Original Research Article

Evaluation of Machine Learning Model for Network Anomaly Detection: Support Vector Machine

ABSTRACT

Effective network anomaly detection plays a pivotal role in safeguarding digital assets against evolving cyber threats in cybersecurity. This study aims to evaluate the performance of Support Vector Machine (SVM) models with different kernel types for network anomaly detection. Leveraging the neural structured learning- knowledge discovery in database (NSL-KDD) dataset, the research investigates SVM models employing linear, polynomial, radial basis function (RBF), and sigmoid kernels. The study employs experimentation methodologies, including data preprocessing, model training, and evaluation, to assess each SVM kernel's accuracy and F1 score. The findings reveal varying performances across kernel types, with linear and polynomial kernels exhibiting superior accuracy and F1 scores compared to RBF and sigmoid kernels. These results shed more light on the effectiveness of SVM kernels in detecting network anomalies and provide insights for future research and development in cybersecurity.

Keywords: Network Anomaly Detection, Support Vector Machine, SVM Kernels, NSL-KDD Dataset, Cybersecurity.

1. INTRODUCTION

In today's interconnected world, where data communication plays a pivotal role in every aspect of our lives, ensuring the security and integrity of computer networks has become paramount. With the exponential growth of network traffic and the increasing sophistication of cyber threats, the need for effective intrusion detection systems (IDS) has never been greater. An intrusion detection system is a crucial component of network security, tasked with identifying and mitigating malicious activities and unauthorized access attempts within a network environment [1].

Intrusion detection systems have traditionally relied on rule-based techniques and signature-based detection methods to identify known patterns of malicious behaviours[2]. While these methods have been somewhat effective, they often struggle to detect novel and previously unseen attacks. As cyber threats evolve and become more sophisticated, there is a growing need for intrusion detection systems that can adapt and learn from the ever-changing network landscape. Hence, Machine learning (ML) techniques have emerged as promising tools for network anomaly detection, offering the potential to detect previously unseen attacks and adapt to changing threat scenarios. Among the various ML algorithms, Support Vector Machine (SVM) has garnered significant attention for its effectiveness in classifying data points into different categories based on their features [3]. SVM is particularly well-

suited for intrusion detection tasks due to its ability to handle high-dimensional data and its robustness against overfitting.

This research work aimedto evaluate the performance of Support Vector Machine models for network anomaly detection. By leveraging SVM's capabilities in identifying patterns and anomalies within network traffic data, thereby seeking to assess its effectiveness in detecting various types of cyber threats and distinguishing between normal and malicious network activities [4]. Through rigorous experimentation and analysis of different kernels, this work intends to gain insights into the strengths and limitations of SVM-based intrusion detection systems and provide valuable guidance for developingmore robust and reliable network security solutions.

TheNeural Structured LearningKnowledge Discoveryin Database (NSL-KDD) dataset, short for "NSL-KDD Network Intrusion Detection Dataset," is a widely used benchmark dataset in the field of network intrusion detection [2]. It serves as a standard reference for evaluating the performance of machine learning models in detecting various types of network attacks and anomalies. Originally derived from the KDD Cup 1999 dataset, the NSL-KDD dataset addresses some of the limitations and biases present in the original dataset, making it more suitable for modern intrusion detection research. The dataset contains a comprehensive collection of network traffic data captured from a simulated computer network environment, encompassing normal and malicious activities [5].

There has been a lotof research in recent times in the area of machine learning-based intrusion detection. Divekar et al [6] explored the vulnerability of Internet of Things (IoT) systems to cyberattacks due to changes and advancements in the IoT environment. The work highlighted the potential harm to physical and business assets resulting from these attacks. To address these security concerns, the authors employed various machine learning (ML) approaches, including Logistic Regression, Decision Tree, K-nearest Neighbor (KNN), Random Forest, and Support Vector Machine. Using the NSL-KDD dataset, these ML approaches were compared based on evaluation measures such as accuracy, precision, F1-score, and recall to predict and visualize attack threats effectively.

In [7], eight machine learning (ML) techniques were applied to detect intrusions, including neural networks, KNN, SVM, random forest, trees, AdaBoost, naive Bayes, and stochastic gradient descent (SGD). Using the NSL-KDD dataset, these ML techniques were trained and tested to classify network and operating system records into one of 24 possible attacks. The performances of these ML methods were analyzed and compared, with random forest achieving the highest performance. This study investigates more than four ML classifiers on this dataset, utilizing the same set of tools.

A machine learning-based methodology was proposed for detecting intrusions in computer networks [8]. The methodology consists of four main phases: preprocessing, feature selection, parameter optimization, and classification. Correlation Based Feature Selection was used to select the most significant features. For classification, Random Tree, AdaBoost, K-Nearest Neighbor (KNN), and Support Vector Machine (SVM) algorithms were employed, while particle swarm optimization is utilized for parameter optimization. The proposed method was evaluated on two extensive datasets, namely NSL-KDD and CIC-DDOS2019, to assess its effectiveness in intrusion detection.

Sekhar C, Pavani K, and Rao MS [9]focused on the potential of Software Defined Network (SDN) as the next-generation network architecture and addresses security concerns for its large-scale deployment. To enhance the accuracy of Network Intrusion Detection Systems (NIDS), the paper applied various Machine Learning (ML) and Deep Learning (DL) models.

The NSL-KDD dataset was used to evaluate the performance of these algorithms and through extensive experiments, the paper demonstrates that the F-measure rate can reach up to 87.72% for multiclass labels on the NSL-KDD dataset with twenty-two features using the K-Nearest Neighbors (KNN) algorithm. Furthermore, the k-Nearest Neighbor model outperformed other ML models in multiclass classification, as observed in numerous experiments.

A novel approach that integrates feature selection and classification for the NSL-KDD Cup 99 intrusion detection dataset using Support Vector Machine (SVM) was proposed in [10]. The goal was to enhance intrusion classification efficiency by employing a reduced set of input features extracted from the training data through feature selection, a crucial step in supervised learning that involves identifying important input features while eliminating irrelevant ones to improve classification accuracy. In this research, the SVM classifier was applied to various subsets of input features from the training dataset of NSL-KDD Cup 99 to assess the effectiveness of the proposed approach.

2. REVIEW OF RELATED WORKS

Anomalies detection is an important issue that has been investigated in various fields of studies. These anomalies are the patterns in data that do not conform to a well-defined notion of normal behaviour or might be induced in the data for a variety of reasons, such as malicious activity, for example, credit card fraud, cyber-intrusion, terrorist activity, or breakdown of a system.Many Intrusion Detection Systems (IDS) have been used to detect attacks and unauthorized access to networks and their resources. Approaches put forward by researchers ranged from traditional statistical methods to Artificial Intelligence (AI) based approaches, with the AI-based techniques gaining an edge over the statistical techniques in the research community due to the ability of its procedures to be designed to display behaviour learnedfrom previous experiences [11].

