AMERICAN UNIVERSITY OF BEIRUT

A SEQUENTIAL MULTI-STAGE ONE-CLASS CLASSIFICATION MODEL IN NETWORK INTRUSION DETECTION SYSTEMS

by
ALI MOUSSA KASSAB

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Industrial Engineering and Management
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2021

AMERICAN UNIVERSITY OF BEIRUT

A SEQUENTIAL MULTI-STAGE ONE-CLASS CLASSIFICATION MODEL IN NETWORK INTRUSION DETECTION SYSTEMS

by

ALI MOUSSA KASSAB

Approved by:

Dr. Jimmy Azar, Assistant Professor Industrial Engineering and Management	Advisor 
Dr. Nadine Marie Moacdieh, Assistant Professor Industrial Engineering and Management	Member of Committee 
Dr. Saif Al-Qaisi, Assistant Professor Industrial Engineering and Management	Member of Committee 

Date of thesis defense: September 3, 2021

AMERICAN UNIVERSITY OF BEIRUT

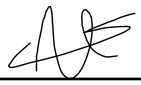
THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name:	Kassab	Ali	Moussa
	Last	First	Middle

⊗ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☒ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☐ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.



September 9, 2021

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

An Abstract of the Thesis of

Ali Moussa Kassab for Master of Engineering
Major: Industrial Engineering and Management

Title: A Sequential Multi-Stage One-Class Classification Model in Network Intrusion Detection Systems

One-class classification has been a promising direction in capturing the properties of a target class. Under multiclass classification problems with severe imbalance in target labels, research proposes the decomposition of a given problem into multiple sub-problems trained as separate one-class classifiers. We propose a sequential multi-stage one-class classification model to detect anomalies found in a multiclass classification context - a network intrusion detection system. We experiment with the model and test its performance using the NSL-KDD dataset, a modified version of the KDD'99 dataset. The model consists of several stages; we start with an initial classifier to detect the presence of an anomaly, followed by a sequence of per class one-class classifiers that will classify the intrusion based on the current class or otherwise pass to the next classifier trained on a less common attack type. Finally, we provide the analysis of our contribution compared to multiclass models trained over the dataset observations, and treated with an imbalanced learning approach.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Cyber attacks are growing in sophistication and offensive nature, leading to expanding challenges in accurately detecting intrusions and keeping organizations safe. There are a large number of cybercriminals around the world motivated to steal information and receive revenues illegitimately. IBM has recently estimated the average financial impact of a data breach to be about $3.8 million, and for companies at the enterprise level with at least a thousand employees, this number can grow ten to a hundred times larger [1]. The number of incidents reported with over $1M losses has also increased (Figure 1.1)[2].

Every year we witness the worst year ever for the number of cyber attacks around the world. Due to the Covid-19 pandemic, most organizations had to switch to working online, using weakly secure connections and leveraging the opportunity for hackers to attack vulnerable networks, damaging businesses and scamming citizens; by September 2020, 9.7 million healthcare records were compromised [3]. A cogent assumption stands out: the increase in the number of organizations relying on online networks for work will result in an expansion of cybercrime acts. In addition, failure to prevent the intrusions will degrade the credibility of security services and data integrity and availability, limiting the choices of organizations to flow under high risk or operate in a closed box. Thus, building a strong network intrusion detection system is crucial for the safety of businesses, governments, and the military.

In trying to detect network intrusions, two categories of data mining techniques were applied, misuse detection and anomaly detection: In misuse detection, each instance in the data is labeled as normal or intrusive, and the learning algorithm used was one of the popular classification models. Misuse detection techniques performed well against known attacks presented with slight variations, however, they failed in detecting attacks not previously observed.

On the other hand, an anomaly detection system was structured by building

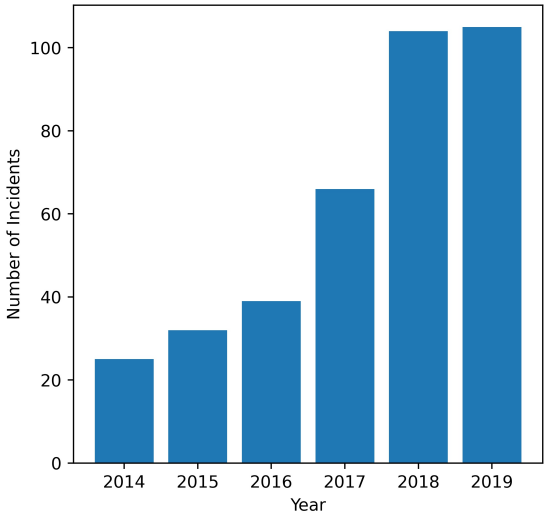


Figure 1.1: *Cyber Attack Incidents with over $1M in Reported Losses .*

a model over normal data, then focusing on detecting deviations from it. The new model preserves the need for learning from intrusions. The main approach then was building systems that focus on determining which instances stand out as being dissimilar to all others [4].

Anomaly detection systems have attracted the machine learning community. Systems of this kind generate a high class imbalance, between daily normal network connections, along with other classes, composed of different types of network attacks, some of which are barely represented. Numerous methods have been applied in the literature to tackle these threats, and one-class classifiers have proven their potential with high prediction accuracies.

However, researchers have been treating all network intrusions as an anomaly, while networks are being attacked in over 39 attack types (at least that we know of). These attack types can be classified into DoS, Probe, R2L, and U2R, and they have different properties and intentions when damaging a network [5]:

- Denial of Service (DoS): The denial of service attack makes some computing or memory resources too busy, by overloading the server with a surge of requests often from a botnet of compromised computers (zombies), whose unwary users may be distributed all over the globe. Thus the legitimate and widely varying range of IP addresses make such attacks difficult to block by traditional means such as configuring iptables of gateway routers or reverse proxies. The network appears full and unable to handle legitimate requests. You can no longer be granted a server connection.

- Probe: Probing is a class of attacks, where an attacker scans network data being transferred to collect information. In addition, the attacker can be using probing to find known vulnerabilities or weaknesses in the network machines and exploit attacks at the weak points.

- Remote to Local Attacks (R2L): A remote to local attack sends packets to the machine trying to exploit the machine's vulnerability and illegally gaining local access as a user.

- User to Root Attacks (U2R): In a user to root attack, the attacker starts with access to a normal user account on the system and takes advantage of regular programming mistakes and environment assumptions to exploit a vulnerability and gain root access to the system.

Since most companies take nearly 6 months to detect a data breach, even major ones, researchers find it essential to implement models that detect intrusions for organizations to react promptly. However, having analyzed the different types of attacks, security specialists need to take advantage of the attack class, to gain additional information on the attacker's intentions and the damage done, to make the right decisions immediately (i.e. ignoring, handling with patience, reporting to the government, or even shutting down the whole system). Therefore, our main goal is constructing a network intrusion detection model that is capable of detecting network intrusions providing their class types to stand against attackers and keep the online world safe.

