Jurnal Teknik Elektro, Vol. 17, No. 2, September 2024: 55-61
P-ISSN 1411-870X, E-ISSN 2656-3304

PERFORMANCE COMPARISON OF SVM KERNELS FOR INTRUSION DETECTION SYSTEM USING UNSW-NB15 DATASET

Iwan Handoyo Putro
Electrical Engineering Department, Petra Christian University
Siwalankerto 121-131, Surabaya 60236, Indonesia
E-Mail: iwanhp@petra.ac.id

Abstract – Given the proliferation of internet security concerns, the Intrusion Detection System (IDS) has become an essential part of the IoT network. The growing demands for study in the realm of cyberattacks necessitate the availability of datasets. UNSW-NB15 is a publicly accessible security dataset. Since its inception in 2015, numerous researchers have used this dataset to elucidate successful models for threat classification and prediction-based machine learning. Nevertheless, there is a deficiency of research specifically examining the comparison of kernels in relation to the SVM classifier. This paper presents a performance comparison of four SVM kernels. The model's outputs are assessed using execution time and false positive rate, along with four assessment metrics: accuracy, precision, recall, and F1 score. The results demonstrate that the Poly kernel attains the maximum performance, with an accuracy of 98.78%, precision of 97.98%, recall of 98.27%, and an F1 score of 98.12. Nevertheless, the execution duration of the RBF kernel is the most rapid among other SVM kernels, totaling 10 minutes and 23 seconds. Regarding the False Positive Rate (FPR), the Linear kernel exhibits optimal performance at 20%.

Keywords – IDS, UNSW-NB15, SVM kernels, machine learning

I. INTRODUCTION

In the present day, safeguarding personal data from potential attackers is an increasingly critical and challenging endeavour. Conventional strategies such as firewalls and antivirus software are inadequate to address all forms of threats. There is a necessity for supplementary security in conjunction with conventional measures. The Intrusion Detection System (IDS), as shown in Figure 1, is crucial in this context. It meticulously monitors network traffic data and differentiates between normal and malicious activity [1].

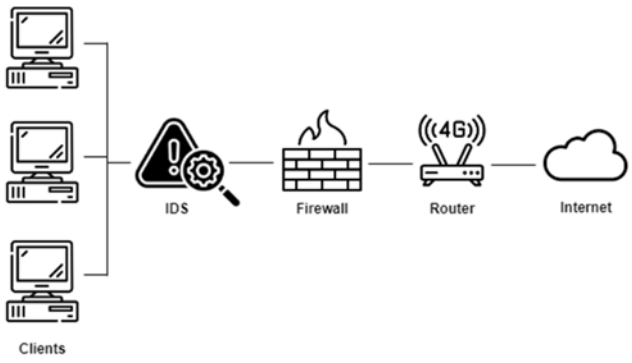

Figure 1. Intrusion Detection System

An Intrusion Detection System (IDS) monitors network traffic to identify malicious behaviour. It can readily identify threats that circumvent the firewall. It perpetually surveils the network, identifies vulnerable segments, and notifies the administrator of intrusions. It can be categorized into two categories such as anomaly detection and signature-based detection [2]. The signature-based detection operates using pre-established patterns of recognized assaults, referred to as signatures. It has great accuracy and low false alarm rates (FAR), although it is incapable of detecting novel attacks.

One method to address this issue is to regularly update the database, which is impractical and an expensive endeavour. As such, anomaly detection algorithms were developed. Anomaly Detection pertains to the analysis of user behaviour profiles. This approach defines a certain model of typical user activity, with any variation from this model classified as anomalous.

Various categories of machine learning (ML) classifiers are employed in literature for intrusion detection [3] [4]. The literature indicates that there is a paucity of research on the comparative study of kernels' comparison inside a machine learning classifier [5] [6]. Therefore, the objective of this research is to conduct a kernel performance comparison of SVM machine learning classifier by utilizing an UNSW-NB15 dataset for intrusion detection.