As far back as 1999 [12] described how thevarious machine learning approachescan be used to create policies to detect possible intrusions in networks. A real-time anomaly detection algorithm withan "earliness" measure was presented in [13] and [14] presented anomaly detection schemes (ADS), that have applied SVM for intrusion and security attack detection. The paper discussed the concepts of SVM classifiers and intrusion detection systems andspecified the primary capabilities, possible limitations, and advantages of the ADS approaches. Reference [15] however argued that the existing SVM-based techniques with the training features cannot efficiently detect short-duration intrusions and attacks in the traffic and rather proposed an anomaly-based SVM detection scheme by extracting and optimizing the training featureswith Kullback-Leibler (KL) divergence and cross-correlation calculated by the control and data planes traffic to effectively enhance the detection accuracy. A researcher [16] proposed a deep learning-based solution called the log-cosh variational autoencoder (LVAE) to address challenges faced by traditional methods in detecting unknown attacks. The LVAE inherits the strong modelling abilities of variational autoencoders (VAE), enabling it to understand complex data distributions and generate reconstructed data. To better simulate discrete features of real attacks and generate unknown types of attacks, they introduced an effective reconstruction loss term utilizing the logarithmic hyperbolic cosine (log-cosh) function in the LVAE. Compared to conventional VAEs, the LVAE showed promising potential in generating data that closely resemble unknown attacks. The research employed eight feature extraction and classification techniques to classify the generated unknown data. Numerous experiments were conducted using the latest CICIDS2017 dataset, training with varying amounts of real and unknown-type attacks. The experimental results surpassed several state-of-the-art techniques, achieving accuracy and average F1 scores of 99.89% and 99.83%, respectively.This work

analyses the performances of different kernels in SVM and their suitability in network anomaly detection.

3. METHODOLOGY

3.1 Data Collection and Preprocessing

The study utilized the Google Colab notebook environment with Python programming language, focusing exclusively on the NSL KDD dataset sourced from Kaggle. This dataset, tailored for intrusion detection studies, encompasses diverse features such as protocol types, service types, and connection attributes. Prior to model training, rigorous preprocessing steps were undertaken, including data cleaning, feature selection, and normalization, to ensure data integrity and consistency.

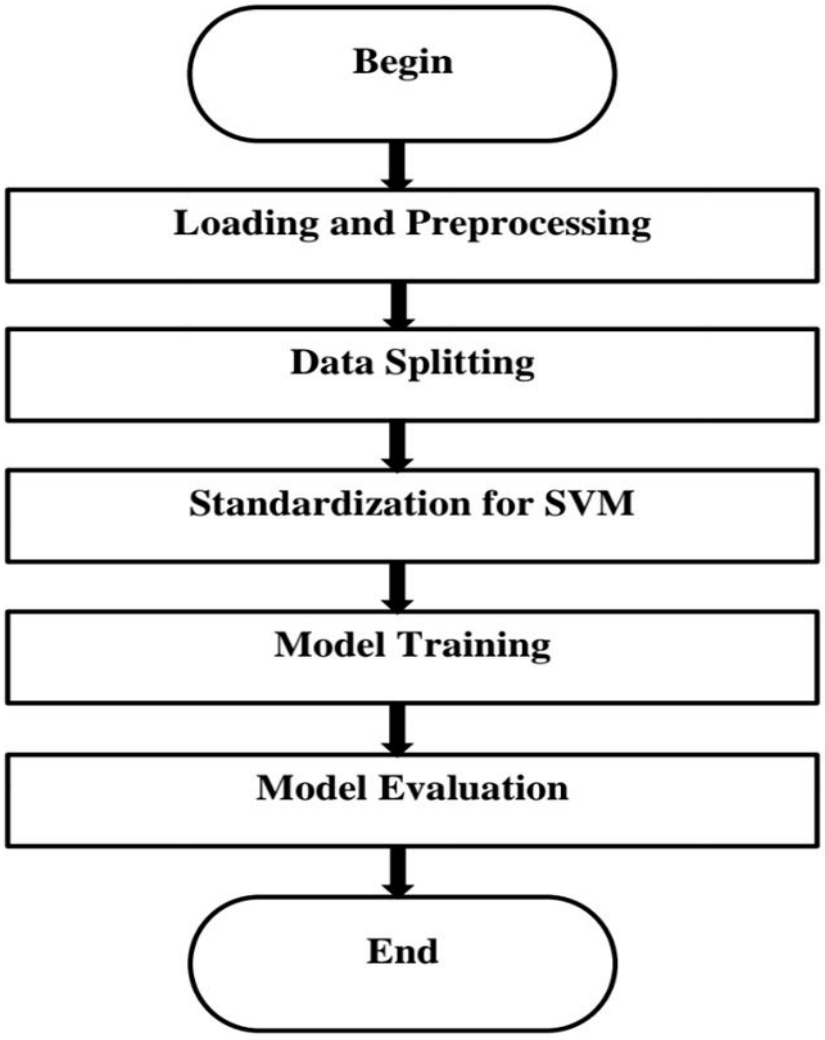

Fig 1. Flow Chart of the methodology

3.2 Performance Comparison

A comparative analysis of SVM models with different kernels was conducted, focusing on their accuracy and F1-score. This analysis will enable the discernment of the relative performance of each kernel in accurately identifying cyber threats and classifying network traffic.

In Support Vector Machine (SVM), kernels are essential components that enable the algorithm to handle non-linear classification tasks effectively. Kernels function by transforming the input data from the original feature space into a higher-dimensional space, where it becomes easier to separate classes with a hyperplane [17]. This transformation allows SVM to capture complex relationships in the data and create non-linear decision boundaries. The types of kernels used in SVM and considered in the work are discussed in the following subsections:

3.2.1 Linear kernel

The linear kernel is the simplest and most used kernel in SVM. It computes the dot product between feature vectors in the original input space, resulting in a linear decision boundary. The decision boundary separates classes by a straight line or hyperplane. Linear kernels are suitable for datasets where classes can be effectively separated by a linear boundary.

3.2.2 Polynomial kernel

The polynomial kernel introduces non-linear decision boundaries by computing the dot product raised to a specified power between feature vectors [18]. This allows SVM to capture more complex relationships in the data that cannot be separated by straight lines. The degree parameter controls the degree of the polynomial, influencing the flexibility of the decision boundary. Higher degrees result in more complex decision boundaries.

3.2.3 Radial basis function (RBF) kernel

The RBF kernel, also known as the Gaussian kernel, measures the similarity between data points based on their Euclidean distance in the original feature space. It maps the data into a higher-dimensional space using a Gaussian similarity measure [19]. The RBF kernel is highly flexible and can capture complex non-linear relationships in the data. It can createdecision boundaries of varying shapes and complexities, making it suitable for a wide range of datasets. The parameters of the RBF kernel, such as gamma (γ) and C, control the smoothness of the decision boundary and the trade-off between model complexity and accuracy.