1.2 Challenges

The structure of our model needs to stay limited to operation and budget constraints which we could encounter:

1. Real-Time Capability:
   Most organizations have their networks running full-time, connecting users to services and collecting and uploading data. Thus, a model must be capable of operating in real-time and handling a huge number of connections simultaneously.

2. False Alarm Rate:
   Companies have limited budgets to spend on security, they aim for technologies that can reduce the number of cybersecurity staff employed. Thus, even though detecting network intrusions is necessary for the company, they cannot handle a high number of false alarm rates.

3. Mislabeled Prediction:
   A model mislabeling a serious breach as normal traffic can be fatal for a

business; attacks like Ransomware encrypt the data inside the computer system and threaten businesses to publish or remove it. Encrypted data is not to be recovered in a matter of time, owners must pay a ransom or otherwise important data will be lost forever.

1.3 Contribution

While one-class classifiers have been the promising direction in anomaly detection, these models on their own are limited to binary classification (normal vs intrusive) and lack the benefit of extra information available about the intrusion class. The principal contributions of this thesis are:

1. We introduce a new multi-stage decomposition approach that encounters highly imbalanced multiclass problems via one-class classifiers trained over each class. To do so, we present a fusion strategy that takes full advantage of the strong preliminary one-class classifier predictions for detecting an intrusion, then move beyond to classify intrusions by their corresponding class type.

2. To increase real-time capability, we provide a slightly modified version of the model structure that is capable of operating more efficiently than the standard model at the cost of losing records that specialists might consider as redundant.

3. We use the NSL-KDD dataset to evaluate our classifier branches based on the area under the ROC curve, then provide a comparison between the final model and other multiclass models based on the confusion matrix outcomes.

4. We finally experiment and provide our analysis on the effect of applying an imbalanced learning strategy - SMOTE oversampling, to our model and the multiclass models implemented previously.

1.4 Outline

The rest of the thesis is organized as follows: In Chapter 2, we give a comprehensive summary of highly imbalanced multiclass problems, and a review on the related works of models described for network intrusion detection. In Chapter 3, we explore our dataset features and statistics and share a brief description of the preprocessing applied over the NSL-KKD dataset prior to training. In Chapter 4, we apply preliminary experimentation to gain insight on the model stages, we later discuss in detail our model structure and explain its advantages with respect to other models in performance and facing the challenges we listed earlier. Next, in Chapter 5 we experiment with our final model and compare it to other

multiclass models with and without oversampling. And finally, we conclude and present future directions in Chapter 6.

Chapter 2

Literature Review

In machine learning, a multiclass classification is a problem that extends beyond the classical binary case and requires identifying the belongingness of an instance over a set of three or more classes. For constructing a classifier, a common objective relies upon improving its evaluation measures, which represent error computations resulting from false predictions.

To solve a multiclass problem, two commonly used approaches are present. The first approach deals with all classes present as one problem, by solving the given context without further modification in structure. Training classifiers of this type often perform well in problems where the data is well balanced. However, it was found that overtraining the model seems to favor classes that are more represented in the dataset [6].

Some researches were based on keeping this approach and manipulating the data to abate the class imbalance; SMOTE stands for Synthetic Minority Oversampling Technique, is a method in which the minority class is over-sampled by taking each minority class sample and introducing new synthetic observations along the line segments joining the k nearest neighbors of existing observations [7]. Other methods include undersampling the majority class [8] or increasing the misclassification cost of the minority class [9].

The second approach is to decompose the problem by applying the divide and conquer strategy and is well-suited for classifiers that cannot naturally handle the multiclass situation. This approach transforms the multiclass problem into several binary sub-problems, and tries to classify an observation between each pair of classes separately, and will then integrate all binary results into a final decision. Significant attention was given to this decomposition method and has proven to perform well in most multiclass problems [10].

This second approach however is not without its own limitations. The dependence of the model on the selected fusion strategy that combines the binary classifiers led to high variation in the outcomes, which made the model less reliable. Besides, having a relatively high imbalance in the data remains a problem, since binary classifiers try to find a decision boundary that will minimize the

error on objects from both classes, where one class is severely undersampled and is difficult to define [11].

While binary classifiers may in such situations fail, another approach had the potential to surpass such problems, and despite not using all the knowledge available, the nature of its training phase was able to capture the unique properties of a target class, without overfitting the data. It became known in the literature as one-class classifier. These classifiers were viewed as promising algorithms that can handle data with the overabundance of a common class, or even in the full absence of all other classes.

Keeping in mind that the decomposition approach is a well-established point of reference, a decomposition of one-class classifiers in the multiclass classification context appeared promising. But, to construct a model with the following properties, it is important to compare the different properties of the binary and one-class classifiers and the different ideas of decomposition and fusion strategies applied [12].

In general, given two classes ω_1 and ω_2, a binary classifier labels observations as ω_1 if: $p(x \mid \omega_1)(\omega_1) > p(x \mid \omega_2)(\omega_2)$ and is classified as ω_2 otherwise. However, one-class classifiers ignore class priors by focusing on a single class and constructing a density function upon it. We can later classify instances by introducing a well-tuned threshold τ on the function. Thus, an observation will be classified as ω_1 if $p(x \mid \omega_1) \geq \tau$ [13].

In a more descriptive context, a kernel-based machine learning technique that is widely used is the Support Vector Machines and can be put into example. In binary classification, an SVM model tries to find the best possible space between the two classes ω_1 and ω_2. But in the one-class classification, one-class SVMs will create a boundary around a target class ω_1. Another interesting property about SVMs is that despite the often high classification performance, computational time, and stability to parameter settings, this technique is not designed to handle multiclass problems without applying a decomposition strategy [14].

The most popular decomposition schemes discussed in the literature are the one-versus-one (OVO) and one-versus-all (OVA). OVO strategy considers all possible pairwise combinations; as the number of classes increases, the number of binary classifiers will significantly increase as $N = \frac{M(M-1)}{2}$, where M is the number of classes. The OVA strategy on the other hand specifies one class as the target class or positive class and labels all the remaining classes as the negative class. OVA returns a lower number of binary classifiers when $M > 3$, yet they are more complex. Besides, the OVA approach usually has a relatively low quantity of positively labeled data compared to the total number of all remaining classes defined as negative labels, therefore most classifiers are constructed with an imbalance between the two classes.