The paper is organized into five sections. Section 1 addresses the background of this research. Section 2 provides a concise literature review pertinent to this research endeavour. Section 3 explains the methodology adopted for this machine learning based research. Section 4 provides a concise analysis of the results and discussion of the findings. Section 5 presents the conclusion and future work.

II. RELATED WORK

This section provides literature review on research conducted using machine learning classifiers. This section aims to present an overview of the research conducted in the domain of intrusion detection using SVM classifier.

The support vector machine (SVM), a supervised learning method, is frequently employed as a classifier for intrusion detection, particularly in high-dimensional spaces. In our paper, we propose a Support Vector Machine kernels' comparison for intrusion detection system purposes. This section constitutes a literature review of studies employing machine learning classifiers. This section specifically seeks to

present an overview of research undertaken in the field of intrusion detection utilizing the SVM classifier.

The support vector machine (SVM), a supervised learning technique, is commonly utilized as a classifier for intrusion detection, especially in high-dimensional environments. In our research, we present a comparison of Support Vector Machine kernels for the aim of intrusion detection systems. We utilize established SVM kernels to further our understanding of IDS modelling with the UNSW-NB15 dataset.

Due to the varying dimensions of distinct evaluation indicators, which can influence data categorization outcomes, data preprocessing is necessary to mitigate this consequence. The varying sizes of data elements might have a negative impact on the modelling of a dataset. MinMax normalization, a linear scaling method, is frequently employed in machine learning for data feature preprocessing. Nonetheless, MinMax normalization is constrained by its reliance on the maximum and minimum values of the sample data thus limiting its performance on a dataset that has condensed values.

Due to the considerable discrepancies in the values of the UNSW-NB15 dataset, we utilized the standard scaler method to evenly normalize the data across multiple dimensions. This will eliminate biased outcomes in forecasts concerning misclassification error and accuracy rates. This experiment employed frequently SVM kernel functions such as Linear, Radial Basis Function (RBF), Sigmoid, and Polynomial.

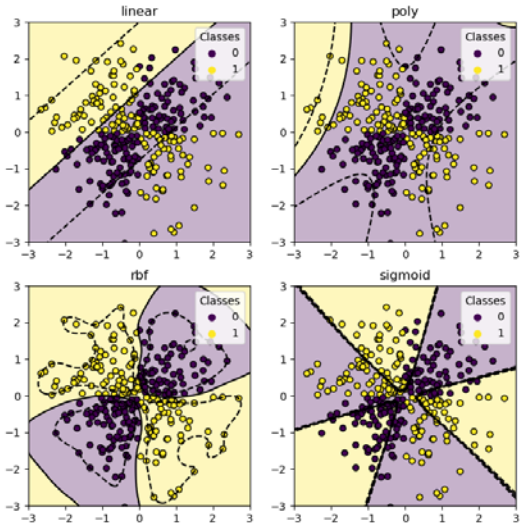

Figure 2. SVM Kernels comparison [7]

Figure 2 illustrates the comparison of the four SVM kernels utilized in this study. The Linear kernel demonstrates a decision boundary characterized by a straight line. This kernel is appropriate for a linearly separable dataset. Moreover, the Linear kernel is a fundamentally straightforward technique that serves as a foundation for comparative analysis.

The second kernel, a polynomial kernel, is represented by a more intricate curve than the linear kernel. This kernel may elucidate a relationship between features in a non-linear fashion, albeit presenting a complex understanding of classification. Nevertheless, this kernel has demonstrated sensitivity to hyperparameter optimization, particularly the gamma parameter.

The Radial Basis Function (RBF), the third kernel utilized in this experiment, is represented by a smooth curve or a series of curves when applied to classification patterns. The RBF kernel is frequently helpful at solving issues involving polynomial patterns. Consequently, it is extensively utilized in addressing challenges in computer vision and picture identification.

The final SVM kernel, Sigmoid, exhibits the classification pattern in a sigmoid form. In contrast to the Linear kernel, the Sigmoid kernel exhibits superior performance with non-linearly separable data. It is also appropriate for use in a dataset that resembles a sigmoid function. Consequently, it is often utilized for neural network applications. However, careful adjustment of its parameters is necessary to attain optimal performance.

