were conducted to resolve this problem in the field of intrusion detection.

Sun et al. [27] proposed an improved SMOTE method based on a network cleaning (NCL) rule that calculated the ratio of each class, the average ratio based on the ratios, the standard deviation of class ratios, and an imbalance metric calculated by dividing the standard deviation by the class ratio. This metric is used to sample minority class data. An additional method was proposed that processed data assumed to be noise using NCL after sampling using the KDD Cup 99 dataset. SMOTE-NCL improved the area under the receiver operating characteristic (ROC) curve (AUC) of both the rare and normal classes.

Table 3. Related studies on imbalance handling of intrusion detection data.

Year	Author	Dataset	Proposed Sampling Methods	Algorithms
2016	Sun and Liu [27]	KDD Cup 99	SMOTE-NCL	KNN, SVM, C4.5, NB
2017	Yan et al. [28]	NSL-KDD	RA-SMOTE	SVM, BPNN, RF
2019	Tripathi and Makwana [15]	KDD Cup 99	SMOTE+Ensemble	AdaBoost, RF
2019	Lee and Park [29]	CICIDS2017	GAN	RF
2019	Merino et al. [30]	UGR16	GAN	MLP
2020	Zhang et al. [31]	UNSW-NB15, CICIDS2017	SGM_CNN	CNN, RF, MLP
2020	Bedi et al. [32]	NSL-KDD	Siamese-NN	CNN, DNN
2020	Zhang et al. [33]	NSL-KDD	ReliefF+B-SMOTE	KNN, C4.5, NB
2020	Ma and Shi [14]	NSL-KDD	AESMOTE	-
2021	Liu et al. [34]	NSL-KDD, CICIDS2018	DSSTE	RF, SVM, XGBoost, LSTM etc.

Yan et al. [28] demonstrated that B-SMOTE and SMOTE+ENN helped resolve the overfitting problems of conventional SMOTE; however, they were found to be unsuitable for intrusion detection. Subsequently, a region-adaptive SMOTE method was advanced, which effectively improved the detection rate of rare classes (e.g., user-to-root (U2L) and remote-to-local (R2L) attacks). NSL-KDD was used as the dataset, and the detection rate improved. Tripathi et al. [15] proposed a combination RF+AdaBoost method to mitigate the class imbalance problem of the KDD Cup 99 dataset. Rare classes (i.e., U2R and R2L) were examined at various ratios from 50 to 1000%, alongside ensemble classification.

Bedi et al. [32] proposed the Siam intrusion detection system (IDS) method, which identified the homogeneity between classes by calculating similarity scores for the pairs of inputs needed to handle class-imbalance problems. High recall values were obtained in CNN- and DNN-based IDSs. However, its performance was lower than that of conventional IDSs. Lee and Park [29] noted that the class imbalance problem was larger when using newer deep learning techniques. They also illuminated the weaknesses of existing data imbalance resolution techniques, including data loss and overfitting. To resolve these problems, a generative adversarial network (GAN) was proposed to generate new virtual data similar to existing ones. Their model outperformed SMOTE, and CICIDS2017 was used as the dataset. Oversampling was performed using a GAN. Rare classes (e.g., Bot, Infiltration, and Heartbleed) comprised less than 0.1% of the dataset.

This examination of approaches to mitigate data imbalances in the field of intrusion detection has revealed that data- and algorithm-level approaches are shifting toward hybrid forms, and studies are being actively conducted using GANs [29] and similar algorithms to resolve these problems.

3. Proposed Method

To overcome the problems of imbalanced and duplicate data for machine learning intrusion detection capabilities, an ensemble mixed sampling method (EmSM) is proposed. We apply the SMOTE+ENN resampling technique to generate a balanced dataset. An ensemble bagging algorithm is then applied to the balanced dataset to improve the accuracy of minority class prediction while maintaining the accuracy of predicting the majority class.

3.1. Process of the Proposed Approach

As shown in Figure 1, this process involves the use of the proposed sampling technique to create a class-balanced training dataset using imbalanced classes to increase the modeling robustness. Efficiency is verified by comparing the training and class-balanced datasets before and after sampling using t-SNE visualization. The details of this process are as follows:

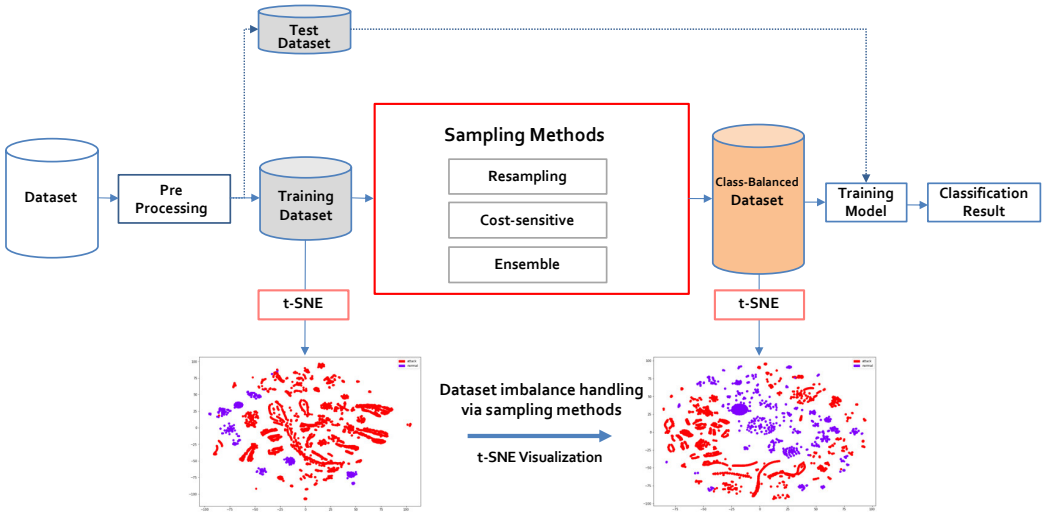

Figure 1. Sampling process for handling imbalanced classes.

First, the dataset with imbalanced classes is subjected to preprocessing for feature extraction based on intrusion detection characteristics. Using the sampling technique on the data before preprocessing may not yield the intended results, as the data distribution may differ after preprocessing. The sampling technique may also differ according to the preprocessing methods used for extraction and feature selection. Second, the dataset is divided into testing and training sets after preprocessing, which are then classified at a fixed ratio per class. Binary classification classifies the data into attack and normal types, and multi-classification classifies them by attack type. Third, the data are classified into majority and minority classes. Then, the sampling method is applied to process the imbalanced data. The sampling technique then generates a class-balanced dataset, and a model is generated, which is optimized by measuring its performance against the test dataset.

3.2. Ensemble Mixed Sampling Method

As shown in Figure 2, the proposed EmSM method is performed as follows. First, the training dataset is preprocessed before resampling. For the features of the intrusion detection dataset, we select hybrid features extracted from the domain and attack features according to the process of Jung et al. [35].