While the decomposition part in the strategy is relatively easy, the reconstruction of a final decision is critical. For OVO and OVA, several aggregation strategies rely on voting such as the Decision Directed Acyclic Graph, which

was not only used in binary problems but also modified to deal with one-class problems [10]. The DDAG is an OVO aggregation scheme, it constructs a rooted binary acyclic graph for each node corresponding to assigned class decisions. An observation is evaluated at the decision node to select a decision corresponding to one of the classes and eliminate the others. The DDAG was easily applied for one-class classifiers, by defining one of the classes as a target class, and the second class as the outlier [15].

Another OVO aggregation scheme studied in the literature is pairwise coupling, which is based on the estimation of the joint probabilities of all possible combinations, then selecting the class with the highest posterior probability [16]. Similar aggregation methods can be applied in one-class decomposition problems by using belongingness measures instead of the posterior probabilities.

Adjacent to the OVO schemes, the maximum confidence strategy is an OVA aggregation scheme developed to handle tie situations by selecting the class with the most positive answer and can be applied directly on both binary and one-class problems.

While network intrusion detection is a specific field of anomaly detection, we can still rely on models described for the latter in all of its corresponding applications. Most approaches in the literature try to build a model over the normal data and then evaluate how new data fits into the model. Some of these methods are density-based, which can be simple yet effective; among the OCC density methods are the Gaussian Mixture Models [17] and Parzen Density [18]. Other methods are boundary-based like the Support Vector Classification [19] or reconstructive like the K-means [20].

Several approaches used artificial neural networks to detect anomalies [21], some of these models generalize from previously observed intrusions to recognize unseen attacks with variations to the attacks trained on, applying a classical feed-forward multi-layer neural network [22]. Other approaches apply non-stationary models [23] or decision trees [24] to obtain their model.

On the contrary, another approach applied an isolation forest model that isolates anomalies instead of learning normal data, and results were promising and able to outperform random forest models and local outlier-based models in terms of AUC and computational cost [25].

Out of the multiclass-focused approaches, one approach proposes a Multiple Classifier System that combines several one-class classifiers that are different along with a dynamic weighted average rule over the classifiers calculated at each test sample [26]. Another approach trains one-class classifiers for each class and then a decision function is based on distance measures, thus specific models that are distance-based had to be used, namely the Support Vector Domain Description and the Kernel Weighting by Kernel Principle Component Analysis [27].

From the approaches experimented on using the NSL-KDD dataset we also note the convolution neural networks model [28], and the sparse auto-encoder

with logistic regression for classification [29].

Chapter 3

Experimentation Dataset

The dataset that will be experimented upon is the NSL-KDD dataset, a modified version of the KDD'99. Although the KDD'99 dataset was widely used as one of the few publicly available, a large number of redundant records in the training set was found whereas NSL-KDD is deemed more challenging and able to emphasize differences in performance across various classifiers. It is also clean of redundancies and other errors that had been observed in the original dataset in the past. [30].

3.1 Exploration

The NSL-KDD dataset is composed of 43 features per observation: 41 of the features referring to the traffic input itself, a feature that includes the label (normal and attack including the attack type), and a final score (not to be used in the classification task). The attack types included in the dataset are of four general types: Denial of Service (DoS), Probe, User to Root (U2R), and Remote to Local (R2L), and they represent the main classes of the attacks. These attacks are of different sub-classes (Table 3.1).

Our model will not be concerned about the sub-class types of attacks and the decomposition problem will include the 4 main attack types; obtaining a specific attack type could be very useful, however, training 40 different classes requires an enormous amount of data which is currently not available. In addition, several attack types found in the testing set are not present while training, and others are present with under 10 samples. After aggregating the data, we recognized that the distribution of the attacks is heavily skewed (Table 3.2).

Table 3.1: List of sub-classes present in the data per attack type.

DoS	Probe	U2R	R2L
apache2	ipsweep	buffer_overflow	ftp_write
back	mscan	loadmodule	guess_passwd
land	nmap	perl	httptunnel
neptune	portsweep	ps	imap
mailbomb	saint	rootkit	multihop
pod	satan	sqlattack	named
processtable		xterm	phf
smurf			sendmail
teardrop			Snmpgetattack
udpstorm			spy
worm			snmpguess
			warzclient
			warezmaster
			xlock
			xsnoop

Table 3.2: The relative percentage of attack types found in the listed datasets.

Dataset	Normal	DoS	Probe	U2R	R2L
Training	67343 (53.46%)	45927 (36.46%)	11656 (9.25%)	52 (0.04%)	995 (0.79%)
Testing	9711 (43.08%)	7458 (33.08%)	2421 (10.74%)	67 (0.3%)	2887 (12.81%)

3.2 Preprocessing

The data is free of missing variables and is originally well split for training and testing. The features include 3 Categorical, 6 Binary, 23 Discrete, and 10 Continuous columns. The categorical features are to be encoded into numeric variables using one-hot encoding. We later obtain 122 features.

After normalizing the data, we apply Linear Discriminant Analysis (Fisher Mapping), a supervised feature extraction method to reduce the dimensionality of our dataset while losing the chances of feature interpretation without ignoring any. This method is highly dependent on the number of classes present for classification; the maximum number of features we can obtain is less than the minimum of the number of classes and the number of features by one, and in our case, we find 5 classes suitable. Also, keeping in mind that the one-class classification models are unsupervised, we favor supervised feature extraction

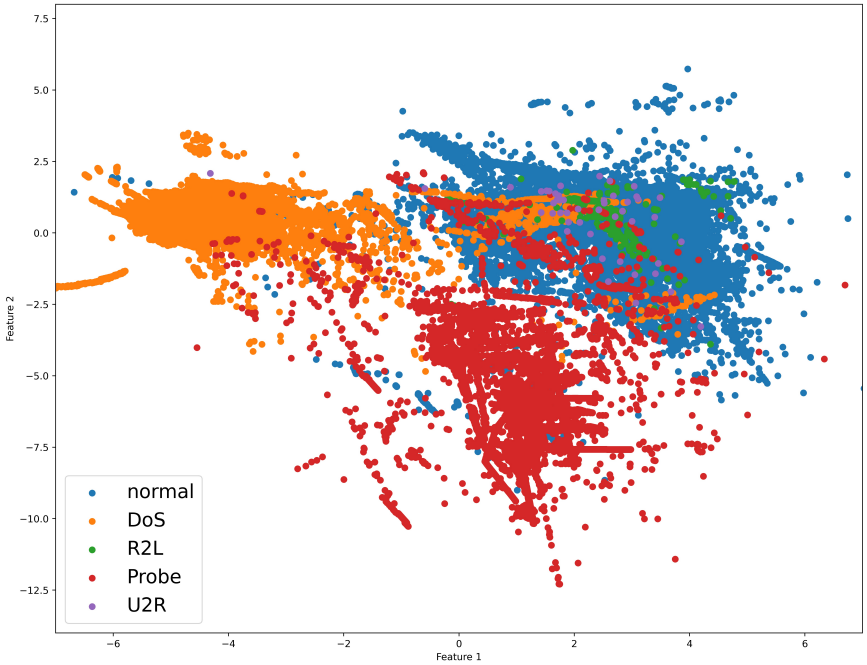


Figure 3.1: *Feautres 1 and 2 of the preprocessed NSL-KDD dataset .*

methods that take advantage of the class labels. Figure 3.1 shows the first two features of the preprocessed dataset.