3.2.4 Sigmoid Kernel

The sigmoid kernel computes the similarity between data points using the hyperbolic tangent function. It maps the data into a higher-dimensional space, allowing SVM to handle non-linear relationships [18]. However, sigmoid kernels are less commonly used compared to linear, polynomial, and RBF kernels. They are suitable for specific datasets with non-linear relationships, but generally may not perform as well as other kernels.

Fig. 2 shows the simulated map from synthetic data of the learning behaviourfor linear, polynomial, RBF and Sigmoid SVM kernels.The maps illustrate the decision boundaries created by different SVM kernels when applied to a synthetic dataset. The synthetic data consists of two classes, represented by different colours distributed in various patterns to highlight the unique capabilities of each kernel.

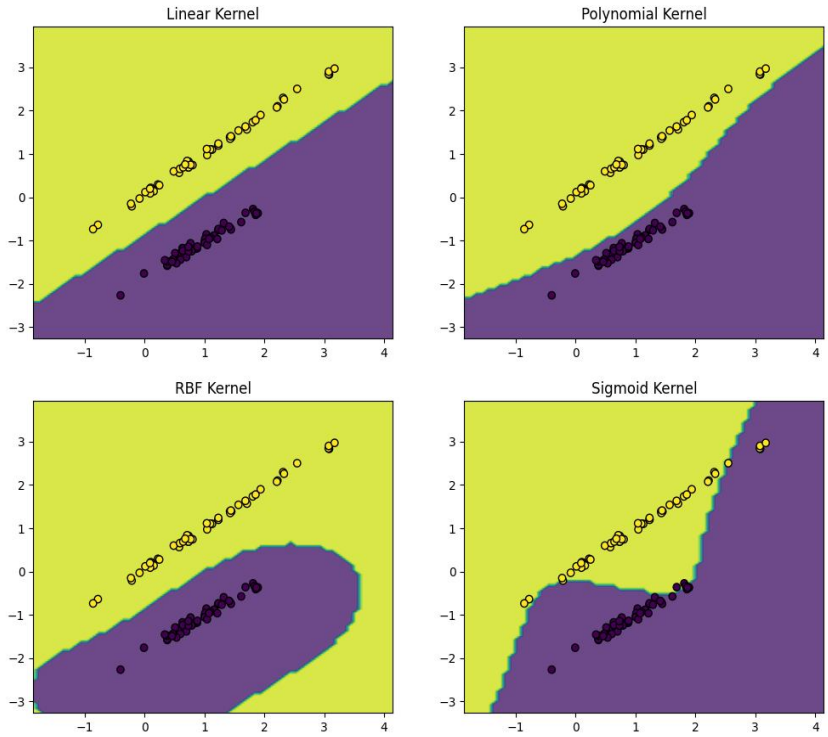

Fig.2. Kernels in Support Vector Machine [Author]

3.3Model Training and Evaluation

The scikit-learn (Sklearn) library is used to facilitate the training of Support Vector Machine (SVM) models with four distinct kernels: linear, polynomial, radial basis function (RBF), and sigmoid. Through stratified cross-validation, the dataset was divided into training and testing sets to maintain class the balance. Subsequently, SVM models were trained on the training data and evaluated using standard performance metrics, namely accuracy and F1-score, to gauge their efficacy in detecting network anomalies and classifying network traffic.

3.3.1 Metrics

Since the data may only have one of four statuses, true or false, positive or negative, T_p was used to connote the true positive values which is the number of instances correctly classified as positive, F_p used to represent the false positive values which is the number of instances incorrectly classified as positive, T_N denoted true negative values which states the number of instances correctly classified as negative and F_N deployed to signify the false negative valuewhich is the number of instances incorrectly classified as negative. The precision, sensitivity, F1 score and accuracy were calculated for the different SVM kernels using Equations 1-4 [20].

The precision or measure of the proportion of correctly predicted positive instances out of all instances predicted as positive was calculated using Equation1.

$$Precision = T_p/(T_p + F_p) \quad (1)$$

The proportion of correctly predicted positive instances out of all actual positive instances is known as the sensitivity or recall. It was determined using Equation 2.

$$Recall = T_p/(T_p + F_N) \quad (2)$$

F1 score or F-measurement is the harmonic mean of precision and recall. It provides a balance between precision and recall and was calculated using Equation 3.

$$F1 - Score = 2\left[\frac{(Precision \times Recall)}{(Precision + Recall)}\right] \quad (3)$$

The accuracy measures the proportion of correctly classified instances (both positive and negative) out of all instances. It is calculated as shown in Equations 4.

$$\boldsymbol{Accuracy = (T_P + T_N)/(T_P + F_P + T_N + F_N)} \quad \boldsymbol{(4)}$$

The Macro average which calculates the average performance metric, such as precision, recall, or F1 score, across different classes independentlyand the Weighted average which considersthe average performance metric by assigning specific weights to each class based on their importance. They were calculated using Equations 5 and 6 [7].

$$MacroX = \frac{1}{N} \times \sum X_i \quad (5)$$

$$Weighted\_X = \frac{\sum(x_i \times c_i)}{\sum c_i} \quad (6)$$

Where N is the number of classes, C is the number of instances in the class and X represent the precision metric under consideration such as Precision, Recall, and F1 Score.

3.3.2 Algorithm

Model training and evaluation was achieved using Algorithm 1.

Algorithm 1: Model Training and Evaluation
1. Start
2. Load dataset from Google Drive df
3. Encode_Categorical_Features(df)
4. Split data into training and testing sets: X_train, X_test, y_train, y_test
5. Standardize features: X_train_scaled,)
6. Define kernel type= ['linear', 'poly', 'rbf', 'sigmoid']
7. f_scores = []
8. accuracies = []
9. Train and evaluate SVM model for each kernel type
10. FOR EACH kernel IN kernel_types:
svm_classifier = Initialize_SVM(kernel)
svm_classifier.fit(X_train_scaled, y_train)

11. Make predictions
12. Evaluate the model:
f_score
accuracy
13. Store results
14. Print results
15. Plot_F_Score_Comparison
16. Plot_Accuracy_Comparison
17. END

Fig. 3 shows the simulated heat map generated. Heatmaps are powerful tools in data visualization, offering a comprehensive representation of relationships within a dataset. They

excel in showcasing patterns, correlations, and variations in numerical data. One key application is visualizing correlation matrices, where each cell represents the correlation coefficient between two variables [21]. The colour gradient indicates the strength and direction of the correlation, facilitating quick identification of positive, negative, or neutral correlations. Beyond correlation, heatmaps help identify clusters and patterns within the data. Similar values form cohesive color patterns, aiding in the recognition of groups or trends that might be less apparent in tabular data. They are particularly useful for highlighting anomalies or outliers, as these deviations from established patterns become visually evident. Heatmaps also prove effective in comparing multivariate data. By visualizing multiple variables simultaneously, they offer a holistic view of interactions. This is beneficial in handling complex datasets where understanding relationships between multiple variables is crucial. The visual representation through color gradients enhances data interpretation, making it more intuitive. Heatmaps are customizable, allowing users to adjust color palettes, labels, and annotations to align with specific analytical goals. This flexibility ensures that the visualization caters to the preferences and needs of the data analyst or audience. In the context of machine learning, heatmaps are valuable for feature selection and model evaluation. They assist in identifying features with high or low importance, contributing to the optimization of model performance. In summary, heatmaps are indispensable for exploring, understanding, and communicating complex relationships within datasets. Whether used for correlation analysis, anomaly detection, or multivariate comparison, heatmaps empower data scientists, analysts, and stakeholders by providing a richer and more intuitive interpretation of data.