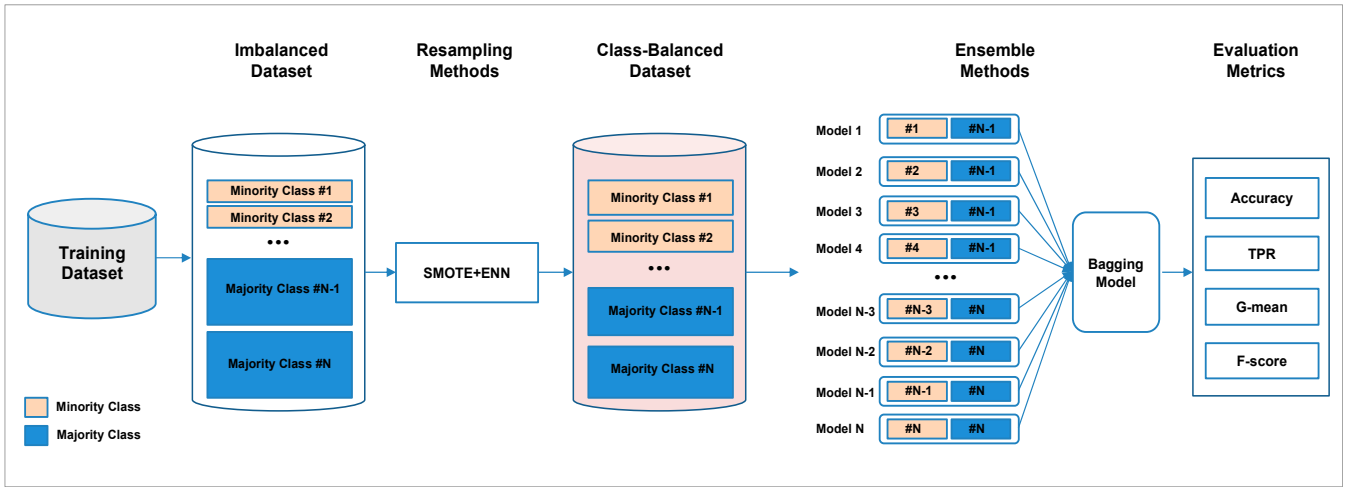

Figure 2. Ensemble mixed sampling method.

Second, the minority and majority classes are selected according to label characteristics. As the intrusion detection dataset is highly imbalanced and has many duplicates, we apply

SMOTE+ENN, wherein synthetic data of the minority class are generated via SMOTE oversampling, and noise and duplicates are removed via ENN undersampling. This reduces data loss and improves the demarcation of classes.

Third, for the newly generated class-balanced dataset, the ensemble method balanced bagging is used to address imbalanced data by randomly restoring and estimating new data from the given training data to construct multiple small training datasets of the same size. The final classification and prediction are performed using these small training datasets, which reliably improves the performance of the model more effectively than general bagging algorithms (e.g., RF). Fourth, balanced accuracy, *G-mean*, and *F*1-*score* are used for evaluation of the imbalanced data.

4. Experiments and Evaluation

This section describes the experiments and evaluations that were conducted using EmSM in three scenarios to improve the machine learning intrusion detection performance on imbalanced data. The first scenario included binary classified dataset labels: normal or attack. The extent to which the sampling techniques affected the machine learning performance was measured. The second scenario included multi-classified labels that were classified based on the attack type. A number of performance changes occur as a result of using the sampling technique. In particular, our aim was to determine whether the performance of the majority class could be maintained while the performance of the minority class was improved. In the third scenario, we compared EmSM to extant resampling techniques.

Two public datasets (i.e., PKDD2007 and CSIC2012) with different degrees of imbalance were selected to demonstrate the efficacy of the EmSM techniques. Class imbalance handling techniques were compared, including the baseline (no sampling) and nine existing techniques (i.e., random undersampling (RUS), ENN, TOMEK, random oversampling (ROS), SMOTE, B-SMOTE, ADASYN, SMOTE+ENN, and SMOTE+TOMEK) [12]. RF, MLP, and XGBoost were also used to evaluate the model after the imbalance-handling techniques were applied. EmSM was then applied to the imbalanced dataset.

The analysis of the algorithmic effects, attack ratio, and data distribution as factors affecting the sampling results are discussed in Section 4.5.

4.1. Datasets

The characteristics of the two selected datasets are listed in Table 4. These datasets were tested by dividing the binary classification into attack and normal types and by dividing the multi-classification according to the attack type. PKDD2007 and CSIC2012 have low and high imbalance ratios, respectively. Most datasets in the field of intrusion detection consist of preprocessed data. However, the two selected datasets include hypertext transfer protocol request headers and payloads; thus, the relationship with feature extraction can be studied during the experiments.

4.1.1. PKDD2007

PKDD2007 was created in response to a web traffic analysis challenge at the combined 18th European Conference on Machine Learning and the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases [11]. As part of the challenge, participants were provided with a dataset containing normal traffic and seven types of attack traffic. It included 35,006 requests classified as normal and 15,110 requests classified as attack (i.e., cross-site scripting (XSS), structured query language injection (SQLi), lightweight directory access protocol (LDAP) injection (LDAPi), extended markup language (XML) path (XPATH) injection, path traversal, command extraction, and server-side included (SSI) attacks). The dataset consists of classes comprising each type of request, as shown in Figure 3.

Table 4. Description of datasets (PKDD2007 and CSIC2012).

Datasets	Category	Payload	Labeled	Binary Class (Normal/Attack)	Multi Class (Attack Type)	Source	Balanced	IR
PKDD2007	WAF *	O	O	50,116 (35,006/15,110)	XSS SQL Injection LDAP Injection XPATH Injection Path Traversal Command Extraction SSL Attack	ECML/PKDD	O	18
CSIC2012	WAF *	O	O	65,767 (8363/57,404)	XSS SQLi Buffer Overflow LDAP XPath FormatString SSI CRLFi Anomalous	CSIC/TORPEDA	X	1160

* WAF, web application firewall.

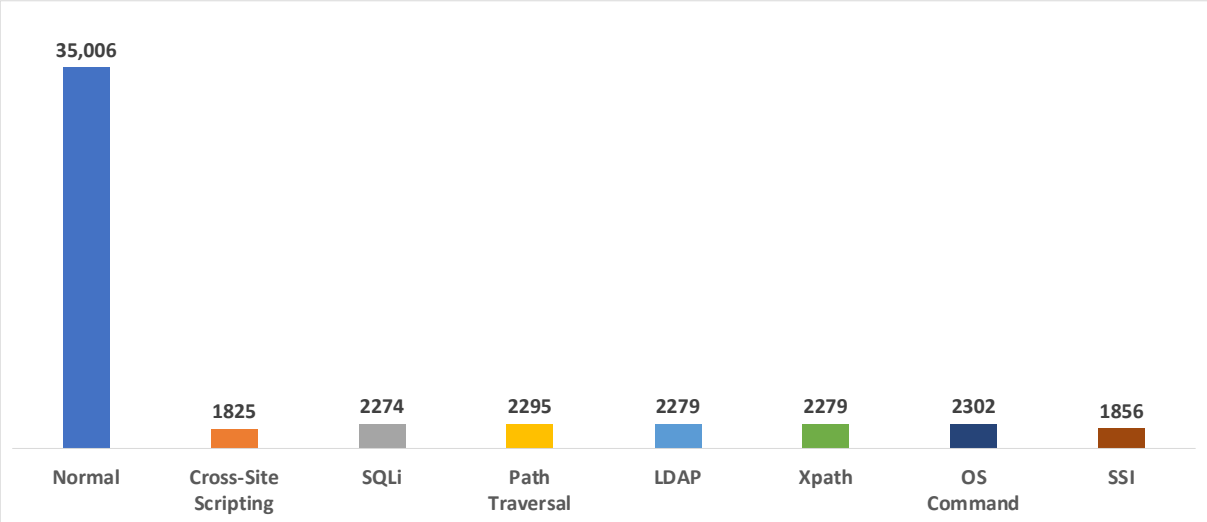

Figure 3. PKDD2007 dataset classes according to attack type.