Chapter 4

Sequential Multi-Stage One-Class Classifier (SMOCC)

4.1 Intuition

The idea was to construct a classifier capable of detecting the attack class in our multiclass context without trivializing the main objective of detecting an intrusion. Motivated by the performance of one-class classifiers in anomaly detection and the novel decomposition methods in the multiclass systems, some of our elements are predetermined - a structure of one-class classifiers.

The selection of the one-class classifier was based on speed, scalability, and performance. One-class SVM, Isolation Forest, Gaussian OCC, and Local Outlier Factor Classifier were on the list:

- Gaussian OCC:
  Although the Gaussian one-class classifier required few parameters and was able to train a relatively significant number of observations in seconds, assuming a gaussian distribution is too restrictive and did not lead to good results.

- Local Outlier Factor:
  The local outlier factor parameters were also few and not too sensitive. It performed well when tested over a small sample of the data within a limited time for training. The LOF cannot handle but small sampled datasets; for the LOF to detect an intrusion, the model needs to be reimplemented for every new observation.

- One-class SVM:
  The One-Class Support Vector Machines classifier is a classifier that performs well if well-tuned. However, choosing the parameters is critical to the performance. Thus, taking longer time to train compared to other classi-

fiers; a hyperparameter tuning stage that falls under a huge grid makes the whole training phase the longest.

- Isolation Forest:
  The Isolation Forest is extremely efficient and fast compared to others, it almost always performs better or is never much behind the best classifier; its parameters have a clear interpretation; moreover, ensemble aggregation makes the classifier stable and robust.

Recall that one-class classifiers performed well in binary classification, constructing a multi-stage fusion structure with a normal-intrusive classifier at the first stage maintains the same accuracy of detecting intrusions with a low false alarm rate, while the following stage/stages are to be concerned with the multiclass attacks. Based on the probabilities of our outcomes, one possible option could be to rely on the maximum confidence strategy [10] to predict an observation attack type (if classified as intrusion).

For the NSL-KDD dataset, the number of samples for every one-class in our training set varies significantly; although one-class classifiers can capture well certain properties of a target class, relying on maximum confidence for a barely represented class is unlikely to succeed.

For the purpose of preliminary experimentation, we apply the isolation forest one-class classifier. Analyzing the results, as the number of samples per class we trained on was less, the resulting AUC score and accuracy (hard labels were based on maximizing the balanced accuracy criterion of the testing set) was worse. The class AUC score decreases from 94% for the normal data to 64% in the R2L intrusion class (Figure 4.1). Designing our fusion, we need to construct a model that relies more on classes highly represented in the dataset.

4.2 Final Model

We introduce a sequential multi-stage one-class classification model composed of a sequence of one-class classifiers arranged according to the empirical class frequencies in the data. We apply novel algorithms used in the literature and assure that anomalies are well-determined. The current stage will result in two different directions, selected according to the classifier predictions. All observations classified as normal shall proceed to label, whereas intrusions will pass through another stage or a sequence of stages to classify its type. The next stage will be a one-class classifier that learns the attack type most commonly represented in the data. According to these last predictions, the anomaly will be classified as the current classification stage label or pass to the next classifier that represents a less common attack type. Figure 4.2 represents a general scheme of the model stages.

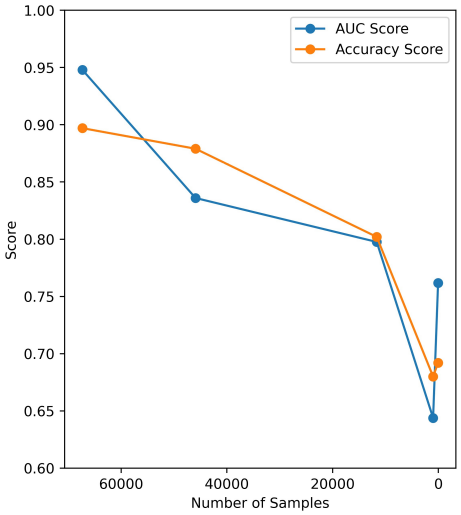


Figure 4.1: *Isolation Forest performance on the NSL-KDD classes of varried training sample sizes .*

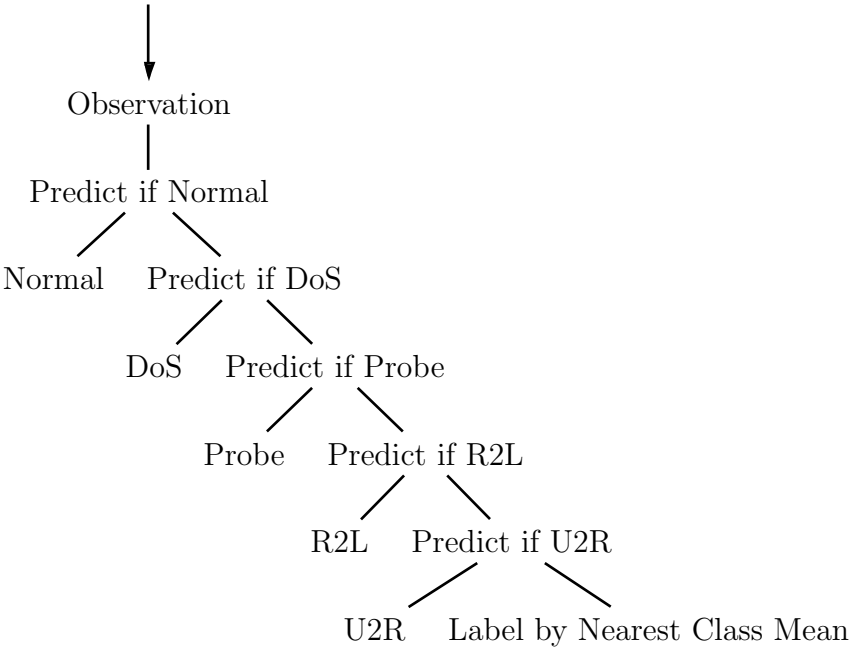


Figure 4.2: *Model Scheme*

4.3 Performance Against Standard Multiclass Classifiers

We test the performance of the SMOCC against standard multiclass classifiers on the NSL-KDD dataset. We implemented Logistic Regression, K-Nearest Neighbors, Linear Discriminant Analysis, Random Forest, Support Vector Machines, and Artificial Neural Networks. We obtain the results of this application in two different scenarios; (i) given the original dataset and (ii) oversampling the data with SMOTE.

