Table 1. Symbolic-typed Attributes

Attribute	Symbolic	Numeric value
Protocol_type	UDP	1
	TCP	2
	ICMP	3
Flag	OTH	1
	REJ	2
	RTSO	3
	RTSOSO	4
	RSTR	5
	S0	6
	S1	7
	S2	8
	S3	9
	SF	10
	SH	11
Service	65 values	from 1 to 65

- R2L: Unauthorized access from a remote machine, e.g. guessing password.
- Probing: surveillance and other probing, e.g. port scanning.
- U2R: unauthorized access to local super user (root) privileges, e.g. buffer overflow.

To make the data set simpler, reducing the redundancy without losing the information, we pre-process the data set as follows:

- **Conversion from the Symbolic type to the Numeric type**: there are 3 attributes in the Symbolic manner such as: Protocol, Service, Flag which are needed to be converted to the Numeric type to be compatible with the inputs of the algorithm. The symbolic values are labeled as in Table 1.
- **Normalization**:Normalization of data in the NSL-KDD data set is necessary since there are many big values in comparison with much smaller values in the set. We apply the Min-max normalization method to turn all values to the range [0,1] as follows:

$$\hat{v}_i = \frac{v_i - min(v_i)}{max(v_i) - min(v_i)}, \text{ for i } = 1, 2, \ldots, 41 \qquad (1)$$

where:
v_i: value of one attribute before normalization.
$\hat{v}_i$: value of one attribute after normalization.
$i = 1, ..., 41$: 41 attributes
After normalization, each data sample becomes a 41-attribute vector x_i which is an input for the detection process later. A 41-attribute vector before and after the normalization process can be illustrated in Figure 3 and Figure 4, respectively.

```
0 tcp ftp_data SF 491 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 1 0
0 150 25 0.17 0.03 0.17 0 0 0 0.05 0 normal
```

Figure 3. An example of an original vector x_i.

```
0 1 18 10 491 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 1 0 0 150
25 0.17 0.03 0.17 0 0 0 0.05 0 normal
```

Figure 4. An example of a normalized vector x_i.

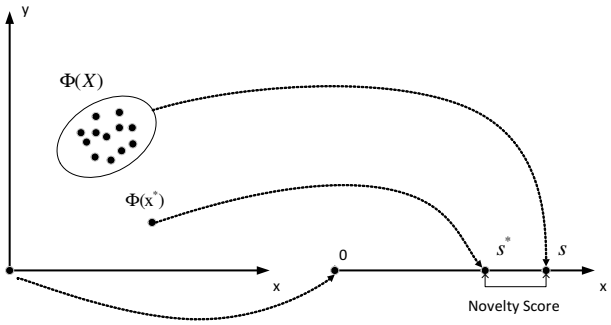

Figure 5. The samples are separated from the origin in the kernel feature space with a mapping Φ, then mapped on a point s, and the novelty score of a testing sample x^* is the distance of its projection s^* to s.

3.3. Control-chart based Kernel Null Space

Before describing EKNS, we briefly re-call the One-Class Classification using Kernel Null Space proposed in [2]. Let us consider a dataset of N training samples $\{x_1, x_2, \ldots, x_N\}$, with each $x_i \in R^D$, and D is the number of observed features. In the one-class setting, all the training samples belong to a single target class. The input features $X = [x_1, x_2, ..., x_N]$ are separated from the origin in the high-dimensional kernel feature space similar to one-class SVM [13]. As described in [2], a single null projection direction is computed to map all samples on a single target value s. A test sample x^* is projected on the null projection direction to obtain the value s^*. Figure 5 illustrates the one-class approach with kernel null space. The novelty score of x^* is the distance between s and s^*:

$$NoveltyScore(x^*) = \mid s - s^* \mid . \qquad (2)$$

A large novelty score indicates more likely novelty. In [2] and [1], a hard decision threshold $\theta_{threshold}$ is used to determine whether the test sample x^* belongs to the target class or not. Determining the threshold plays a very important role to the performance of the novelty detection process. To the best of our knowledge, this threshold has been selected heuristically up till

Table 3. Performance Comparison

σ = 0.5957	Kernel Null Space			OCSVM	Origin Kernel Null Space with fixed threshold=0.05
	q=0.05 $\theta_{threshold}$= 0.0097	q=0.025 $\theta_{threshold}$= 0.0233	q=0.01 $\theta_{threshold}$= 0.0514		
Accuracy	0.9548	0.9598	0.92	0.9445	0.9212
FPR	0.0443	0.018	0.006	0.0433	0.006
Recall	0.954	0.9377	0.846	0.9323	0.8483
AUC	0.9910	0.9910	0.9910	0.9849	0.9910

4.2. Performance analysis

There are some important performance metrics in the novelty (anomaly) detection domain that have been widely used to analyze the performance of a certain detection method. Here, we used confusion matrix for measuring Recall, False positive rate and Accuracy to evaluate detection performance at one value of threshold.