Additionally, the presence or absence of an attack in the dataset and the label for each type were visualized using t-SNE, as shown in Figure 4.

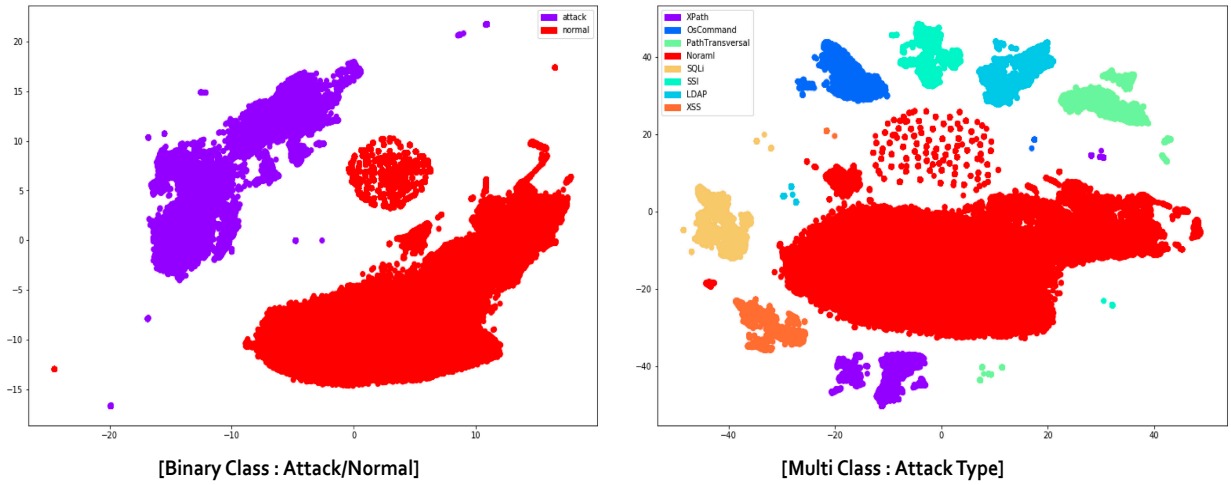

Figure 4. Results of t-SNE visualization for the PKDD2007 dataset.

4.1.2. CSIC2012

The CSIC2012 dataset was presented as part of the Torpedo framework at RECSI2012. The framework was used to develop labeled web traffic for web-attack detection system evaluation and testing [9]. The data are comprised of 10 classes, including 8363 requests classified as normal, 16,456 requests classified as anomalous, and 40,948 requests classified as attacks. The 10 attack types include nor mal, XSS, SQLi, buffer overflow, LDAP, XPath, FormatString, SSI, carriage return-line feed injection (CRLFi), and anomalous. The dataset applies the XML file format and consists of labels and requests. A request is divided into method, protocol, path, headers, and body.

The dataset consists of classes for each type of request, as shown in Figure 5. The t-SNE visualization of the labels for the presence/absence of attacks and each attack type for the dataset are shown in Figure 6.

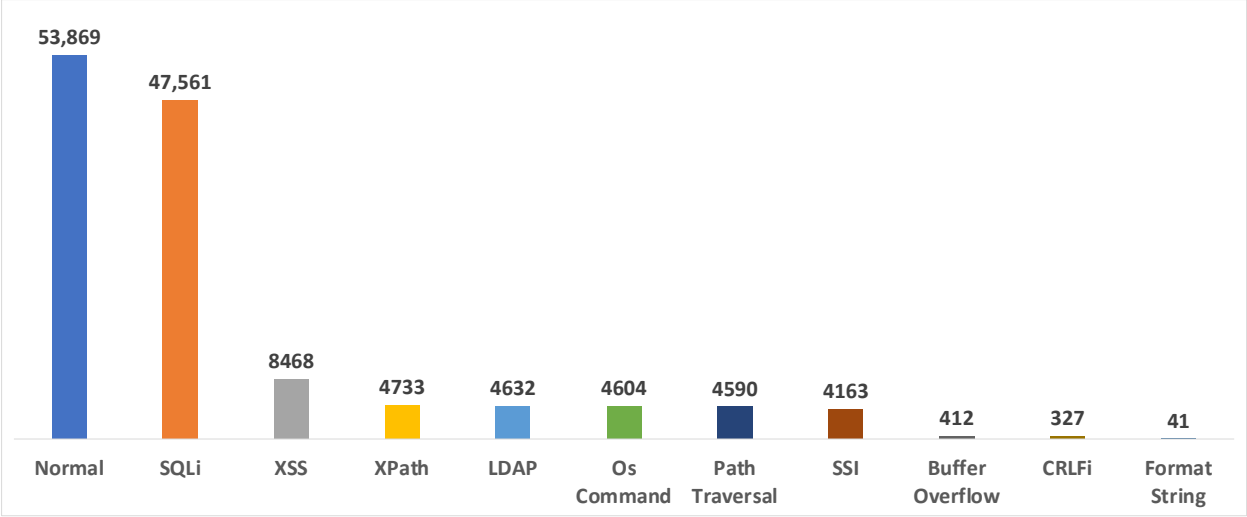

Figure 5. CSIC2012 dataset classes according to attack type.

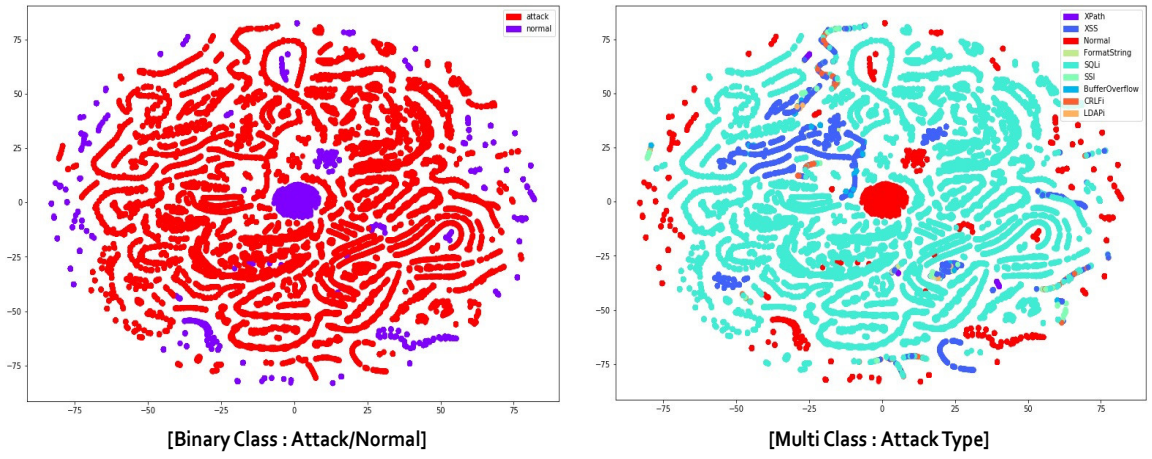

Figure 6. Results of t-SNE visualization for the CSIC2012 dataset.