4.3.1 Application - Original Dataset

We start by evaluating the one-class classifier's performance per class, we implement an Isolation Forest classifier with stable parameters. For each trained classifier, a receiver-operating characteristic curve (ROC) is constructed over the test set, and the area under the curve is computed (Figure 4.3). An operating point based on maximizing balanced accuracy is selected to showcase the optimal performance possible. Whereas in the final multiclass model evaluation, thresholds are computed using the same strategy on a validation set; these classifiers do not represent the exact predictions used in the model structure elements/events, they are however useful to study their behavior and compare their characteristics when trained on oversampled data. We expect the testing set computed thresholds to be shifted more to the right, favoring anomalies (known as all), the class more represented in each event testing set.

In the normal vs all case, the classifier was capable of capturing most anomalies yet was not perfectly precise; several normal observations were classified as anomalies. Similarly in the DoS class, the Isolation Forest was again well capturing anomalies, while the precision was getting even lower. Moving to a less represented class in the dataset, this time recalling an anomaly from a Probe attack has also decreased compared to the prior, and likewise for the less represented classes, R2L and U2R.

Normal vs All

	precision	recall	f1-score	support
Outlier Class	0.88	0.95	0.91	12833
Inlier Class	0.93	0.83	0.88	9711
accuracy			0.9	22544
macro avg	0.9	0.89	0.9	22544
weighted avg	0.9	0.9	0.9	22544

DoS vs All

	precision	recall	f1-score	support
Outlier Class	0.9	0.97	0.94	15086
Inlier Class	0.94	0.79	0.85	7458
accuracy			0.97	22544
macro avg	0.92	0.88	0.9	22544
weighted avg	0.91	0.91	0.91	22544

Probe vs All

	precision	recall	f1-score	support
Outlier Class	0.96	0.89	0.93	20123
Inlier Class	0.44	0.711	0.54	2421
accuracy			0.87	22544
macro avg	0.7	0.8	0.71	22544
weighted avg	0.91	0.87	0.88	22544

R2L vs All

	precision	recall	f1-score	support
Outlier Class	0.97	0.46	0.63	19657
Inlier Class	0.2	0.9	0.32	2887
accuracy			0.52	22544
macro avg	0.58	0.68	0.47	22544
weighted avg	0.87	0.52	0.59	22544

U2R vs All

	precision	recall	f1-score	support
Outlier Class	1	0.76	0.86	22477
Inlier Class	0.01	0.61	0.01	67
accuracy			0.76	22544
macro avg	0.5	0.68	0.44	22544
weighted avg	1	0.76	0.86	22544

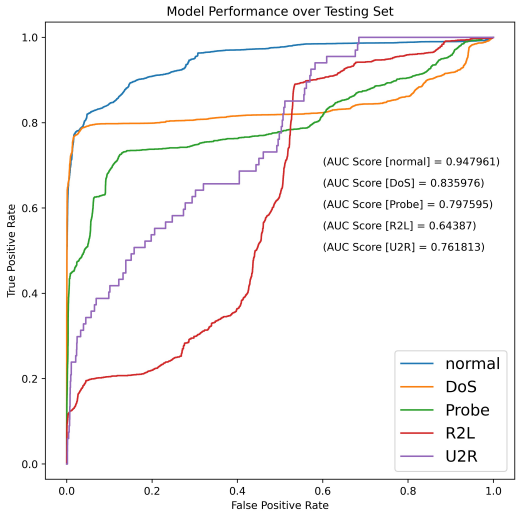


Figure 4.3: *AUROC for the Isolation Forest Classifier over the test set.*

The area under ROC has also decreased. (Figure 4.3) shows the ROC curves for all classifiers.

Next, we implement the sequential multistage one-class classifier along with other standard multiclass classifiers that are hyper-tuned. Table 4.1 includes all confusion matrices. Results were close amongst predictions of classes highly represented. While the Logistic Regression classifier and SMOCC came at the top of predictions for normal observations, thus the lowest false alarm rate, the KNN, and SVM performed best in predicting DoS observations. Probe attacks had also close predictions across classifiers, with SMOCC and Random Forest in the lead. For the two remaining barely represents classes, Logistic Regression, Random Forest, SVM, and Neural Networks failed significantly. Whereas the KNN has predicted few observations of each class, SMOCC, and LDA were the best in capturing these classes; LDA performed better in R2L while SMOCC performed better in U2R.

Comparing mislabeled normal class predictions, a group of classifiers: KNN, Random Forest, SVM, and Neural Networks had the highest number of mislabeled predictions. Other classifiers performed better, while Logistic Regression performed best.

		normal	DoS	Probe	R2L	U2R
SMOCC	normal	9286	85	307	17	16
	DoS	1332	5605	521	0	0
	Probe	532	239	1650	0	0
	R2L	2480	1	18	378	10
	U2R	26	0	0	7	34
		normal	DoS	Probe	R2L	U2R
Logistic Regression	normal	9354	86	263	6	2
	DoS	1352	5603	503	0	0
	Probe	538	279	1604	0	0
	R2L	2881	2	2	2	0
	U2R	42	0	0	1	24
		normal	DoS	Probe	R2L	U2R
KNN	normal	9018	61	629	2	1
	DoS	1236	5689	510	23	0
	Probe	494	295	1626	0	6
	R2L	2557	5	22	300	3
	U2R	41	0	0	6	20
		normal	DoS	Probe	R2L	U2R
LDA	normal	9258	85	334	18	16
	DoS	1328	5605	525	0	0
	Probe	562	228	1631	0	0
	R2L	2335	1	16	525	10
	U2R	26	0	0	8	33
		normal	DoS	Probe	R2L	U2R
Random Forest	normal	9011	58	638	2	2
	DoS	1311	5633	514	0	0
	Probe	460	291	1670	0	0
	R2L	2831	4	4	47	1
	U2R	41	0	0	1	25
		normal	DoS	Probe	R2L	U2R
SVM	normal	9025	61	622	2	1
	DoS	1194	5687	577	0	0
	Probe	485	330	1606	0	0
	R2L	2813	4	23	47	0
	U2R	65	0	0	1	1
		normal	DoS	Probe	R2L	U2R
ANN	normal	9006	56	649	0	0
	DoS	1463	5479	516	0	0
	Probe	628	132	1661	0	0
	R2L	2878	1	8	0	0
	U2R	63	0	4	0	0