- Accuracy = $\frac{TP+TN}{TP+FP+TN+FN}$
- ReCall-True Positive Rate = $\frac{TP}{TP+FN}$
- FPR - False Positive Rate: $FPR = \frac{FP}{FP+TN}$

Where TP (True Positive) is the number of anomalies correctly diagnosed as anomalies; TN (True negative) is the number of normal events correctly diagnosed as normal; FP (False Positive) is the number of normal events incorrectly diagnosed as anomalies; and FN (False Negative) is the number of anomalies incorrectly diagnosed as normal events.

As mentioned in Section 3.3, we have tested with 3 different q values: 0.01, 0.025 and 0.05. As can be seen in Table 3, with the normalized and pre-processed 41-attribute data set $\{X_1, X_2, \ldots, X_N\}$, the optimal kernel parameter estimated is $\sigma^* = 0.5957$. Subsequently, from the given data set of Novelty scores $\{NS_1, NS_2, \ldots, NS_M\}$, the found threshold is $\theta_{threshold} = 0.0097$, $\theta_{threshold} = 0.0233$, $\theta_{threshold} = 0.0514$ for $q = 0.05$, $q = 0.025$ and $q = 0.001$ respectively.

In the security context, accuracy is more important than recall when you would like to have less False Positives in trade off to have more False Negatives. Therefore, $q = 0.025$ brings best performance in terms of Accuracy, FPR among of the 3 different values q as shown in Table 3.

As another way to evaluate the performance of the detection solution, "ROC-AUC Curve" is often used as a measure of quality of the classification models at various thresholds settings [16]. ROC is a probability curve, it tells how much model is capable of distinguishing classes. This curve depicts relative trade-offs between benefit (TPR) and cost (FPR). To compare classifiers, a common method is to calculate the area under the ROC curve called AUC. AUC stands for "Area under the ROC Curve", represents degree or measure

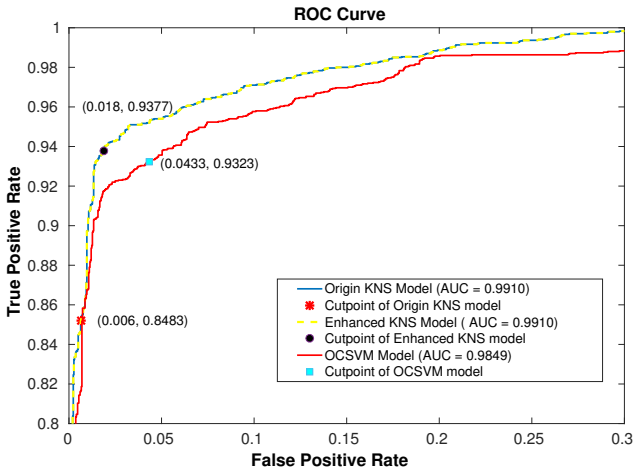

Figure 8. ROC-Curve of the EKNS, original Kernel Null Space, and OCSVM

of separability between two or more different classes. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

In our test, we compare the performance of the EKNS with the original Kernel Null Space in which the threshold is heuristically selected and fixed at 0.05 [2] and with the One Class Support Vector Machine method (OCSVM) [17].

The ROC curves of three models are shown in Figure 8 with the corresponding cutpoints. The cutpoint of the EKNS model with $q = 0.025$ and $\theta_{threshold} = 0.0233$ has a coordinate of (0.018,0.9377), where 0.018 is the false positive rate, 0.9377 is the true positive rate; of the original Kernel null space is (0.006,0.8483); and the ROC cutpoint of the OCSVM method is (0.0433,0.9323).

We can see, the point at (0.018,0.9377) has the highest accuracy and lowest false positive rate as it produces accuracy of 95,98% and closer to the best point in the ROC Space (0,1). This result represents a balance between true positive rate and false positive rate. The AUC value (Area Under the ROC Curve) of the EKNS method (e.g. 0.991) is higher than OCSVM method (e.g. 0.9849), shows that the ability of classification is better.

The obtained results show that; EKNS slightly outperforms the OCSVM and the original Kernel Null Space methods in both terms of Accuracy and AUC while a bit inferior to the Original Kernel Null Space method in terms of FPR.