4.2. Data Preprocessing

This stage entailed the use of data preprocessing to prepare the datasets for machine learning and comprised the following steps: normalization, field selection, and feature extraction and selection.

4.2.1. Normalization

The collected datasets consisted of typical XML format data. First, the same normalization was performed on the datasets in the form of method, version, universal resource indicator (URI), query, and body. The values containing user information were included in the body field. For the URI, query, and body fields, the "\n" character was removed, and URI decoding was applied.

4.2.2. Field Selection

The fields used in the experiment were selected from each dataset. In PKDD2007 and CSIC2012, the category type was divided into class, method, and version. For the text type, the URI, query, and body were selected. At this point, if a query did not exist in the data generated by CSIC2012, a missing value was noted as the "?" character in those fields.

4.2.3. Feature Extraction and Selection

This study is based on a method for extracting the feature of an attack from the header and payload portion of an intrusion detection event by considering the characteristics of a web intrusion detection dataset with payloads. The feature extraction method has also been verified for performance by Pastrana et al. [36] and Torrano-Gimenez et al. [37]. Features were extracted using the keywords associated with the attack types for the separated fields: http_url, http_query, and http_body. The features were then categorized, and string vectorization was conducted [35].

4.3. Evaluation Environment and Metrics

The experimental environment was implemented using Python in Ubuntu 18.04.2 LTS. Scikit-learn 0.20.4 was used as the machine learning algorithm. The hardware specifications included two Nvidia GeForce RTX 2060 GPUs, 128-GB RAM, 8-GB SSD, and an AMD Ryzen Threadripper 1900X 8-core processor.

The selected evaluation method was used to calculate the confusion matrix metric, an approach generally used in machine learning. The confusion matrix contains four types of information: true positive (TP) refers to samples that are actually positive (attack) samples that are correctly judged as positive (attack) samples; false negative (FN) refers to samples that are actually positive (attack) samples, but are mistakenly judged to be negative (normal) samples; false positive (FP) refers to negative (normal) samples that are misjudged as positive (attack) samples; true negative (TN) refers to samples that are actually negative (normal) samples, and are correctly judged as negative (normal) samples [34].

The experimental results were evaluated using the following evaluation metric based on the confusion matrix. *Accuracy* (2) is defined as the ratio of items that were correctly classified as "normal" or "attack" for all sample items. *Precision* (3) refers to the ratio of items that were classified as actual attacks to the items that were predicted to be attacks. *Recall* (4) is the ratio of items that were predicted to be attacks to actual attacks. It has the same meaning as the detection ratio used for intrusion detection datasets. The *F1-score* (5) is the harmonic mean of *precision* and *recall*.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN), \tag{2}$$

$$Precision = TP/(TP + FP), \tag{3}$$

$$Recall = TP/(TP + FN), \tag{4}$$

$$F1\text{-}score = (2 * Recall * Precision)/(Recall + Precision). \tag{5}$$

However, these general evaluation methods can produce biased or skewed results on unbalanced datasets. Accordingly, many researchers have suggested various evaluation methods (*G-mean*, *F1-score*, PR AUC, Brier Score, etc.) to compensate for the imbalance. In this study, our selection of metrics supported our aim to improve the performance of the minority class and also maintain the performance of the majority class. Thus, the false

positive rate (*FPR*) (8), true negative rate (*TNR*) (9), and *G-mean* (7) were added to these universal evaluation metrics. FPR refers to the rate at which normal traffic is correctly predicted to be an attack, and TNR refers to the rate at which normal traffic is correctly predicted to be normal. The *G-mean* value proposed by Kubat and Matwin [38] is calculated as the geometric mean of sensitivity and specificity [12].

$$Balaced\ Accuracy(BA) = ((TP/(TP+FN)) + (TN/(TN+FP)))/2, \quad (6)$$

$$G\text{-}mean = TP/(TP+FP), \quad (7)$$

$$FPR = FP/(TN+FP), \quad (8)$$

$$TNR = TN/(TN+FP). \quad (9)$$

Therefore, accurate classification of the majority groups would cause the *accuracy* and *G-mean* values of the minority groups to be low. Thus, as the *accuracy* of the minority groups improves, the value of *G-mean* increases. In multi-classification problems, the index is calculated using the weighted average method according to the number of samples for each class to evaluate the detection performance of the model on the imbalanced class set.

4.4. Experimental Results

We now report the experimental results for the three given scenarios.

4.4.1. Scenario 1: Binary Classification

PKDD2007 and CSIC2012 both contain imbalanced data, although the data in CSIC2012 are much more imbalanced. Table 5 lists the number of data points before and after PKDD2007 and CSIC2012 sampling. The "Base" column contains the number of data points before sampling. For the undersampling methods (i.e., RUS, ENN, and TOMEK), the number was reduced to match the 6691 normal minority class data. In the case of ENN and TOMEK, the number of data points was determined according to the sampling properties of each, rather than accurately matching the minority class (e.g., RUS). With respect to the oversampling methods, ROS, SMOTE, and B-SMOTE generated 39,449 results similar to the attack majority class, but ADASYN increased the number beyond that.

Table 5. Data distribution after sampling (datasets CSIC2012 and PKDD2007).

DATASET	Label	BASE	RUS	ENN	TOMEK	ROS	SMOTE	B-SMOTE	ADASYN	SMOTE +ENN	SMOTE +TOMEK
CSIC2012	Attack	39,449	6691	6685	39,449	39,449	39,449	39,449	39,449	39,446	39,449
	Normal	6691	6691	6691	6691	39,449	39,449	39,449	39,451	39,449	39,449
PKDD2007	Attack	36,405	24,179	25,239	35,789	36,405	36,405	36,405	36,405	23,630	35,906
	Normal	24,179	24,179	24,179	24,179	36,405	36,405	36,405	37,296	26,009	35,906

The traditional sampling methods (i.e., ROS and RUS) reduced the imbalances of the training set and produced synthetic data that closely resembled the actual data. The RUS algorithm can lose valid information, which leads to data duplication and overfitting. Simultaneously, SMOTE and B-SMOTE can increase the number of difficult samples in the training set by generating noise traffic and data overlaps.

Table 6 presents the results of the performance assessment of the model after sampling using XGBoost as the algorithm. The lower the FPR, the more accurate the results. For the base data, the FPR was 0.0024 for CSIC2012 and 0.0203 for PKDD2007. In the case of the highly imbalanced CSIC2012 dataset, the oversampling (i.e., ROS, SMOTE, B-SMOTE, and ADASYN) and hybrid sampling (i.e., SMOTE+ENN and SMOTE+TOMEK) methods produced satisfactory results. However, in the case of PKDD2007, which contained balanced classes, the improvement was not significant. Moreover, apart from TOMEK, most of the methods did not produce satisfactory results. If the majority group is classified accurately but the ability to predict the minority group is low, the *G-mean* value is low. Hence, the *G-mean* is an important metric for identifying class imbalances. Here, the results obtained

with the undersampling methods were superior to those of the base, except for ENN, and the oversampling and hybrid sampling methods both yielded reliable results. In terms of the recall values and *F*1-*score*s, the base and mixed-sampling methods produced satisfactory results, whereas the performances of the undersampling and oversampling methods were slightly lower.