Table 4.1: Confusion Matrices

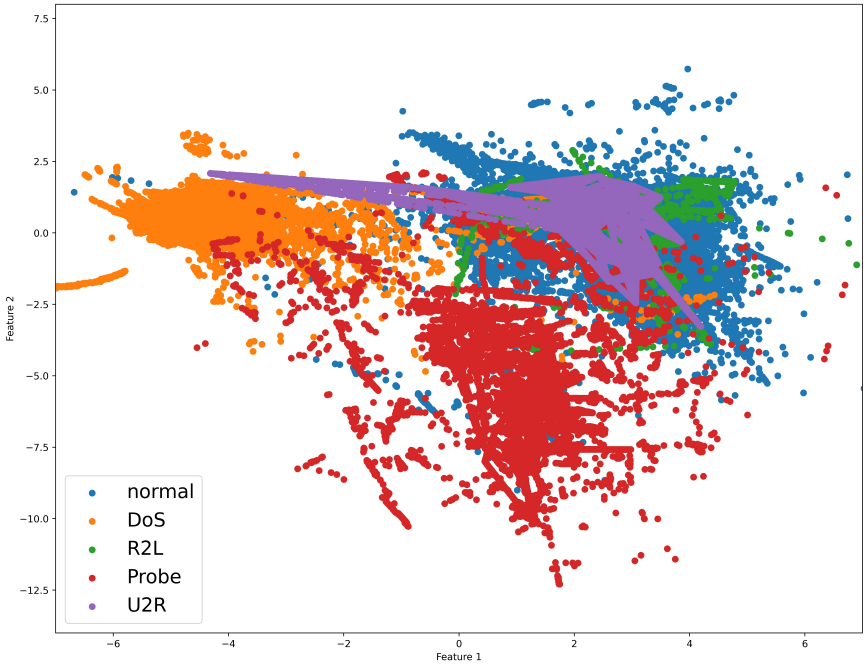


Figure 4.4: *Features 1 and 2 of the oversampled dataset.*

4.3.2 Application - SMOTE Oversampling

Standing on the side of standard multiclass classifiers, we apply SMOTE oversampling to manipulate the class distributions by sampling classes to half the size of the majority class (Figure 4.4). Since imbalanced learning strategies are designed for these classifiers, we expect them to improve. We also experiment with the SMOCC to see if it can be considered as a competitor and a choice for the machine learning community under oversampled preprocessing or balanced class distributions.

Evaluating the one-class classifier's performance per class. The Isolation Forest would still perform well in the normal and DoS classes, while the classification precision has slightly decreased for the Probe class, a slightly oversampled class. For the attacks least represented and most oversampled in the current data, classification precision for each class was almost zero. Figure 4.5 also shows the ROC curves for all classifiers. Comparing figures 4.5 and 4.3, the ROC curves of all classes have shifted slightly downwards after oversampling.

Normal vs All

	precision	recall	f1-score	support
Anomaly	0.88	0.94	0.91	12833
Class	0.92	0.83	0.87	9711
accuracy			0.89	22544
macro avg	0.9	0.89	0.89	22544
weighted avg	0.9	0.89	0.89	22544

DoS vs All

	precision	recall	f1-score	support
Anomaly	0.9	0.98	0.94	15086
Class	0.94	0.78	0.86	7458
accuracy			0.91	22544
macro avg	0.92	0.88	0.9	22544
weighted avg	0.91	0.91	0.91	22544

Probe vs All

	precision	recall	f1-score	support
Anomaly	0.96	0.88	0.92	20123
Class	0.42	0.71	0.53	2421
accuracy			0.86	22544
macro avg	0.69	0.8	0.72	22544
weighted avg	0.9	0.86	0.88	22544

R2L vs All

	precision	recall	f1-score	support
Anomaly	0.97	0.47	0.64	19657
Class	0.2	0.9	0.33	2887
accuracy			0.53	22544
macro avg	0.59	0.68	0.48	22544
weighted avg	0.87	0.53	0.6	22544

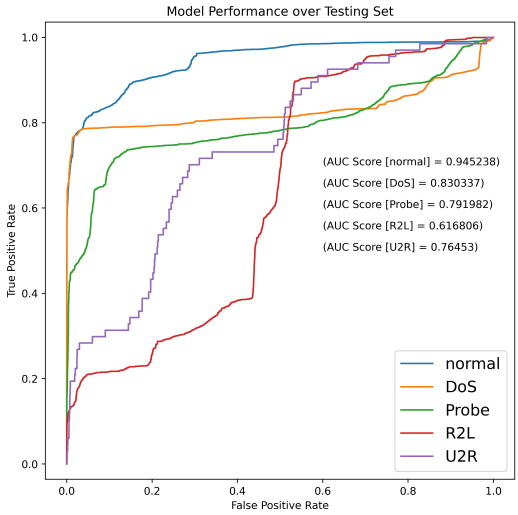


Figure 4.5: *AUROC for the Isolation Forest Classifier over the oversampled test set.*

U2R vs All

	precision	recall	f1-score	support
Anomaly	1	0.72	0.84	22477
Class	0.01	0.72	0.01	67
accuracy			0.72	22544
macro avg	0.5	0.72	0.43	22544
weighted avg	1	0.72	0.83	22544

We again implement the sequential multistage one-class classifier along with other standard multiclass classifiers. Table 4.2 includes all confusion matrices.

The SMOCC classifier has surprisingly improved in capturing attacks that were oversampled, while the false alarm rate has slightly improved; predicting less normal observations. Similar results were obtained by the Logistic Regression classifier representing an improvement over the model previously trained. The KNN and LDA classifiers results were very close to the original dataset results; Probe and R2L predictions have slightly increased while the U2R predictions slightly deteriorated. And for the remaining classifiers: Random forest, SVM, and ANN, we again did not record promising results.