III. METHODOLOGY

An essential element when developing an IDS model is the presence of a dataset. Its accessibility allows academics to focus on algorithm building and evaluation. A dataset is essential for IDS research; nevertheless, it is mostly suffered from redundant features, missing values, and data format incompatibility when using ML algorithms.

Ahmad et al. [8] highlight this issue when conducting IDS research using the UNSW-NB15 dataset. Thus, they implement rigorous protocols during the data preprocessing stage of this dataset. This ensures that the dataset is sufficiently prepared for future study.

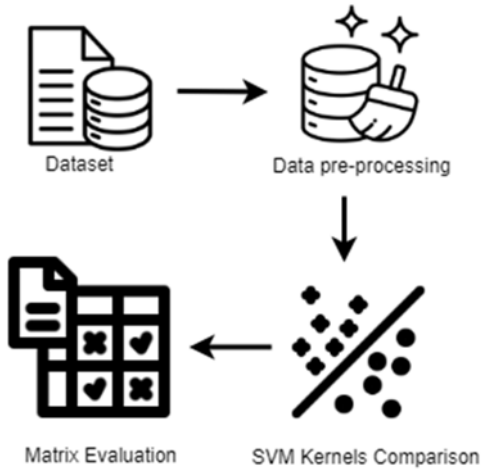

Figure 3. Research Method

Figure 3 illustrates the proposed methodology in this research. The graphic indicates that the dataset utilized in this experiment is UNSW-NB15 (details can be found on sub section A of this part III Methodology). The next step of the proposed experiment includes data pre-processing technique that further explains on sub section B. In this part, data cleaning is conducted which them followed with feature removal and missing value amputation.

After that, a comparison is conducted that utilized four SVM kernels, namely: Linear, RBF, Sigmoid, and Poly. Finally, a matrix evaluation is structured based on assessment aspects, namely: accuracy, precision, recall, F-1 score, and FPR.

A. UNSW-NB15 Dataset

Table 1 illustrates the distribution of all records within the UNSW-NB15 dataset. The primary classifications of the records are normal and attack. The attack records are then categorized into nine types based on the characteristics of the attacks. The details of the records are presented in the UNSW-NB15 dataset paper [9].

Table 1. Description of dataset categories

No	Type	Decsription
1	Normal	Natural transaction data
2	Analysis	An attack to invade web applications through emails, ports, or web scripts
3	Backdoor	A covert attempt to circumvent normal authentication measures or other processes by allowing for secure remote access.
4	DoS	A malicious attempt to disrupt the computer resources by attacking memory.
5	Exploits	An instruction to take advantage of bugs or errors caused by unintentional behaviour on the network.
6	Fuzzers	An attack to crash the system by inputting a lot of random data.
7	Generic	A technique to clash the block-cipher configuration by using hash functions.
8	Reconnaissance	A probe to evade network security controls by collecting relevant information.
9	Shellcode	A piece of code that is executed to exploit software vulnerabilities.
10	Worms	A set of virus code which can add itself to computer system or other programs.

Table 2. List of dataset features

No	Feature's Name	No	Feature's Name
1	srcip	26	res_bdy_len
2	sport	27	Sjit
3	dstip	28	Djit
4	dsport	29	Stime
5	proto	30	Ltime
6	state	31	Sintpkt
7	dur	32	Dintpkt
8	sbytes	33	tcprtt
9	dbytes	34	synack
10	sttl	35	ackdat
11	dttl	36	is_sm_ips_ports
12	sloss	37	ct_state_ttl
13	dloss	38	ct_flw_http_mthd
14	service	39	is_ftp_login
15	Sload	40	ct_ftp_cmd
16	Dload	41	ct_srv_src
17	Spkts	42	ct_srv_dst
18	Dpkts	43	ct_dst_ltm
19	swin	44	ct_src_ltm
20	dwin	45	ct_src_dport_ltm
21	stcpb	46	ct_dst_sport_ltm
22	dtcpb	47	ct_dst_src_ltm
23	smeansz	48	attack_cat
24	dmeansz	49	Label
25	trans_depth		

There are four version of this dataset that available publicly. This dataset now consists of 49 features, while in the earlier version it comprises 44 features. Table 2 indicates all 49 features in the dataset. It can be classfied into information that collected from the packets (Feature 1 to 32), connection related data (Feature 33 to 47), and two feature (48 and 49) as labelled features.