Table 6. Model performance results after binary sampling (datasets CSIC2012 and PKDD2007, algorithm XGBoost).

DATASET	Label	BASE	RUS	ENN	TOMEK	ROS	SMOTE	B-SMOTE	ADASYN	SMOTE +ENN	SMOTE +TOMEK
CSIC2012	Accuracy	0.9994	0.9991	0.9992	0.9990	**0.9999**	**0.9999**	0.9997	0.9997	0.9997	**0.9999**
	FPR	0.0024	0.0006	0.0036	0.0054	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
	G-mean	0.9987	0.9992	0.9981	0.9972	**0.9999**	**0.9999**	0.9998	0.9998	0.9998	**0.9999**
	F1-score	0.9996	0.9995	0.9995	0.9994	**0.9999**	**0.9999**	0.9998	0.9998	0.9998	**0.9999**
PKDD2007	Accuracy	**0.8865**	0.8741	0.8524	0.8857	0.8749	0.8544	0.8400	0.8772	0.8718	0.8809
	FPR	0.0203	0.0596	0.1182	**0.0160**	0.0682	0.1137	0.1521	0.0441	0.6054	0.0343
	G-mean	0.8549	0.8532	0.8442	0.8520	**0.8575**	0.8454	0.8380	0.8515	0.8481	0.8529
	F1-score	0.8398	0.8306	0.8137	0.8375	0.8342	0.8154	0.8051	0.8313	0.8256	**0.8346**

In summary, imbalance handling had a positive effect on machine learning performance with the datasets. Additionally, after class-imbalance handling, the results obtained for the CSIC2012 dataset were more satisfactory than those of the PKDD2007 dataset.

4.4.2. Scenario 2: Multi-Classification

The labels in this experiment reflected the different attack types. The CSIC2012 dataset was used for this experiment. The class imbalance-handling technology included the base pre-sampling dataset and nine sampling technologies. The purpose of this experiment was to improve the detection rate for the minority class while maintaining the detection rate for the majority class of an intrusion detection dataset with a class imbalance.

The results of sampling the CSIC2012 dataset in Table 7 indicate that the majority classes were SQLi, normal, and XSS, whereas the minority classes were FormatString (F/S), LDAPi, XPath, and CRLFi. To examine the problem more closely, SQLi and XSS were used as the majority classes, whereas FormatString (F/S) and LDAPi were the minority classes.

Table 7. Data distribution after sampling (dataset CSIC2012).

DATASET	Label	BASE	RUS	ENN	TOMEK	ROS	SMOTE	B-SMOTE	ADASYN	SMOTE +ENN	SMOTE +TOMEK
CSIC2012	XPath	143	33	75	142	34,351	34,351	34,351	34,343	**32,701**	34,351
	XSS	3907	33	3382	3894	34,351	34,351	34,351	34,326	**33,532**	34,351
	B/O *	324	33	228	319	34,351	34,351	34,351	34,344	**31,650**	34,351
	SQLi	34.351	33	34,128	34,350	34,351	34,351	34,351	34,351	**34,079**	34,351
	SSI	371	33	126	366	34,351	34,351	34,351	34,377	**33,676**	34,351
	F/S *	33	33	33	33	34,351	34,351	34,351	34,355	**32,917**	34,351
	LDAPi	62	33	8	56	34,351	34,351	34,351	34,359	**30,899**	34,351
	CRLFi	258	33	65	256	34,351	34,351	34,351	34,326	**33,751**	34,351

* B/O, buffer overflow; F/S, FormatString.

In the next step, we evaluated the models generated by the RF, MLP, and XGBoost algorithms after sampling. The results in Table 8; Table 9 were obtained after using 10 sampling techniques, including the base, undersampling (i.e., RUS, ENN, and TOMEK), oversampling (i.e., ROS, SMOTE, B-SMOTE, and ADASYN), and hybrid sampling (i.e., SMOTE+ENN and SMOTE+TOMEK). The *G-mean* and *F1-score* values are shown as the evaluation metrics.

Table 8. Model performance results after sampling (dataset CSIC2012, *G-mean*).

	BASE	RUS	ENN	TOMEK	ROS	SMOTE	B-SMOTE	ADASYN	SMOTE +ENN	SMOTE +TOMEK
F/S_RF *	0.8655	0.9983	0.8656	0.9991	0.8657	0.9997	0.9997	0.9997	**0.9998**	0.9997
LDAPi_RF	0.9253	0.9245	0.8864	0.8861	**0.9258**	0.8864	0.8864	0.8452	0.8864	0.8864
SQLi_RF	0.9886	0.9588	0.9981	0.9853	0.9986	**0.9988**	**0.9988**	**0.9988**	**0.9988**	**0.9988**
XSS_RF	0.9479	0.8693	0.9851	0.9452	0.9878	0.9894	0.9893	0.9894	**0.9909**	0.9894
F/S_XGBoost *	0.8660	0.7785	1.0000	0.7906	**1.0000**	**1.0000**	**1.0000**	0.9999	**0.9999**	**1.0000**
LDAPi_XGBoost	0.9258	0.9192	0.9258	0.9258	**0.9258**	0.8864	**0.9258**	0.6545	0.8864	0.8864
SQLi_XGBoost	0.9986	0.8724	0.9976	0.9969	0.999	**0.9991**	**0.9991**	0.9972	**0.9991**	**0.9991**
XSS_XGBoost	**0.9922**	0.9488	0.9868	0.9869	0.9885	0.9896	0.9917	0.9844	0.9922	0.9896
F/S_MLP *	0.5000	0.5587	0.7071	0.3535	0.9998	0.9997	0.999	0.7905	**0.9998**	0.9997
LDAPi_MLP	**0.8864**	0.5695	0.8863	0.8863	**0.8864**	**0.8864**	0.7071	0.6546	0.8858	0.8863
SQLi_MLP	0.9988	0.0000	0.9985	0.9985	0.999	0.9991	0.9984	0.9985	0.999	0.9991
XSS_MLP	**0.9841**	0.5793	0.9793	0.9831	0.9821	0.9801	0.9728	0.984	0.9801	0.9790

* F/S_RF, FormatString + random forest; F/S_XGBoost, FormatString + XGBoost; F/S_MLP, FormatString + MLP.

Table 9. Model performance results after sampling (dataset CSIC2012, *F1-score*).