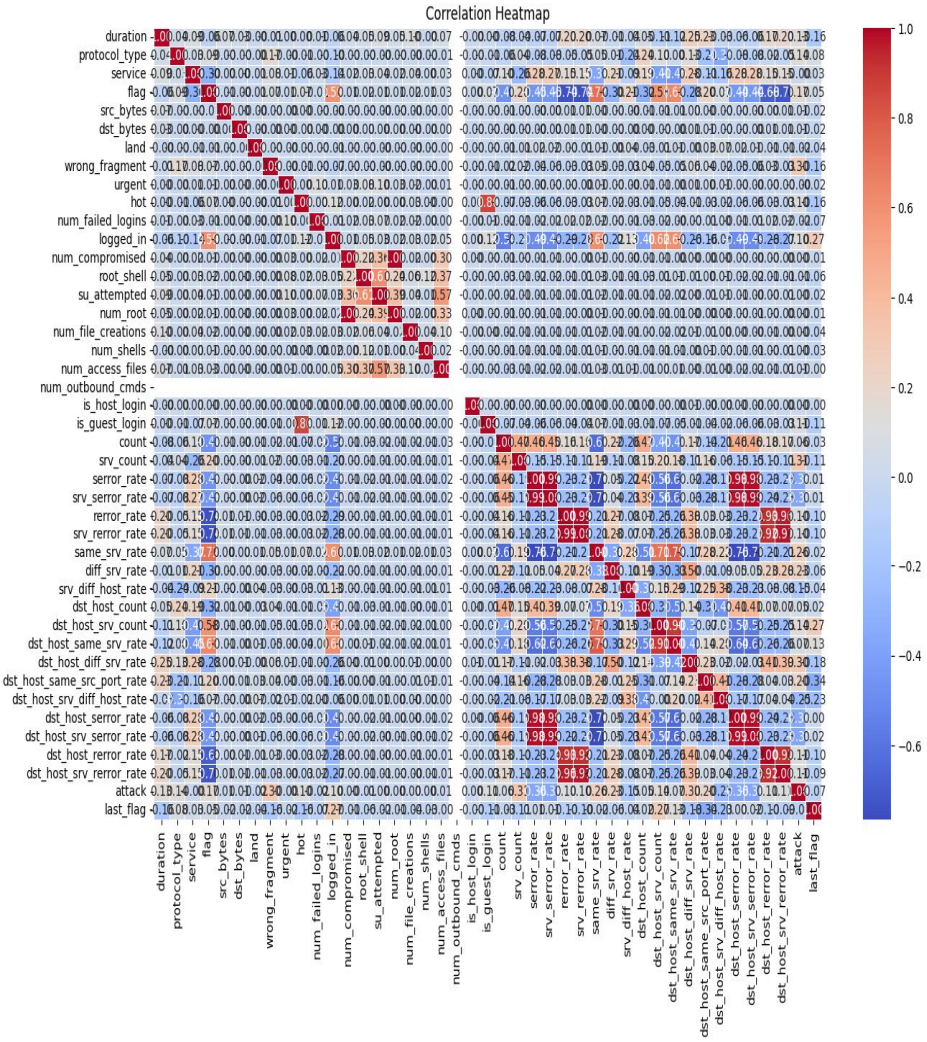

Fig. 3. Correlation Heat Map [Author]

4. RESULTS AND DISCUSSION

4.1 Results

The attack class mappings and the results obtained from the different kernels simulations are here presented. the provided results present an evaluation of support vector machine (svm) models with different kernel types for network anomaly detection.

4.1.1 Attack class mapping for NSL KDD

0. Back:	Refers to a network attack attempting to overwhelm a system or network resource, causing denial of service.
1. Buffer Overflow:	Involves exploiting a program's buffer overflow vulnerability to execute malicious code.
2. FTP Write:	Indicates an attack where unauthorized users attempt to write or transfer files via the FTP service.
3. Guess Password:	Involves attempting to gain unauthorized access by repeatedly guessing passwords.

4. IMAP: Stands for Internet Message Access Protocol, an attack involving unauthorized access or manipulation of email accounts using IMAP.
5. IPSweep: Refers to scanning a range of IP addresses to identify live hosts on a network.
6. Land: A type of network attack where a malicious packet is crafted to cause a system to respond to itself.
7. Loadmodule: Involves loading a malicious module or code into a system to exploit vulnerabilities.
8. Multihop: Indicates a network attack involving multiple intermediate hosts to conceal the true origin of the attack.
9. Neptune: Denotes a denial-of-service attack that floods the target with traffic, rendering it unavailable.
10. Nmap: Stands for Network Mapper, an attack using the Nmap tool to discover and map network hosts.
11. Normal: Represents normal network traffic, not associated with any malicious activity.
12. Perl: Involves exploiting vulnerabilities in Perl scripts or applications.
13. PHF: Stands for "Personal Home Page," an attack attempting to exploit the CGI program used for accessing personal webpages.
14. Pod: Refers to a network attack that attempts to overload the target system's resources.
15. Portsweep: Indicates scanning multiple hosts for open ports, often as a precursor to an attack.
16. Rootkit: Involves installing software to gain unauthorized access while concealing its presence.
17. Satan: Refers to a network attack using the tool Satan, which performs security vulnerability assessments.
18. Smurf: Denotes a type of denial-of-service attack that floods a network with spoofed ICMP echo requests.
19. Spy: Suggests espionage-related activity, possibly involving unauthorized access to sensitive information.
20. Teardrop: A denial-of-service attack involving fragmented packets designed to crash the target system.
21. Warezclient: Indicates a client involved in downloading or distributing pirated software.
22. Warezmaster: Denotes a central figure or server coordinating the distribution of pirated software.