B. Data Pre-processing

Prior to doing model training, the dataset requires cleansing via a pre-processing phase. Two data pre-processing activities are conducted here. Initially, we must define features, as most machine learning classifiers cannot directly manage the training and testing processes due to incompatibility of input types. This dataset contains five features that must be removed due to its notional data type (neither integer nor float). The features include srcip, dstip, proto, state, and service.

Subsequently, we must ascertain if any features have missing values. This is significant since SVM classifier is incapable of processing datasets with missing values. Utilizing the missingno library in Python it is revealed that all features are devoid of missing values, except for ct_flw_http and is_ftp_login. These two characteristics have a significant number of missing values distributed throughout (see Figure 4).

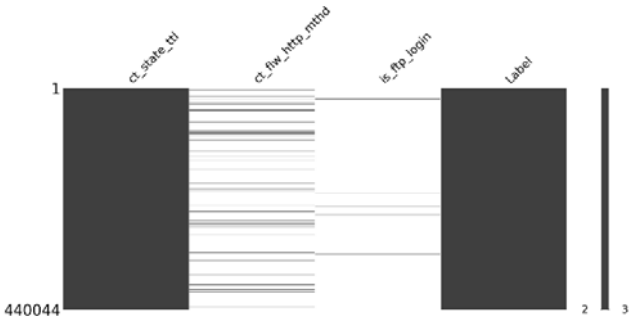

Figure 4. Missing values in the dataset

Two further features, ct_state_ttl and Label, which possess no missing values, are included as references. We had the choice to eliminate these characteristics or perform imputation, and we choose to eliminate these two elements. It is worth noting that Tama et al. [10] and Kumar et al. [11] similarly excluded these two class in their experiment.

C. SVM Kernel

Kernel is a mathematical operation that facilitates data organization and classification. The principal objective of a Support Vector Machine (SVM) is to identify a hyperplane that optimally distinguishes between classes of data points. In practical situations, however, the data is not linearly separable within the original feature space. SVM has a built-in Kernels that facilitate the implicit transformation of the original feature space into a higher-dimensional space, thus potentially enhancing data separability.

Utilizing a kernel function, the data is converted into a new, higher-dimensional space where it may achieve linear separability. In the newly developed feature space, SVM can identify a linear hyperplane that proficiently distinguishes the classes, despite the data appearing non-linear in the original space.

Therefore, when SVM is implemented, it is not necessary to explicitly calculate the mapping to the higher-dimensional feature space. The kernel function evaluates the similarity between data points in a higher-dimensional space without involving the direct computation of each point's coordinates

in that space. This enables SVMs to manage intricate, non-linear associations among features while preserving computing efficiency.
In this experiment we adopted four common SVM kernels, namely: linear, RBP, sigmoid, and poly. The details of these kernels are provided in the previous section, Related Work.

D. Matrix Evaluation

The confusion matrix reflects the accuracy level of a classification model. This is utilized to assess the efficacy of the suggested IDS model. This matrix offers a comprehensive assessment of the classification model's performance, highlighting misclassifications and offering insights for enhancing the model's accuracy.

Table 3. Confusion Matrix

		Predicted Values	
		Attack	**Normal**
Actual Values	**Attack**	TP	FN
	Normal	FP	TN

Table 3 delineates the essential elements of the confusion matrix. A 2 × 2 matrix is utilized for binary classification problems, with these following subsequent interpretations:

- True Positive (TP) signifies that the predictive threat has been accurately detected.
- True Negative (TN) signifies that the predicted normal activity is accurately recognized.
- A False Positive (FP) occurs when an activity is erroneously identified as an attack while being normal traffic.
- A False Negative (FN) occurs when the activity is normal, yet the model erroneously predicts it as a threat.