	BASE	RUS	ENN	TOMEK	ROS	SMOTE	B-SMOTE	ADASYN	SMOTE +ENN	SMOTE +TOMEK
F/S_RF	0.4444	0.2857	0.4615	0.4324	0.5217	0.6667	0.6667	0.6957	**0.7619**	0.6667
LDAPi_RF	0.6154	0.4068	0.8800	0.6875	**0.9231**	0.8800	0.8800	0.8333	0.8800	0.8800
SQLi_RF	0.9890	0.9579	**0.9994**	0.9856	0.9990	0.9990	0.9990	0.9990	0.9990	0.9990
XSS_RF	0.9418	0.8590	0.9821	0.9339	0.9853	**0.9880**	0.9869	0.9874	**0.9880**	**0.9880**
F/S_XGBoost	0.8571	0.0277	1.0000	0.7692	0.9412	0.9412	**1.0000**	0.8889	0.8889	0.9412
LDAPi_XGBoost	**0.9231**	0.1257	**0.9231**	**0.9231**	**0.9231**	0.8462	0.9231	0.4444	0.8462	0.8800
SQLi_XGBoost	**0.9995**	0.8647	0.9992	0.9990	0.9991	0.9991	0.9991	0.9974	0.9991	0.9991
XSS_XGBoost	0.9870	0.9475	0.9853	0.9817	0.9884	0.9895	**0.9916**	0.9694	0.9895	0.9895
F/S_MLP	0.3636	0.0031	0.6154	0.2000	0.8000	0.6667	0.4103	0.6250	**0.8000**	0.6957
LDAPi_MLP	0.8462	0.0093	0.8148	0.8148	**0.8800**	0.8462	0.6364	0.5217	0.5366	0.7857
SQLi_MLP	**0.9996**	0.0000	0.9995	0.9995	0.9991	0.9991	0.9984	0.9985	0.9990	0.9991
XSS_MLP	0.9714	0.1873	0.9767	0.9759	**0.9815**	0.9799	0.9636	0.9704	0.9799	0.9783

In the case of multi-classification, the results were slightly different for each algorithm, but the *F*1-*score* and *G-mean* of the minority classes, F/S and LDAPi, increased, and the metrics for the majority classes, SQLi and XSS, increased or decreased slightly owing to small errors. However, the values of the metrics varied according to the sampling method, indicating that the minority class detection performance was strengthened, and the detection performance of the majority class changed minimally, which was the goal of this study.

Therefore, dataset sampling techniques for imbalanced data handling had a positive effect on model performance, and it was confirmed that of all the sampling techniques, the hybrid SMOTE+ENN and SMOTE+TOMEK techniques produced the most stable results.

4.4.3. Scenario 3: Comparison with Ensemble Based Algorithms

This section compares the results of applying ensemble algorithms. First, SMOTE+ENN resampling was performed on an imbalanced dataset, and balanced bagging was applied to the resampled dataset. We compared the results before and after applying SMOTE to the base using traditional algorithms (i.e., RF, MLP, and XGBoost) and ensemble algorithms (i.e., AdaBoost, EasyEnsemble, RUSBoost, balanced bagging (BB), and balanced random forest (BRF)) [12].

As shown in Figure 7, applying an ensemble algorithm to imbalanced data lowered the performance as compared with the use of resampling. Overall, the dataset that was subject to imbalance handling yielded favorable results. As compared with the application of a traditional algorithm to a dataset resampled using SMOTE or a similar technique, the accuracy or *G-mean* of EmSM did not reflect a noticeable improvement in the balance, although it did yield a superior *F*1-*score* of at least 4%. In particular, the proposed method resulted in much more stable *F*1-*score*, *G-mean*, and balanced accuracy results.

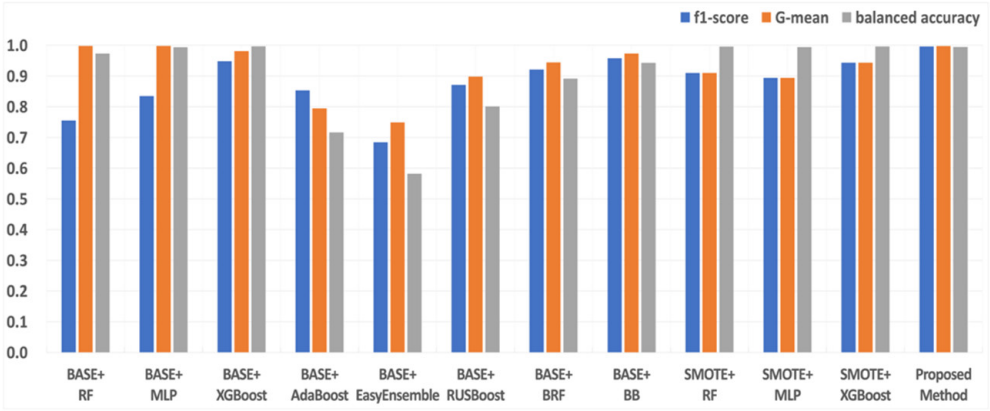

Figure 7. Comparison of the performance of different sampling methods.

4.5. Results and Discussion

Using three experimental scenarios, this study applied and compared various sampling techniques to determine their effectiveness in handling imbalanced intrusion detection datasets. First, nine resampling techniques were applied and compared in Scenarios 1 and 2. As a result, it was confirmed that the sampling methods had the effect of both binary and multi-classification in common. Binary classification yielded good results with various performance metrics (accuracy, FPR, *G-mean*, *F1-score*) using the sampling technique. In particular, good results were obtained in oversampling and hybrid sampling. In multi-classification, the gap between the *G-mean* and *F1-score* for each minority and majority class was wide before sampling methods were used, and it could be seen that the gap between the minority and majority classes was narrowed after using sampling methods. This approach enables the performance of the majority class to be maintained while improving the performance of the minority class. Second, in Scenario 3, the first imbalance data-handling technique using sampling and ensemble sampling yielded more satisfactory results than applying only ensemble sampling (i.e., AdaBoost, EasyEnsemble, RUSBoost, balanced bagging, and balanced RF). In particular, the proposed method yielded favorable results in terms of the *F1-score*, *G-mean*, and balanced accuracy.

However, in this study, the degree of influence of the algorithm, attack ratio, and data distribution, which are factors that affect the sampling results for imbalanced data handling, was also investigated. We found that the sampling method affected the results of RF, MLP, and XGboost in particular. Generally, XGBoost is not greatly affected by imbalanced data, whereas RF and others show improved results with more balanced data.

The performance of the model changes markedly when the attack class ratio exceeds 70%. This indicates that the dataset class ratio must be adjusted such that the imbalance ratio is below 70%. Moreover, the data distribution changes caused by preprocessing and feature extraction techniques influences the results.

In summary, we confirmed that the performance of the proposed sampling technique for imbalanced intrusion detection datasets is superior to that of existing data-resampling methods or ensemble approaches. In particular, it improves the classification performance for the minority class while maintaining the performance for the majority class, demonstrating its ability to successfully improve intrusion detection.

4.5.1. Results of Each Algorithm before and after Sampling

The features of the respective algorithms became apparent by considering the multi-classification performance on a dataset with imbalanced classes. As shown in Figure 8, the results produced by tree-structured algorithms (e.g., RF) in terms of the accuracy, precision, recall, and *F1-score* were stabilized as a result of sampling. In the case of MLP and XGBoost, sampling did not significantly affect the outcome. By comparison, after sampling, the machine-learning performance improved in the order RF > MLP > XGBoost.