4.1.2 Results of Linear Kernel SVM

Accuracy: 0.9946814844215122
F-score: 0.9947144861559098
Macro average: 0.79
Weighted average: 0.99

Table 1: Classification Report for Linear Kernel

ATTACK CLASS	PRECISION	RECALL	F1 SCORE
0	1.00	0.98	0.99
1	1.00	0.44	0.62
2	0.00	0.00	0.00
3	0.92	1.00	0.96
4	0.50	1.00	0.67
5	0.96	0.99	0.98

6	0.75	1.00	0.86
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	1.00	1.00	1.00
10	0.96	0.97	0.97
11	1.00	0.99	1.00
12	1.00	1.00	1.00
13	1.00	1.00	1.00
14	1.00	0.93	0.96
15	0.98	0.99	0.99
16	0.00	0.00	0.00
17	0.98	0.97	0.97
18	0.98	1.00	0.99
19	0.00	0.00	0.00
20	1.00	1.00	1.00
21	0.91	1.00	0.95
22	1.00	0.75	0.86

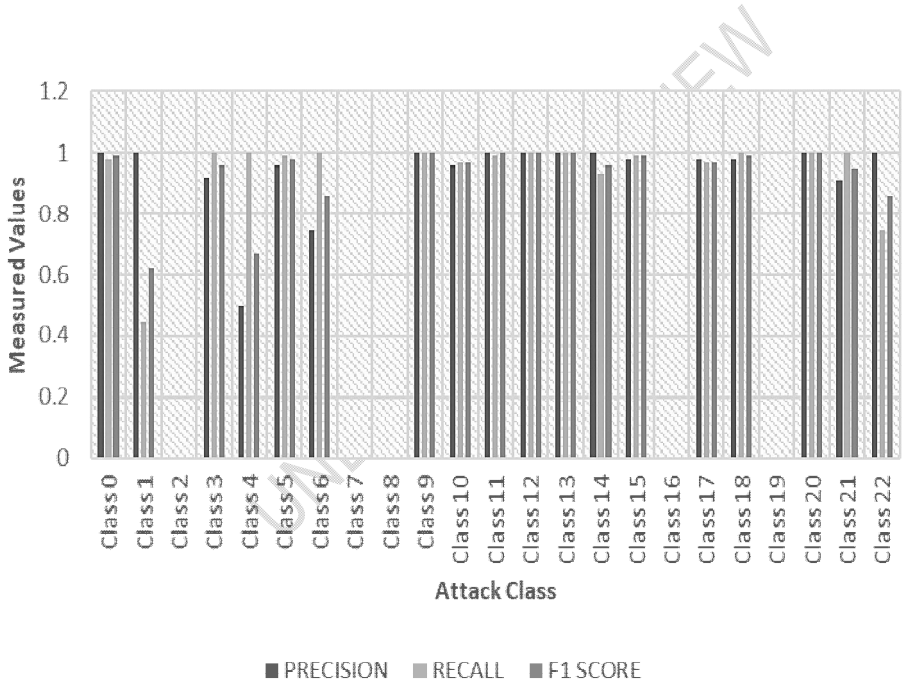

Fig.4. Linear Kernel Attack Classification Results

The Linear Kernel SVM results as shown in Table 1 and charted in Fig.4 achieved a high accuracy of 99.47% and an F-score of 99.47%. The macro average F-score was 0.79, indicating variability in performance across different classes, while the weighted average was 0.99, showing strong overall performance. The classification report highlighted excellent precision and recall for most classes, especially with several achieving a perfect score (e.g., classes 0, 11, 12, 13, 20). However, some classes like 1, 7, 8, 16, and 19 had significantly lower performance, with zero recall and F-scores.

4.1.3 Results of Poly Kernel SVM:

Accuracy: 0.9954752927168089

F-score: 0.9952928631615904
Macro avg: 0.72
Weighted avg: 1.00

Table 2: Classification Report for Poly Kernel SVM

ATTACK CLASS	PRECISION	RECALL	F1 SCORE
0	1.00	0.98	0.99
1	1.00	0.33	0.50
2	0.00	0.00	0.00
3	0.92	1.00	0.96
4	1.00	1.00	1.00
5	0.96	0.99	0.97
6	1.00	1.00	1.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	1.00	1.00	1.00
10	0.95	0.98	0.97
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	0.50	1.00	0.67
14	1.00	0.93	0.96
15	1.00	1.00	1.00
16	0.00	0.00	0.00
17	0.99	0.98	0.99
18	0.97	0.99	0.98
19	0.00	0.00	0.00
20	1.00	1.00	1.00
21	0.93	0.99	0.96
22	0.00	0.00	0.00

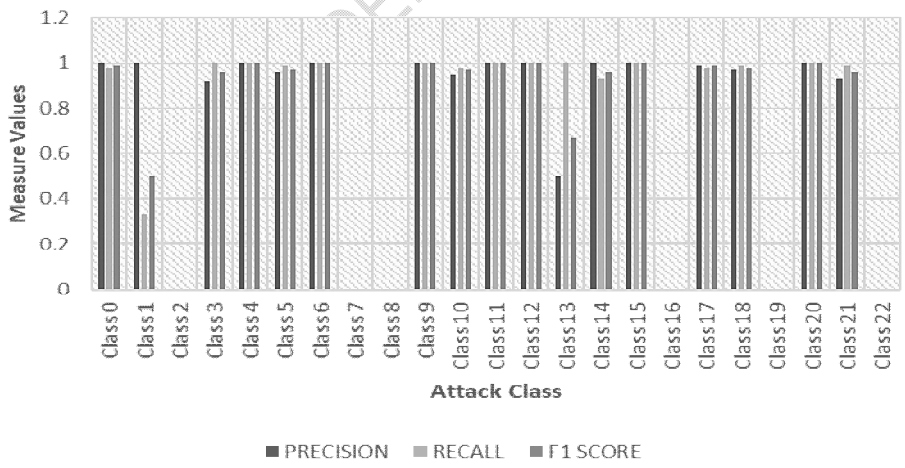

Fig. 5. Poly Kernel Attack Classification Results

The result of Poly Kernel SVM demonstrated an accuracy of 99.55% and an F-score of 99.53%as shown in Table 2 and represented in Fig.5. The macro average F-score was 0.72, suggesting notable variation in performance between classes, while the weighted average

was 1.00. The classification report showed high precision and recall for many classes, with several achieving perfect scores (e.g., classes 0, 4, 6, 9, 11, 12, 20). Nonetheless, certain classes such as 1, 7, 8, 16, 19, and 22 showed poor performance, with zero recall and F-scores.

4.1.4 Results ofRbf Kernel SVM:

Accuracy: 0.9955149831315737
F-score: 0.9952845972345609
macro avg: 0.48
weighted avg: 0.92

Table 3: Classification Report for Rbf Kernel SVM

ATTACK CLASS	PRECISION	RECALL	F1 SCORE
0	0.99	0.98	0.99
1	1.00	0.33	0.50
2	0.00	0.00	0.00
3	1.00	0.91	0.95
4	1.00	1.00	1.00
5	0.96	0.99	0.97
6	1.00	0.67	0.80
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	1.00	1.00	1.00
10	0.95	0.98	0.97
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	1.00	1.00	1.00
14	1.00	0.93	0.96
15	0.99	0,99	0.99
16	0.00	0.00	0.00
17	0.99	0,99	0.99
18	0.98	0,99	0.99
19	0.00	0.00	0.00
20	1.00	1.00	1.00
21	0.92	0.99	0.95
22	0.00	0.00	0.00