The metrics derived from the confusion matrix are utilized to compute various model indicators, specifically: accuracy (1), precision (2), recall (3), F1 score (4), and false positive rate (FPR) (5).

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \quad (1)$$

$$Precision = \frac{TP}{TP+FN} \quad (2)$$

$$Recall = \frac{TP}{TP+FP} \quad (3)$$

$$F1\ Score = \frac{2\ x\ Recall\ x\ Precision}{Recall+Precision} \quad (4)$$

$$FPR = \frac{FP}{FP+FN} \quad (5)$$

The FPR score is computed to reflect the model's performance. The elevated rates of true positives (TP) and true negatives (TN), together the diminished rates of false positives (FP) and false negatives (FN), indicate that our model is neither overfitted nor underfitted.

IV. EXPERIMENTAL RESULT AND DISCUSSION

The dataset utilized in this research is UNSW-NB15, supplied by the Intelligent Security Group at UNSW Canberra, Australia. The UNSW-NB15 dataset is advantageous due to its small size, availability in CSV format, and minimal data redundancy. These benefits enable the data pre-processing phase more practicable.
In this experiment purposes, we choose the CSV format. The CSV dataset type includes four versions of the subset. This experiment utilizes the UNSW-NB15_4.csv dataset, which contains 440,044 records. The dataset was subsequently divided into training and testing sets, comprising 70% and 30%, respectively. This dataset is deemed imbalanced, with the target class including 351,149 normal activities and 88,895 attack instances (see Figure 5). Normal record represents by zero vale, while threat labelled as one value.

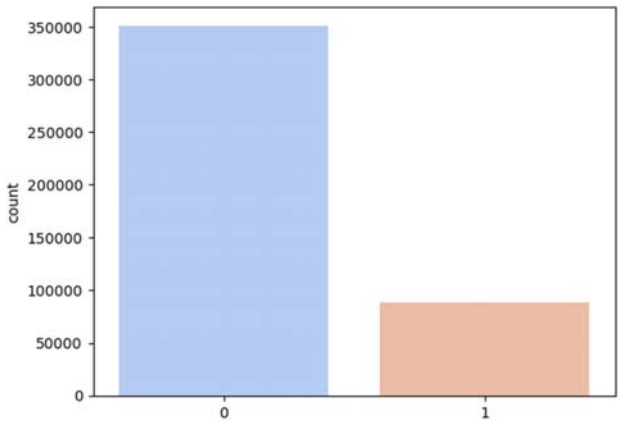

Figure 5. Target class distribution

We utilized the Windows 11 operating system and Python 3, employing various machine learning libraries: Pandas, NumPy, Matplotlib, Seaborn, and Sklearn. The model was executed on a PC with the following specifications: Intel I5-13500 CPU, 32GB DDR5 RAM, 1TB SSD, and Nvidia RTX 3060 graphics card.

A. Distribution of the attacks

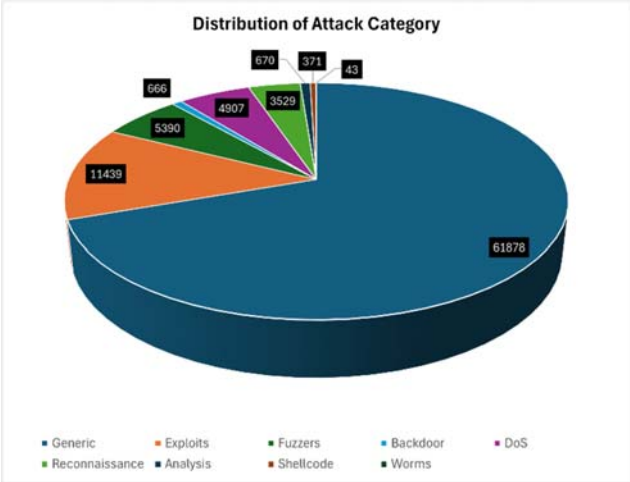

Figure 6. Attack distribution

The distribution of attacks that found in the UNSW-NB15 dataset indicates in Figure 6. The dominant threat is Generic

attacks with can be defined as a penetration attack in which employs techniques effective against all block ciphers (given a specific block and key size) and irrespective of the block cipher's structure.