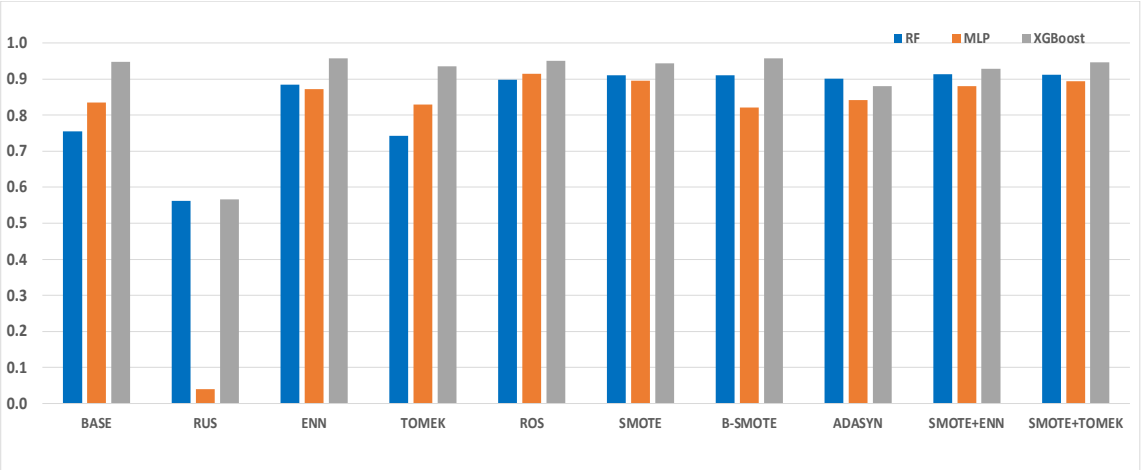

Figure 8. Results of each algorithm before and after sampling (*F*1-*score*, dataset CSIC2012).

4.5.2. Results According to Attack Ratio

Figure 9 compares the *F*1-*score*s according to the attack class ratio as another way to measure the effectiveness of the different methods. As shown, for PKDD2007, the performance of the model does not depend on the sampling technique or the modeling method for attack data ratios of 40% or less. However, the performance of the model changed markedly relative to that of the original base dataset for attack data ratios of 70% or more. We can conclude that the proposed method for data imbalance handling is effective, and that the extent of change varies according to the algorithm.

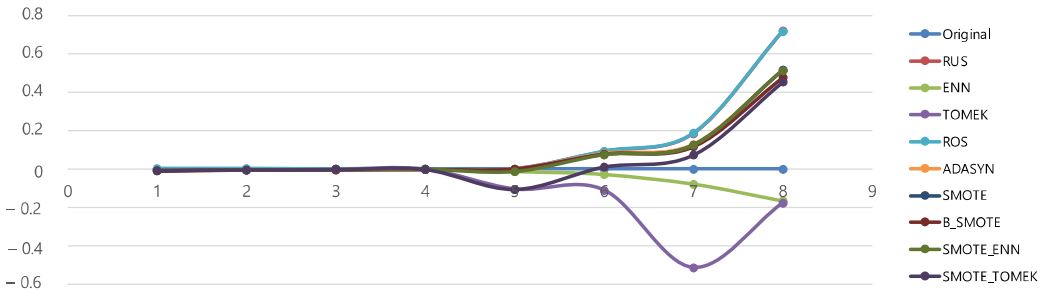

Figure 9. Results of the sampling models according to attack data ratio (algorithm RF, *F*1-*score*, dataset PKDD2007).

4.5.3. Results According to Data Distribution

The effect of the sampling technique and data distribution on the results was determined by applying the RF algorithm to CSIC2012. The data distribution was changed by subjecting the strings to vectorization when features were extracted from the payload segments to change the data distribution. The sampling techniques that were used were custom [35], term frequency-inverse document frequency (TF-IDF), hash, and Word2vec string vectors. As shown in Figure 10, the performance results depend on the type of string vector. The base was the optimal sampling technique for custom, and RUS was the optimal technique for TF-IDF. In the case of hash, the optimal sampling technique was ADASYN. These results show that the optimal sampling technique varies according to the data distribution, which changes for each feature extraction.

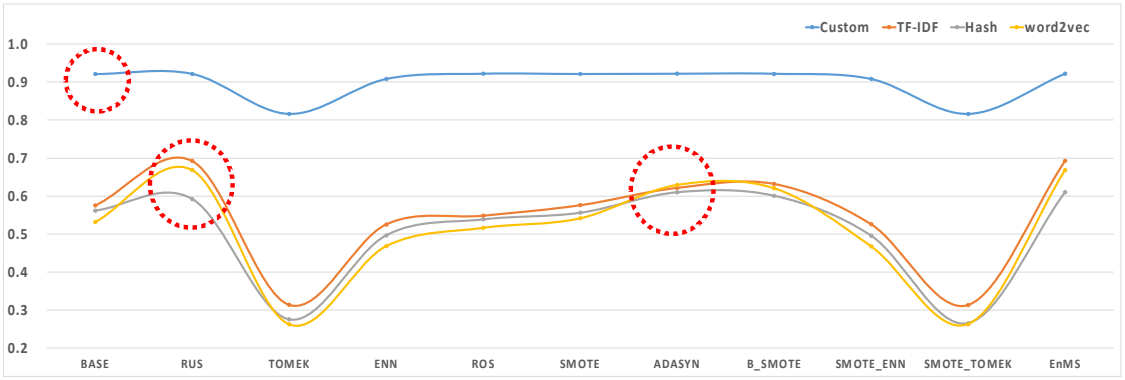

Figure 10. Results of sampling models according to data distribution (algorithm RF, *F1-score*, dataset CSIC2012).

5. Conclusions and Future Work

The class imbalance issue, which is common in the field of intrusion detection, significantly increases the difficulty of using existing algorithms for classification. In this paper, we propose EmSM, which combines a hybrid resampling technique and an ensemble technique to overcome the data imbalance problem in the intrusion detection field. As a result, it was confirmed that the sampling methods had the effect of both binary and multi-classification in common. Binary classification yielded good results with various performance metrics using the sampling technique. In multi-classification, the gap between the *G-mean* and *F1-score* for each minority and majority class was narrowed after using sampling methods. Furthermore, the proposed technique improved true positive identification and classification of serious threats that rarely occur, representing a major functional innovation. However, subsequent experiments showed that many factors (applied algorithms, attack ratios, and data distribution by features) could adversely affect the sampling results and could cause difficulties in practical applications. Therefore, in the future, we aim to increase the utilization of machine learning in the intrusion detection field by attempting various approaches such as more extended GAN and one-class classification to solve the imbalance problem in consideration of these problems.

Author Contributions: Conceptualization, I.J.; Methodology, I.J. and C.C.; Software, I.J.; Validation, I.J. and J.J.; Writing—Original Draft Preparation, I.J.; Writing—Review and Editing, I.J.; Visualization, I.J. and J.J.; Supervision, C.C. and J.J.; Project Administration, I.J. and C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted in 2021 with the support of the National Land, Korea Agency for Infrastructure Technology Advancement (KAIA) with funding from the government (Ministry of Land, Infrastructure and Transport) (21TLRP-B152768-03).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hussain, A.; Mohamed, A.; Razali, S. A Review on Cybersecurity: Challenges & Emerging Threats. In Proceedings of the 3rd International Conference on Networking, Information Systems & Security, ACM, Marrakech, Morocco, 31 March 2020; pp. 1–7.
2. Ibor, A.E.; Oladeji, F.A.; Okunoye, O.B. A survey of cyber security approaches for attack detection, prediction, and prevention. *Int. J. Secur. Its Appl.* **2018**, *12*, 15–28. [CrossRef]
3. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity* **2019**, *2*, 20. [CrossRef]
4. Pachghare, V.K.; Khatavkar, V.K.; Kulkarni, P.A. Pattern based network security using semi-supervised learning. *International J. Inf. Netw. Secur.* **2012**, *1*, 228–234. [CrossRef]