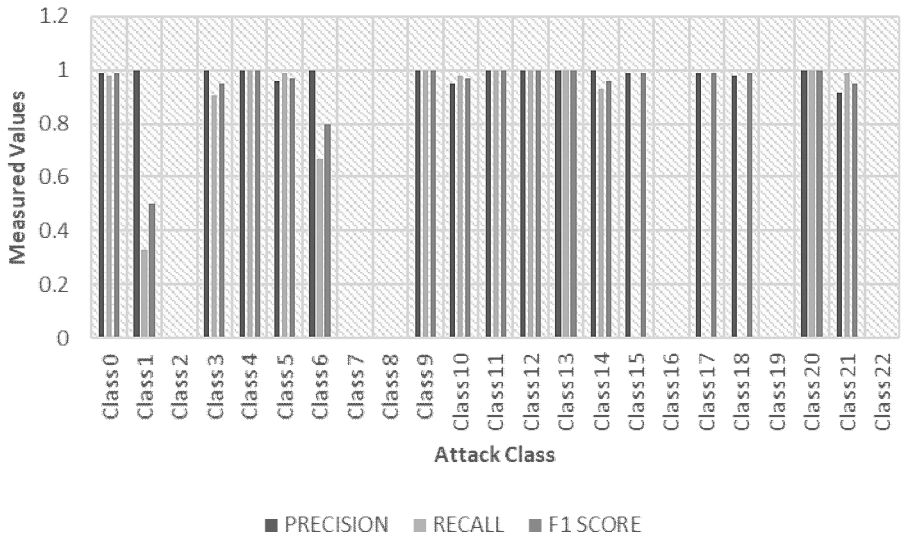

Fig.6. Kbf Kernel Attack Classification Results

Looking at Table 3 and Fig. 6, the RBF Kernel SVM recorded an accuracy of 99.55% and an F-score of 99.53%. The macro average F-score was 0.48, indicating considerable variation in performance across classes, while the weighted average was 0.92. The classification report indicated high precision and recall for several classes, particularly those achieving perfect scores (e.g., classes 0, 4, 9, 11, 12, 13, 20). However, classes such as 1, 7, 8, 16, 19, and 22 had very low performance, with zero recall and F-scores.

4.1.5 Results for Sigmoid Kernel SVM:

Accuracy: 0.9210954554475095
F-score: 0.9179688416593323
macro avg: 0.79
weighted avg: 0.99

Table 4: Classification Report for Sigmoid Kernel SVM

ATTACK CLASS	PRECISION	RECALL	F1 SCORE
0	0.56	0.86	0.68
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.35	0.82	0.49
4	1.00	1.00	1.00
5	0.79	0.84	0.82
6	0.67	0.67	0.67
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.96	0.99	0.98
10	0.60	0.81	0.69
11	0.96	0.95	0.95
12	0.00	0.00	0.00
13	0.00	0.00	0.00
14	0.59	0.31	0.40
15	0.56	0.31	0.40
16	0.00	0.00	0.00
17	0.39	0.36	0.37
18	0.98	0.93	0.96
19	0.00	0.00	0.00
20	1.00	0.89	0.94
21	0.70	0.39	0.50
22	0.00	0.00	0.00

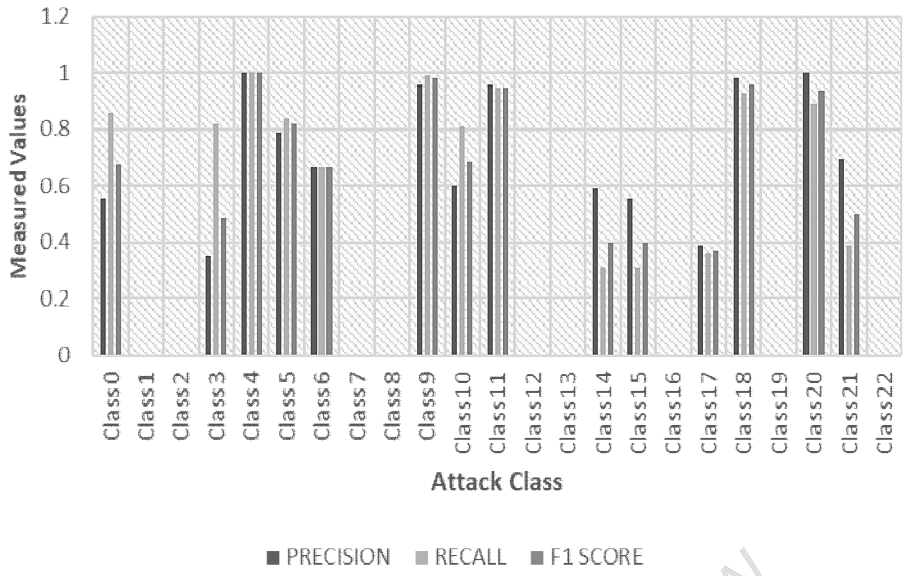

Fig. 7. Sigmoid Kernel Attack Classification Results

Table 4 and Fig. 7 show the Sigmoid Kernel SVM achieved a lower accuracy of 92.11% and an F-score of 91.80%. The macro average F-score was 0.79, and the weighted average was 0.99. The classification report revealed varying performance, with some classes like 4, 9, 11, and 20 achieving high precision and recall. However, several classes such as 1, 8, 12, 13, 16, 19, and 22 performed poorly, with zero recall and F-scores. This kernel showed considerable inconsistency and lower overall effectiveness compared to the other kernels.