In contrast, the smallest amount of threat is Worms with only 43 records identified. This Worms attack is a self-replicating malicious code attack that disseminates itself to other computers, primarily over a computer network, without integrating into a software like a virus.

The second highest threat founds is called Exploits as 11,439 records. This Exploits attack is a kind of penetration attack that employs a series of instructions or code to exploit a flaw, defect, or vulnerability in the victims' operating system.

Other threats, which is including Worms (43), are low as under 10,000 records. This includes Fuzzers (5,390), DoS (4,907), Reconnaissance (3,529), Analysis (670), Backdoor (666), and Shellcode (371).

B. Kernels' Comparison Result

The first result of SVM kernels comparison is revealing the execution time needed to each of kernel. The method to obtain this timestamp is gathered by utilizing nbextensions for JupyterNotebook. This extension provides time needed per part of the code executed. To maintain fairness and equity for each of the kernel testing, we reset the JupyterNotebook after each of kernel's iteration.

Table 4. Experiment Result

Kernels	Execution time
RBF	10m 23s
Linear	20m 28s
Sigmoid	1h 48m 47s
Poly	19m 55s

Table 4 shows the comparison of execution time among four kernels implemented in this experiment. The fastest SVM kernel when managing UNSW-NB14 ver. 4 dataset is RBF. It is among the most favoured and utilized kernel functions in support vector machines. It is typically selected for non-linear data. It facilitates appropriate separation in the absence of prior data knowledge.

The second-best kernel, in terms of time execution, is Linear SVM kernel. It is the most fundamental sort of SVM kernel, often widely utilized in a one-dimensional data. The linear kernel is favoured for text classification tasks, as many of these classification issues can be linearly separable.

This kernel demonstrates optimal performance when attributes are present and typically exhibit superior speed compared to alternative kernel functions. In this experiment although it is not achieving the best duration of execution, it is on par with other two kernels: RBF and Poly.

The worst performance in terms of time needed to run separation code based SVM is the Sigmoid kernel. It takes 1 hour and 48 minutes to completely run the code. This is about nine times slower than RBF kernel for the same task.

This result happened as Sigmoid kernel required further computational work compared to Linear, Polynomial, and RBF kernels. It is worth to noting that a fundamental operation distinction exists between the Sigmoid kernel and other the other three kernels. Here, among other SVM kernels, Sigmoid operation involve advanced mathematical processes, including exponential and trigonometric functions, as well as multiplication operations.

The second result of this study is an evaluation matrix as shown in Table 5. This matrix consists of four aspects, namely: accuracy, precision, recall, and F-1 score. It is worth noting that no parameter tuning is conducted during experiment.

Table 5. The evaluation matrix of each SVM kernel

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F-1 score (%)
RBF	98.73	97.84	98.24	98.04
Linear	98.15	96.29	98.17	97.19
Sigmoid	90.46	84.98	85.95	85.45
Poly	98.78	97.98	98.27	98.12

The best model evaluation is achieved by using the Poly kernel. The second-best performance is indicated by RBF kernel. The third best kernel in this experiment is Linear and then followed by Sigmoid kernel.

Poly kernel records the highest evaluation scores in all four aspects with 98.78% of accuracy, 97.98% of precision, 98.27% of recall, and 98.12% of F-1 score. This figure specifies that the accuracy of the model is 98.78%, meaning that the amount of prediction of correct and incorrect is high.