However, within the security context, accuracy is the more important metric since we desire have less False Positives in trade off to have more False Negatives. In this context, our solution is proved to be slightly better than the competitors.

5. Conclusion and future work

In this research, we have proposed an Intrusion Detection System using the so-called Enhanced Kernel Null Space method - EKNS with data-driven threshold retrieval. The proposed solution with data-driven findings such as $q = 0.025$ and $\sigma = 0.5957$ is proved to outperform the current OCSVM and Original Kernel Null Space methods in terms of detection Accuracy and AUC.

In the future, we would like to address the intrusion detection and the monitoring problem using deep learning, targeting on time series data with uncertainties. We also focus on the detection ability of our proposed approach for large stream data.

Acknowledgements

This paper has been accepted in part to the INISCOM conference 2019 - 5th EAI International Conference on Industrial Networks and Intelligent Systems.

References

[1] Bodesheim, P., Freytag, A., Rodner, E., Denzler, J.: Local novelty detection in multi-class recognition problems. In: Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on. pp. 813–820. IEEE (2015)

[2] Bodesheim, P., Freytag, A., Rodner, E., Kemmler, M., Denzler, J.: Kernel null space methods for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3374–3381 (2013)

[3] Borkar, A., Donode, A., Kumari, A.: A survey on intrusion detection system (ids) and internal intrusion detection and protection system (iidps). In: Inventive Computing and Informatics (ICICI), International Conference on. pp. 949–953. IEEE (2017)

[4] Ferguson, P., Senie, D.: Network ingress filtering: Defeating denial of service attacks that employ ip source address spoofing. In: Internet RFC 2827 (2000)

[5] Gil, T.M., Poleto, M.: Multops: a data-structure for bandwidth attack detection. In: 10th Usenix Security Symposium. p. 2238 (2001)

[6] J. Mirkovic, G.P., Reiher, P.: Attacking ddos at the source. In: 10th IEEE International Conference on Network Protocols (2002)

[7] Liu, J., Lian, Z., Wang, Y., Xiao, J.: Incremental kernel null space discriminant analysis for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 792–800 (2017)

[8] Mercado, G.R., Conerly, M.D., Perry, M.B.: Phase i control chart based on a kernel estimator of the quantile function. Quality and reliability engineering international **27**(8), 1131–1144 (2011)

[9] Nguyen, Q.T., Tran, K.P., Castagliola, P., Truong, T.H., Nguyen, M.K., Lardjane, S.: Nested one-class support vector machines for network intrusion detection. In: 2018 IEEE Seventh International Conference on Communications and Electronics (ICCE). pp. 7–12. IEEE (2018)

[10] Park, Y., Baek, S.H., Kim, S.H., Tsui, K.L.: Statistical process control-based intrusion detection and monitoring. Quality and Reliability Engineering International **30**(2), 257–273 (2014)

[11] Phan Van Trung, Truong Thu Huong, e.a.: A multi-criteria-based ddos attack prevention solution using software defined networking. In: IEEE Inter. Conf. on Advan. Tech. for Commun. pp. 308–313 (2015)

[12] S. Shin, V. Yegneswaran, P.P., Gu, G.: Avant-guard: scalable and vigilant switch flow managament in software-defined networks. In: ACM SIGSAC Conf. Com. Commun. pp. 413–424 (2013)

[13] Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation **13**(7), 1443–1471 (2001)

[14] Sheather, S.J., Marron, J.S.: Kernel quantile estimators. Journal of the American Statistical Association **85**(410), 410–416 (1990)

[15] Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the kdd cup 99 data set. In: Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on. pp. 1–6. IEEE (2009)

[16] T.Fawcett: An introduction to roc analysis. Pattern Recognition Letter **27**, 861–874 (2006)

[17] Trinh, V.V., Tran, K.P., Huong, T.T.: Data driven hyperparameter optimization of one-class support vector machines for anomaly detection in wireless sensor networks. In: 2017 International Conference on Advanced Technologies for Communications (ATC). pp. 6–10 (Oct 2017). doi:10.1109/ATC.2017.8167642

[18] Van Tuyen Dang, Truong Thu Huong, e.a.: Sdn-based syn proxy - a solution to enhance performance of attack mitigation under tcp syn flood. The Computer Journal **62**(4), 518–534 (2019)

[19] Wang, Y., Wong, J., Miner, A.: Anomaly intrusion detection using one class svm. In: Information assurance workshop. pp. 358–364 (2004)