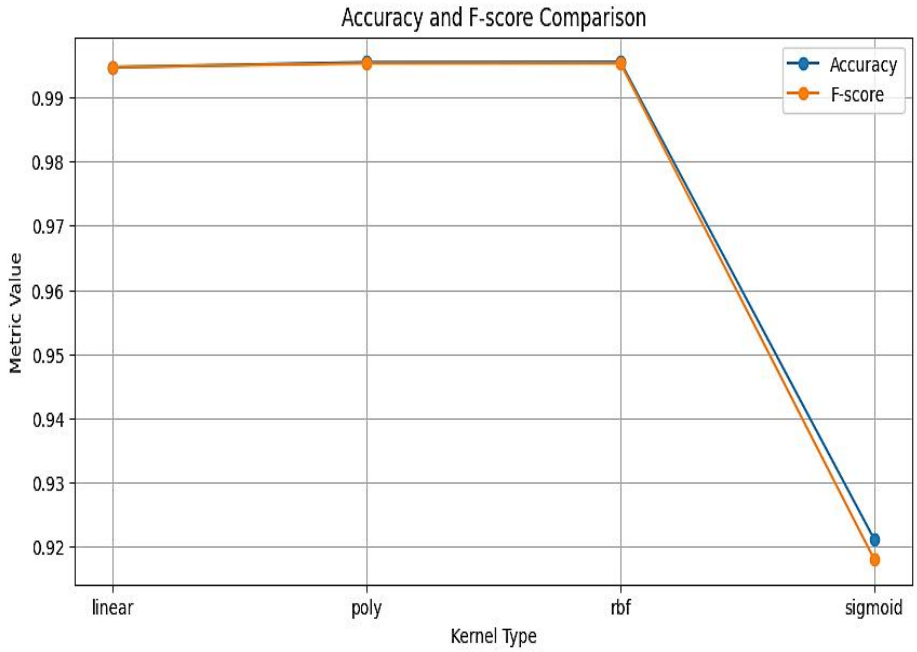

Fig. 8. Accuracy and F-scores comparison

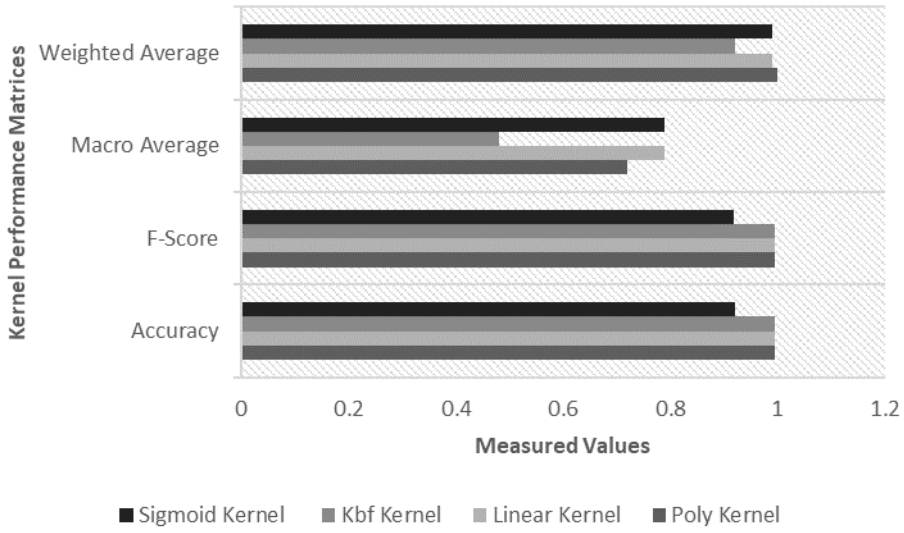

Fig. 9: Kernel Performance Analysis

From Fig. 8 and Fig. 9, the SVM models exhibit high accuracy and F1-scores across all kernel types, indicating their effectiveness in classifying network traffic and detecting anomalies.The accuracy values range from approximately 92% to 99.55%, demonstrating the models' ability to correctly classify most instances of the dataset. The linear and polynomial kernel SVMs demonstrate the highest accuracy and F1-scores among all kernels, indicating their robust performance in capturing the underlying patterns in the data. The radial basis function (RBF) kernel SVM performs slightly lower than the linear and polynomial kernels but still achieves high accuracy and precision.In contrast, the sigmoid kernel SVM exhibits noticeably lower accuracy and F1-scores compared to other kernels, suggesting its limited suitability for network anomaly detection in this scenario.

The results underscore the importance of selecting an appropriate SVM kernel for network anomaly detection tasks.Linear and polynomial kernels demonstrate superior performance, making them preferable choices for this application.The findings also highlight the need for further investigation into the factors influencing the performance of different SVM kernels, such as dataset characteristics and feature representation.SVMs with different kernels have different hyperparameters that need to be tuned for optimal performance. Inadequate tuning of hyperparameters, such as the regularization parameter for linear SVM and the kernel coefficient for RBF SVM, could lead to suboptimal results.

5. CONCLUSION

The study presents an evaluation of Support Vector Machine (SVM) models with different kernel types for network anomaly detection. The experiments and analysis aimed to discern the performance characteristics of each SVM kernel and their implications for network security applications.

The findings of the research offer valuable insights into the comparative effectiveness of SVM kernels in accurately classifying network traffic and detecting anomalies. Notably, the

linear and polynomial SVM kernels emerged as top performers, showcasing robust performance in capturing underlying patterns within the data. These kernels exhibited high accuracy and F1-scores, underscoring their suitability for anomaly detection tasks.Conversely, the sigmoid kernel SVM demonstrated comparatively lower accuracy and F1-scores, indicating its limited effectiveness in capturing the complexity of network traffic and identifying anomalies. Despite its computational efficiency, the sigmoid kernel may not be well-suited for nuanced anomaly detection scenarios.

Our research underscores the importance of informed model selection in network security applications. By understanding the strengths and limitations of different SVM kernels, practitioners can make judicious decisions in deploying intrusion detection systems. Moreover, our findings highlight the need for ongoing research to explore novel approaches and hybrid models that leverage the strengths of multiple kernels for enhanced anomaly detection capabilities. Machine learning can be beneficial in many ways, and it is necessary to study its functionality in many cases. For instance, machine learning can be of great value in anomaly detection in devices through vibration analysis [22], in rain attenuation analysis [23, 24], In energy systems monitoring and performance prediction [25, 26] and in monitoring and evaluation of spectrum occupancy [27, 28] and network performance monitoring [29].

In conclusion, the study contributes to advancing anomaly detection techniques and developing resilient network security solutions. By providing empirical evidence and actionable insights, the research contributes to the empowerment of cybersecurity professionals in their efforts to safeguard critical network infrastructures against evolving cyber threats. Through continued collaboration and research endeavours, we can further strengthen the resilience of modern network systems and mitigate the risks posed by malicious activities.

Building upon the findings of this research, several avenues for future exploration may emerge, offering opportunities to advance the field of network anomaly detection and enhance cybersecurity practices. Investigating the efficacy of hybrid models that combine multiple SVM kernels or integrate SVM with other machine-learning techniques could yield improved anomaly detection capabilities. Hybrid approaches have the potential to leverage the strengths of different algorithms and enhance overall performance. Experimenting with ensemble learning techniques, such as bagging or boosting, in conjunction with SVM models could lead to more robust and resilient anomaly detection systems. Ensemble methods harness the collective intelligence of multiple models to achieve superior performance compared to individual classifiers.

The study's findings provided valuable insights into the effectiveness of SVM kernels for network anomaly detection. By comparing their accuracy and F1-score, the most suitable kernel for detecting specific cyber threats was identified, thereby contributing to the ongoing discourse in network security research.

From the results obtained, it will be necessary to carry out an analysis of the decision tree algorithm and compare its performance with the SVM. It will also be of great value to consider the use of hybrid methods in the intrusion detection for more robust performance.

Disclaimer (Artificial intelligence)

Authors hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc) and text-to-image generators have been used during the writing or editing of manuscripts.

REFERENCES

1. Divekar A, Parekh M, SavlaV, Mishra R, ShiroleM. Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives, in 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), Kathmandu, Nepal; 2018.doi: 10.1109/CCCS.2018.8586840.

2. Ao H. Using Machine Learning Models to Detect Different Intrusion on NSL-KDD,in 2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE), SC, USA; 2021, doi: 10.1109/CSAIEE54046.2021.9543241.

3. Sangve SM,Thool R. A formal assessment of anomaly network intrusion detection methods and techniques using various datasets,in 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Davangere, India; 2015, doi: 10.1109/ICATCCT.2015.7456894.