		normal	DoS	Probe	R2L	U2R
SMOCC	normal	8967	86	619	20	19
	DoS	1308	5605	545	0	0
	Probe	442	227	1752	0	0
	R2L	2214	1	20	635	17
	U2R	21	0	0	8	38
		normal	DoS	Probe	R2L	U2R
Logistic Regression	normal	8937	88	613	20	53
	DoS	1326	5592	538	0	2
	Probe	360	209	1844	0	8
	R2L	2138	2	9	705	33
	U2R	20	0	0	7	40
		normal	DoS	Probe	R2L	U2R
KNN	normal	9006	61	632	6	6
	DoS	1181	5681	525	18	53
	Probe	466	288	1640	0	27
	R2L	2293	4	15	375	200
	U2R	36	0	0	14	17
		normal	DoS	Probe	R2L	U2R
LDA	normal	9012	85	577	21	16
	DoS	1326	5605	527	0	0
	Probe	474	228	1719	0	0
	R2L	2208	0	14	657	8
	U2R	30	0	0	8	29
		normal	DoS	Probe	R2L	U2R
Random Forest	normal	9021	58	628	2	2
	DoS	1278	5644	536	0	0
	Probe	465	302	1654	0	0
	R2L	2810	3	4	69	1
	U2R	44	0	0	1	22
		normal	DoS	Probe	R2L	U2R
SVM	normal	9025	61	622	2	1
	DoS	1194	5687	577	0	0
	Probe	485	330	1606	0	0
	R2L	2813	4	23	47	0
	U2R	65	0	0	1	1
		normal	DoS	Probe	R2L	U2R
ANN	normal	8985	56	670	0	0
	DoS	1119	5739	600	0	0
	Probe	447	200	1774	0	0
	R2L	2874	2	11	0	0
	U2R	67	0	0	0	0

Table 4.2: Confusion Matrices - SMOTE Oversampled

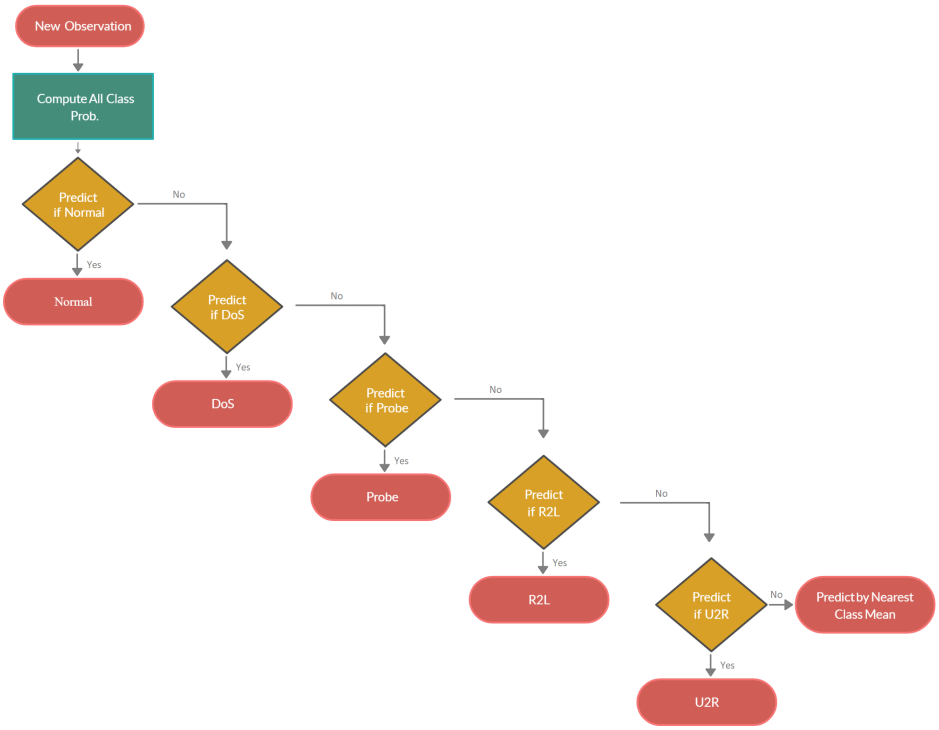


Figure 4.6: *A flow chart of the classifier structure with probabilities predetermined.*

4.4 Improving Real-Time Capability

In contrast to all aggregation schemes discussed in the literature review, we can leverage the sequential model to increase real-time capability; while the DDAG, pairwise coupling, and maximum confidence strategy require all posterior probabilities to function, the sequential structure can skip all remaining events after obtaining a final decision at the stage we are currently on.

The training phase with SMOCC requires less time than an OVO approach with the type of classifier (although in practice regular re-training would be done offline) while the NIDS remains active. The procedure for labeling a new observation in SMOCC is also generally shorter because once a decision is made at any stage, the remaining stages are not pursued.

To design the model described, rather than obtaining all posterior probabilities at an early stage (Figure 4.6), they are calculated within their corresponding event (Figure 4.7). This modification is capable of reducing computational cost significantly, assuming that the distribution of classes is close to the data we studied. The model will however lose data outcomes which for some could be useful for future purposes, while others consider redundant.

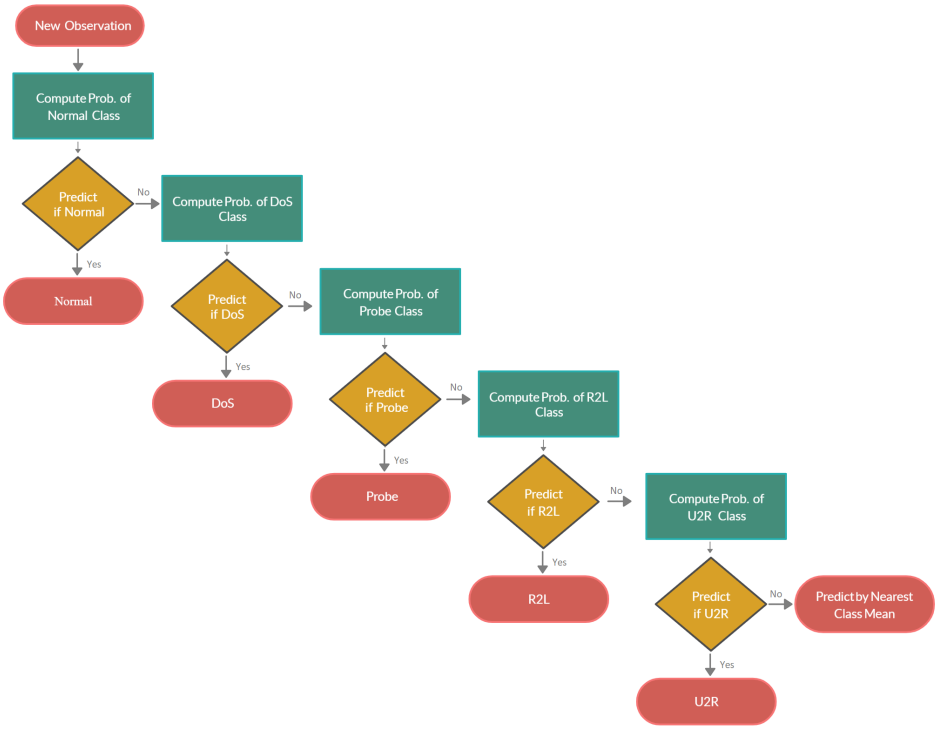


Figure 4.7: *A flow chart of the classifier structure with probabilities calculated per event.*

Chapter 5

Conclusion and Future Work

In this thesis, we propose a sequential multiclass one-class classifier that is capable of transforming one-class classifiers into the world of multiclass classification. The structure of the classifier itself does not require any parameters, as they are purely left to the one-class classifier chosen during implementation. The sequential fusion strategy has also proven to be well suited for skewed class distributions and able to capture classes barely represented in the dataset.