However, as it was stated previously, this UNSW-NB15 is an imbalanced dataset and therefore accuracy itself cannot be solely used as an evaluation. A such, we need to consider the next aspect, precision, which indicates predicted positives that were correctly identified as positive. Again, the Poly kernel result shows that 97.98% of the prediction is correctly done. Checking through the next aspect, recall, it gives a significantly high percentage as 98.27%. It means that the proportion of actual positives that were correctly predicted is remarkable.

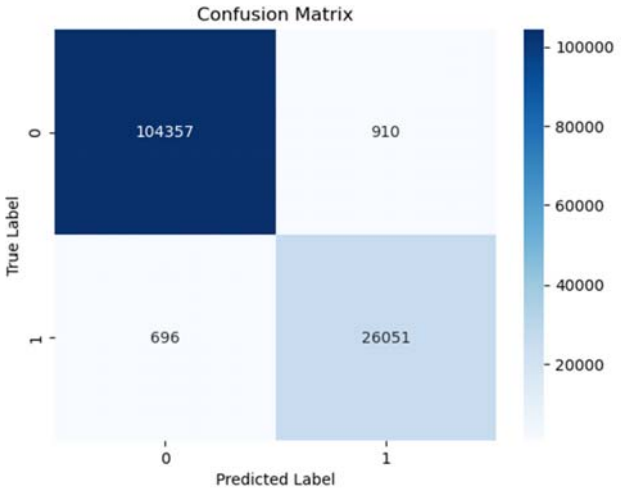

Figure 7. Confusion matrix using Poly kernel

From Figure 7, the confusion matrix of SVM Poly kernel, it can be found that the number of False Negative and False Positive are low as 910 and 696 records. Meanwhile, the number of True Positive and True negative are significantly high as 104,357 and 26,051 records. This indicates the benefit of using Poly kernel when dealing with UNSW-NB15 dataset.

In contrast, Sigmoid, the worst kernel performance in this experiment, show inferior performance with under 90% except for accuracy (90.46%). This needy performance

happened as the result of the dataset distribution that is not resembling of a sigmoid function.

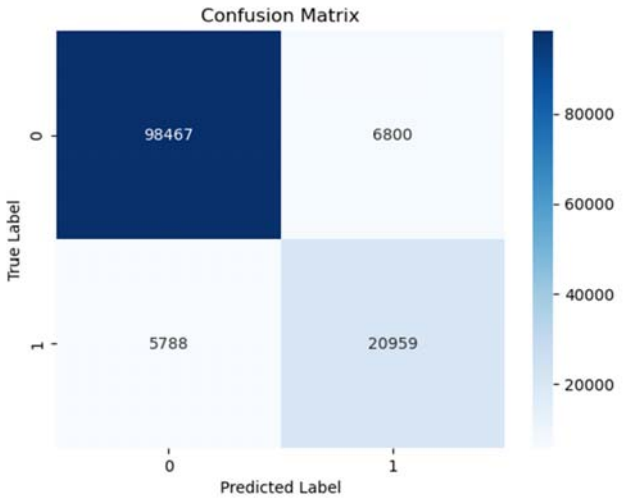

Figure 8. Confusion matrix using Sigmoid kernel

The confusion matrix generated from Sigmoid kernel is pictured in Figure 8. When it is compared with Poly kernel performance, it has a vast difference in terms of TP, TN, FP, and FN numbers. In terms of False prediction, Sigmoid kernel produce significantly high False Positive and False negative figure as 6,800 and 5,788 records, respectively. It is about seven times worst that the Poly performance in the same matrix.

In the IDS domain, although both FP and FN should be minimized, mitigating the FN number is much important than reducing the FP. FN is more perilous, as the principal objective of an Intrusion Detection System (IDS) is to identify and address threats. Therefore, neglecting a genuine threat compromises the system's efficacy.

These low figures both for FP and FN in Figure 8, can be used to indicate the performance gaps between the best and worst of SVM kernel when using UNSW-NB15 dataset.

The last experiment conducted is to measure the FPR for each SVM kernel. The FPR figure is indicated in percentage. The FPR within a confusion matrix domain represents the ratio of false positives (FP) to the total number of actual negative instances. It quantifies the frequency with which the model erroneously classifies a negative instance as positive.