4. Sekhar C, Pavani K, Rao MS.Comparative analysis on Intrusion Detection system through ML and DL Techniques: Survey,in 2021 International Conference on Computational Intelligence and Computing Applications (ICCICA), Nagpur, India; 2021,doi: 10.1109/ICCICA52458.2021.9697291.

5. Singh S, Banerjee S. Machine Learning Mechanisms for Network Anomaly Detection System: A Review,in 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India; 2020,doi: 10.1109/ICCSP48568.2020.9182197.

6. Sharma A, Babbar H. NSL-KDD: Cyberattack Detection in IoT Utilizing Machine Learning Approaches,in 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gautam Buddha Nagar, India; 2023,doi: 10.1109/UPCON59197.2023.10434690.

7. Sibai FN, Asaduzzaman A,Sibai A. A Comparative Study of Machine Learning Methods for Intrusion Detection,in 2023 10th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkiye; 2023, doi: 10.1109/ICEEE59925.2023.00041.

8. Yilmaz AA. Intrusion Detection in Computer Networks using Optimized Machine Learning Algorithms, in 2022 3rd International Informatics and Software Engineering Conference (IISEC), Ankara, Turkey; 2022, doi: 10.1109/IISEC56263.2022.9998258.

9. Mhamdi L, Hamdi H, Mahmood MA. Network Intrusion Detection in Software-Defined Network Using Deep and Machine Learning, in 2023 IEEE Global Communications Conference, Kuala Lumpur, Malaysia; 2023, doi: 10.1109/GLOBECOM54140.2023.10437050.

10. Pervez MS, and D. M. Farid DM. Feature Selection and Intrusion Classification in NSL-KDD cup 99 Dataset Employing SVMs, in the 8th International Conference on Software, Knowledge, Information Management and Applications (SKIMA 2014), Dhaka, Bangladesh; 2014, doi: 10.1109/SKIMA.2014.7083539.

11. Nath MD, and Bhattasali T. Anomaly Detection Using Machine Learning Approaches. Azerbaijan Journal of High Performance Computing, 2020; 3(2):196-206. https://doi.org/10.32010/26166127.2020.3.2.196.206.

12. SinclairC., Pierce L., and Matzner, S. An application of machinelearning to network intrusion detection. In Proceedings 15th Annual Computer SecurityApplications Conference (ACSAC'99)(1999, December):371-377. IEEE.

13. Rostovski J, Krivošei, A, Kuusik A, Alam MM and AhmadovU.Real-Time Gait Anomaly Detection Using SVM Time Series Classification, in the 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco; 2023:1389-1394, doi: 10.1109/IWCMC58020.2023.10182666.

14. Hosseinzadeh, M., Rahmani, A.M., Vo, B. et al. Improving Security Using SVM-based Anomaly Detection: Issues and Challenges. Soft Comput 25, 2021:3195–3223. https://doi.org/10.1007/s00500-020-05373-x.

15. Zhang Y, Yang Q, Lambotharan S, Kyriakopoulos K, GhafirI, and AsSadhan B. Anomaly-Based Network Intrusion Detection Using SVM, in the 11th International Conference on Wireless Communications and Signal Processing (WCSP), Xi'an, China, 2019:1-6, doi: 10.1109/WCSP.2019.8927907.

16. Yu L, Xu L, Jiang X. An Effective Method for Detecting Unknown Types of Attacks Based on Log-Cosh Variational Autoencoder. Applied Sciences. 2023; 13(22):12492. https://doi.org/10.3390/app132212492

17. Sridhar P, Arivan SD, Akshay R, Farhathullah R. Anomaly Detection using CNN with SVM, in 2022 8th International Conference on Smart Structures and Systems (ICSSS), Chennai, India; 2022, doi: 10.1109/ICSSS54381.2022.9782229.

18 Goel A, Srivastava SK. Role of Kernel Parameters in Performance Evaluation of SVM, in 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, India; 2016, doi: 10.1109/CICT.2016.40.

19. Zare T, Sadeghi MT,Abutalebi HR. A comparative study of Multiple Kernel Learning approaches for SVM classification, in the 7th International Symposium on Telecommunications (IST'2014), Tehran, Iran; 2014, doi: 10.1109/ISTEL.2014.7000674.

20. Zhao Y, Chen J, Wu D, Teng J, Sharma N, Sajjanhar A, Blumenstein M. Network Anomaly Detection by Using a Time-Decay Closed Frequent Pattern. Information. 2019; 10(8):262. https://doi.org/10.3390/info10080262.

21. Andika H, Junho H, Chen-Ching L, and Kusuma GP. Evaluation of Machine Learning Techniques for Anomaly Detection on Hourly Basis kpi. Journal of Theoretical and Applied Information Technology 2023;101: 2023.

22. Jameson, F., Ubom, E., andUkommi, U. Vibration Analysis for Predictive Maintenance and Improved Machine Reliability of Electric Motors in Centrifugal Pumps. In: Ekpo, S.C. (eds) The Second International Adaptive and Sustainable Science, Engineering and Technology Conference. ASSET 2023. Signals and Communication Technology. Springer, Cham. 2024. https://doi.org/10.1007/978-3-031-53935-0_16

23. K. Ekanem, E. Ubom and U. Ukommi, Analysis of Rain Attenuation for Satellite Communication in Akwa Ibom State, Nigeria in the 18th International Conference and Exhibition on Power and Telecommunication (ICEPT), Abeokuta, Ogun State, Nigeria, 10th – 14th October 2022. pp. 25 – 34.

24. Ukommi, U, Ekanem, K, Ubom, E and Udofia, K, Evaluation of Rainfall Rates and Rain-Induced Signal Attenuation for Satellite Communication in the South-South region of Nigeria. Nigerian Journal of Technology (NIJOTECH), 2024; 42(4):472-477.

25. C. A. Adadu, D. D. Ekpo and D. A. Kogh, Design and Development of Standard Modern 12 Volts Mobile Electric Battery Charging Machine. Journal of Research and Innovations in Engineering, 2020; 5(1):75-80.

26. D. D. Ekpo, "Electricity Generation Potential from Municipal Solid Waste in Uyo Metropolis, Nigeria," Doctoral Dissertation, Ibadan, Nigeria, 2019.

27. Ubom, E., Akpanobong, A. and Ukommi, U., Spectrum Occupancy in Rural Nigeria: A Case for A Lightly Licensed Spectrum Band for Rural Broadband Enhancement. International Journal of Computer Science and Information Technology (IJCSIT), 2019; 11(4): 81-99.

28. Ubom, E. and Ukommi U, Comparative evaluation of spectrum occupancy of the broadcasting bands in urban, sub-urban and rural environments. Nigerian Journal of Technology. 2023; 41(6): 1008–1016.

29. Vincent B. Umoh, Emmanuel A. Ubom, Joan B. Umoh, Unwana M. Ekpe, “A Framework for a User-Centric Determination of Mobile Broadband Performance in Nigeria”, 1st International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS), Abuja, Nigeria, 15-17 July, 2021. DOI: 10.1109/ICMEAS52683.2021.9692425.