5. Lateef, A.A.A.; Al-Janabi, S.T.F.; Al-Khateeb, B. Survey on intrusion detection systems based on deep learning. *Period. Eng. Nat. Sci.* **2019**, *7*, 1074. [CrossRef]
6. Johnson, J.M.; Khoshgoftaar, T.M. Survey on deep learning with class imbalance. *J. Big Data* **2019**, *6*, 27. [CrossRef]
7. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 27 February 2022).
8. IDS 2017 Datasets Research Canadian Institute for Cybersecurity UNB. Available online: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 27 February 2022).
9. CSIC2012. Available online: https://www.tic.itefi.csic.es/torpeda/datasets.html (accessed on 27 February 2022).
10. The NSL-KDD Data Set. Available online: https://web.archive.org/web/20150205070216/http:/nsl.cs.unb.ca/NSL-KDD/ (accessed on 28 February 2022).
11. ECML/PKDD Workshop. Available online: http://www.lirmm.fr/pkdd2007-challenge/index.html#dataset (accessed on 28 February 2022).
12. Fernández, A.; García, S.; Galar, M.; Prati, R.C.; Krawczyk, B.; Herrera, F. *Learning from Imbalanced Data Sets*; Springer International Publishing: Cham, Switzerland, 2018; ISBN 978-3-319-98073-7.
13. Tavassoli, S.; Koosha, H. Hybrid Ensemble Learning Approaches to Customer Churn Prediction. *Kybernetes* **2022**, *51*, 1062–1088. [CrossRef]
14. Ma, X.; Shi, W. AESMOTE: Adversarial reinforcement learning with SMOTE for anomaly detection. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 943–956. [CrossRef]
15. Tripathi, P.; Makwana, R.R.S. An ensemble classification approach with selective under and over sampling of imbalance intrusion detection dataset. *Int. J. Secur. Its Appl.* **2019**, *13*, 41–50. [CrossRef]
16. Zhang, G.; Wang, X.; Li, R.; Song, Y.; He, J.; Lai, J. Network intrusion detection based on conditional Wasserstein generative adversarial network and cost-sensitive stacked autoencoder. *IEEE Access* **2020**, *8*, 190431–190447. [CrossRef]
17. Two Modifications of CNN. *IEEE Trans. Syst. Man Cybern.* **1976**, *SMC-6*, 769–772. [CrossRef]
18. Sáez, J.A.; Luengo, J.; Stefanowski, J.; Herrera, F. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf. Sci.* **2015**, *291*, 184–203. [CrossRef]
19. Dong, Y.; Wang, X. A New Over-Sampling Approach: Random-SMOTE for Learning from Imbalanced Data Sets. In *Knowledge Science, Engineering and Management*; Xiong, H., Lee, W.B., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7091, pp. 343–352. ISBN 978-3-642-25974-6.
20. Han, H.; Wang, W.-Y.; Mao, B.-H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In *Advances in Intelligent Computing*; Huang, D.-S., Zhang, X.-P., Huang, G.-B., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3644, pp. 878–887. ISBN 978-3-540-28226-6.
21. He, H.; Bai, Y.; Garcia, E.A. Shutao Li ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; IEEE: Piscataway, NJ, USA; pp. 1322–1328.
22. Chawla, N.V.; Lazarevic, A.; Hall, L.O.; Bowyer, K.W. SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In *Knowledge Discovery in Databases: PKDD 2003*; Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2838, pp. 107–119. ISBN 978-3-540-20085-7.
23. Freund, Y.; Schapire, R. A Short Introduction to Boosting. *J.-Jpn. Soc. Artif. Intell.* **1999**, *14*, 1612.
24. Fan, W.; Stolfo, S.; Zhang, J.; Chan, P. AdaCost: Misclassification Cost-Sensitive Boosting. *Int. Conf. Mach. Learn.* **1999**, *99*, 97–105.
25. Joshi, M.V.; Kumar, V.; Agarwal, R.C. Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements. In Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, 29 November–2 December 2001; IEEE: Piscataway, NJ, USA; pp. 257–264.
26. Wu, G.; Chang, E.Y. Class-Boundary Alignment for Imbalanced Dataset Learning. In Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets II, Washington, DC, USA, 21 August 2003; pp. 49–56.
27. Sun, Y.; Liu, F. SMOTE-NCL: A Re-Sampling Method with Filter for Network Intrusion Detection. In Proceedings of the 2nd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 14–17 October 2016; pp. 1157–1161.
28. Yan, B.; Han, G.; Sun, M.; Ye, S. A Novel Region Adaptive SMOTE Algorithm for Intrusion Detection on Imbalanced Problem. In Proceedings of the 3rd IEEE International Conference on Computer and Communications (ICCC), Chengdu, China, 13–16 December 2017; pp. 1281–1286.
29. Lee, J.; Park, K. GAN-based imbalanced data intrusion detection system. *Pers. Ubiquitous Comput.* **2021**, *25*, 121–128. [CrossRef]
30. Merino, T.; Stillwell, M.; Steele, M.; Coplan, M.; Patton, J.; Stoyanov, A.; Deng, L. Expansion of Cyber Attack Data from Unbalanced Datasets Using Generative Adversarial Networks. In *Software Engineering Research, Management and Applications*; Lee, R., Ed.; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2019; Volume 845, pp. 131–145. ISBN 978-3-030-24343-2.
31. Zhang, H.; Huang, L.; Wu, C.Q.; Li, Z. An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset. *Comput. Netw.* **2020**, *177*, 107315. [CrossRef]
32. Bedi, P.; Gupta, N.; Jindal, V. Siam-IDS: Handling class imbalance problem in intrusion detection systems using Siamese neural network. *Proc. Comput. Sci.* **2020**, *171*, 780–789. [CrossRef]

33. Zhang, J.; Zhang, Y.; Li, K. A Network Intrusion Detection Model Based on the Combination of ReliefF and Borderline-SMOTE. In Proceedings of the 4th High Performance Computing and Cluster Technologies Conference & 3rd International Conference on Big Data and Artificial Intelligence, ACM, Qingdao, China, 3 July 2020; pp. 199–203.
34. Liu, L.; Wang, P.; Lin, J.; Liu, L. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE Access* **2021**, *9*, 7550–7563. [CrossRef]
35. Jung, I.; Lim, J.; Kim, H.K. PF-TL: Payload feature-based transfer learning for dealing with the lack of training data. *Electronics* **2021**, *10*, 1148. [CrossRef]
36. Pastrana, S.; Torrano-Gimenez, C.; Nguyen, H.T.; Orfila, A. Anomalous Web Payload Detection: Evaluating the Resilience of 1-Grams Based Classifiers. In *Intelligent Distributed Computing VIII*; Camacho, D., Braubach, L., Venticinque, S., Badica, C., Eds.; Studies in Computational Intelligence; Springer International Publishing: Cham, Switzerland, 2015; Volume 570, pp. 195–200. ISBN 978-3-319-10421-8.
37. Torrano-Gimenez, C.; Nguyen, H.T.; Alvarez, G.; Petrovic, S.; Franke, K. Applying Feature Selection to Payload-Based Web Application Firewalls. In Proceedings of the 2011 Third International Workshop on Security and Communication Networks (IWSCN), Gjovik, Norway, 18–20 May 2011; IEEE: Piscataway, NJ, USA; pp. 75–81.
38. Kubat, M.; Matwin, S. Addressing the curse of imbalanced training sets: One-sided selection. In Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, USA, 8–12 July 1997; Volume 7, pp. 179–186.