We trained our model on the NSL-KDD dataset and confusion matrices were calculated to compare all the models and approaches. We observed that the SMOCC model was called several times when mentioning the best classifiers of a certain class. It also stood in the group of low false alarms rate and low false predictions of the normal class.

SMOTE oversampling has been an improvement for the standard multiclass classifiers as well as SMOCC, meaning that classes of least observations in the data are still not represented enough for classifiers to capture all their properties.

Finally, the sequential structure can be simply modified to improve real-time capability.

For future work, the sequential multistage classifier is capable of crossing all one-class classification stages without obtaining a final decision. In our model we used a Nearest Class Mean classifier to give direct classification if the sequential model did not predict a class, this classifier is however too restrictive. For the NSL-KDD dataset, only 6.12% of the testing set observations reached this level. For the final stage, multiple classifiers can be utilized within SMOCC: possibly a different one at each stage depending on performance.

We also have few challenges that could be observed: at first, we believe that the classifier is sensitive to the classification threshold obtained by a validation set. Thus, searching for the best strategies of computing thresholds in data with heavily skewed distribution could improve the multiclass model and better leverage the promising one-class classification AUC scores. Second, while SMOTE oversampling has improved the classification scores, further studies could be done to test out several oversampling distributions and find which methods could stand

out.

Bibliography

[1] I. Corporation, "Cost of a data breach report 2021," 2021.

[2] T. C. for Strategic & International Studies (CSIS), "Significant cyber incidents," 2019.

[3] S. Alder, "September 2020 healthcare data breach report," 2021.

[4] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, pp. 222–232, 1987.

[5] D. B. Monowar Hussain Bhuyan and J. Kalita, "Incremental approaches for network anomaly detection: Existing solutions and challenges," *International Journal of Communication Networks and Information Security*, vol. 0, pp. 1–14, 2011.

[6] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *Machine Learning Research*, pp. 1–11, 2003.

[7] L. H. N. Chawla, W. Bowyer and W. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *International Journal of Communication Networks and Information Security*, vol. 0, pp. 1–14, 2011.

[8] P. Ksieniewicz, "Undersampled majority class ensemble for highly imbalanced binary classification," *Proceedings of Machine Learning Research*, pp. 1–13, 2018.

[9] Z. G. N. Thai-Nghe and L. Schmidt-Thieme, "Cost-sensitive learning methods for imbalanced data," *International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2010.

[10] F. H. Bartosz Krawczyk, Michal Wozniak, "On the usefulness of one-class classifier ensembles for decomposition of multi-class problems," *Pattern Recognition*, pp. 3969–3982, 2015.

[11] P. G. G. N. Sanchez-Marono, A. Alonso Betanzos, "Multiclass classifiers vs multiple binary classifiers using filters for feature selection," *IEEE World Congress on Computational Intelligence*, pp. 1–11, 2010.

[12] L. H. M.W. Koch, M.M. Moya, "Cueing, feature discovery, and one-class learning for synthetic aperture radar automatic target recognition," *Neural Networks*, vol. 8, pp. 1081–1102, 1995.

[13] N. J. Colin Bellinger, Shiven Sharma, "One-class versus binary classification: Which and when?," *International Conference on Machine Learning and Applications*, pp. 1–6, 2012.

[14] L. Allwein, E. Schapire, and Y. Singer, "A deep learning approach for intrusion detection using recurrent neural networks," *Machine Learning Research*, vol. 1, pp. 113–141, 2000.

[15] S.-F. T. Kun-Lun Li, Hou-Kuan Huang, "A novel multi-class svm classifier based on ddag," *First International Conference on Machine Learning and Cybernetics*, pp. 1–5, 2002.

[16] R. T. Trevor Hastie, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, pp. 451–471, 1998.

[17] W. H.Zuo, O.Wu, "Recognitionofbluemoviesbyfusionofaudioand video," *Proceedings of 2008 IEEE International Conference on Multimedia and Expo*, pp. 37–40, 2008.

[18] A. G. G. Cohen, H. Sax, "Novelty detection using one-class parzen density estimator.an application to surveillance of nosocomial infections," *Studies in Health Technology Informatics*, vol. 136, pp. 21–26, 2008.

[19] J. S.-T. B. Scholkopf, J. Platt, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 2001.

[20] R. D. A.K. Jain, "Algorithms for clustering data," *Prentice- Hall*, pp. 1–14, 1988.

[21] D. Endler., "Intrusion detection: Applying machine learning to solaris audit data," *Proceedings of the 1998 Annual Computer Security Applications Conference*, pp. 268–279, 1998.

[22] A. S. Anup K. Ghosh, "A study in using neural networks for anomaly and misuse detection," *Proceedings of the 8th USENIX Security Symposium*, pp. 1–12, 1988.

[23] P. C. M. Mahoney, "Learning non-stationary models of normal network traffic for detecting novel attacks," *Proceeding of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 376–385, 2002.

[24] S. J. S. W. Lee, "Data mining approaches for intrusion detection," *Proceedings of the 1998 USENIX Security Symposium*, 1988.

[25] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proceedings of Eighth IEEE International Conference on Data Mining*, pp. 413–422, IEEE Computer Society, 2008.

[26] Y. G. Bilal Hadjadji, Youcef Chibani, "Multiple one-class classifier combination for multi-class classification," *International Conference on Pattern Recognition*, pp. 1–6, 2014.

[27] S. A. Tao Ban, "Implementing multi-class classifiers by one-class classification methods," *International Joint Conference on Neural Networks*, pp. 1–6, 2006.

[28] Y. Z. Yalei Ding, "Intrusion detection system for nsl-kdd dataset using convolutional neural networks," *Procedia Computer Science*, vol. 114, pp. 341–348, 2018.

[29] A. S. Sandeep Gurung, Mirnal Kanti Ghose, "Deep learning approach on network intrusion detection system using nsl-kdd dataset," *Computer Network and Information Security*, vol. 3, pp. 8–14, 2019.

[30] W. L. Mahbod Tavallaee, Ebrahim Bagheri, "A detailed analysis of the kdd cup 99 data set," *IEEE Symposium on Computational Intelligence in Security and Defense Applications*, pp. 1–6, 2009.