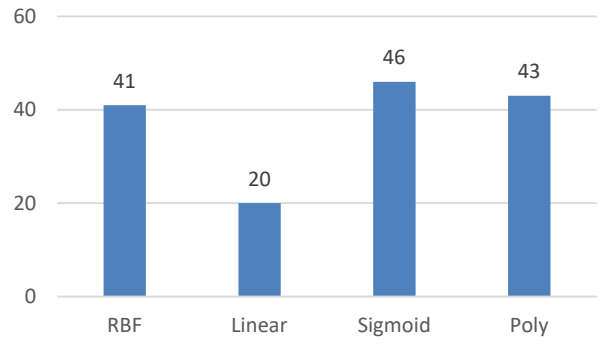

Figure 9. FPR comparison of SVM kernel

Figure 9 depicts the comparison of FPR figures from four SVM kernels utilized in the experiment. The FPR assesses the false alarm rate of a model. In systems such as IDS, a significant false positive rate can result in an abundance of erroneous alerts, potentially inundating analysts and diminishing confidence in the system. Therefore, the lower percentage of FPR indicates better IDS model.

While the Poly kernel provides best result in terms of evaluation matrix, it has significantly high percentage of FPR as 43%. This also happened to two other kernels: Sigmoid and RBF. Interestingly, Linear kernel can achieve lower FPR percentage as 20%. This result indicates that the dataset used in this experiment id considered as non-linear.

V. CONCLUSION

The choice of kernel when utilizing SVM classifier is dataset dependent. Different dataset might be suitable for a particular kernel and tuning parameter.

When using UNSW-NB15 dataset, in terms of matrix evaluation, Poly kernel provides higher number of Accuracy, Precision, Recall, and F-1 score. However, as far as the FPR is concerned, Linear kernel gives the best performance of 20% in which preferred for the IDS domain research.

For further research, parameters tuning, feature selection, and balancing the class should be investigated to further gain better insights into various SVM kernels performance.

REFERENCES

[1] E. Ozdogan, "A Comprehensive Analysis of the Machine Learning Algorithms in IoT IDS Systems," *IEEE Access,* vol. 12, pp. 46785-46811, 2024.

[2] Y. Otoum and A. Nayak, "AS-IDS: Anomaly and Signature Based IDS for the Internet of Things," *Journal of Network and Systems Management,* vol. 29, no. 3, pp. 1-26, 2021.

[3] M. Thankappan, N. Narayanan, M. S. Sanaj, A. Manoj, A. P. Menon, and M. Gokul Krishna, "Machine Learning and Deep Learning Architectures for Intrusion Detection System (IDS): A Survey," presented at the 2024 1st International Conference on Trends in Engineering Systems and Technologies (ICTEST), 2024.

[4] S. Pansare, A. Malik, and I. Batra, "Hybrid Machine Learning Algorithm for Intrusion Detection Systems," presented at the 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE), 2024.

[5] M. A. Almaiah *et al.*, "Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels," *Electronics,* vol. 11, no. 21, 2022.

[6] M. Mohammadi *et al.*, "A comprehensive survey and taxonomy of the SVM-based intrusion detection systems," *Journal of Network and Computer Applications,* vol. 178, 2021.

[7] Scikit-learn. (2024). *Plot classification boundaries with different SVM Kernels.* Available: https://scikit-learn.org/1.5/auto_examples/svm/plot_svm_kernels.html

[8] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-

set," *EURASIP Journal on Wireless Communications and Networking,* vol. 2021, no. 10, pp. 1-23, 2021.

[9] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data set for Network Intrusion Detection systems (UNSW-NB15 Network Data Set)," in *Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1-6.

[10] B. A. Tama, M. Comuzzi, and K.-H. Rhee, "TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System," *IEEE Access,* vol. 7, pp. 94497-94507, 2019.

[11] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Computing,* vol. 23, no. 2, pp. 1397-1418, 2019.