Research Article

Intrusion Detection System Based on Machine Learning Algorithms:(SVM and Genetic Algorithm)

Abdulazeez Khlaif Shathir Alsajri[1],*, ID Amani Steiti[2], ID

[1] Computer Science Department, University Arts, Sciences and Technology, Beirut, Lebanon.

[2] University Tishreen, faculty of information engineering, Department of Computer Systems And Networks, Latakia, Syria.

ABSTRACT

The widespread utilization of the internet and computer systems has resulted in notable security concerns, characterized by a surge in intrusions and vulnerabilities. Malicious users manipulate internal systems, resulting in the exploitation of software flaws and default setups. With the integration of the internet into society, there is an emergence of new risks such as viruses and worms, which highlights the importance of implementing robust security measures. Intrusion detection systems (IDS) are security technologies utilized to monitor and analyze network traffic or system activity with the purpose of identifying hostile behavior. This article presents a proposed method for detecting intrusion in network traffic using a hybrid approach, which combines a genetic algorithm and an SVM algorithm. The model underwent training and testing on the KDDCup99 dataset, with a reduction in features from 42 to 29 using the hybrid approach. The results demonstrated that throughout the system testing, it exhibited a remarkable accuracy of 0.999. Additionally, it achieved a true positive value of 0.9987 and a false negative rate of 0.012.

1. INTRODUCTION

In recent decades, internet and computer systems have faced significant security challenges as a result of the widespread utilization of networks. The CERT numbers indicate a significant surge in intrusions, which have the potential to result in catastrophic events. and breach computer security protocols such as Confidentiality, Integrity, and Availability (CIA) [1]. The rapid advancement of internet technology has resulted in the creation of novel computer network applications across multiple industries, rendering them susceptible to exploitation and vulnerabilities. Malicious individuals and hackers manipulate internal systems, resulting in software glitches, administrative problems, and default settings. Therefore, it is imperative to implement security measures in order to safeguard users' systems from unauthorized access. The study of IDSs confronts obstacles in effectively addressing arbitrary intrusion categories and computational power limitations, which are caused by the increasing network throughput and security threats. These challenges are key research issues that need to be addressed.[2].

Constructing and upholding such systems poses significant technical and economic challenges. Intrusion detection systems are designed to identify and respond to attacks on computer systems and networks, with the goal of ensuring the ongoing security of information systems. These systems oversee usage to identify insecure conditions and efforts by authorized users or other entities to exploit security weaknesses. The definition of a completely secure information system may be restricted by legacy or operational limitations [3].

SVMs are a type of supervised learning models that are accompanied by learning algorithms. These models are utilized to evaluate data for the purposes of classification and regression analysis. The Support Vector Machine (SVM) technique, is a prediction approach that is resilient and based on the statistical learning framework or VC theory. SVMs construct models that classify fresh samples into one of two categories, so functioning as non-probabilistic binary linear classifiers. SVMs

*Corresponding author. Email: aka104@live.aul.edu.lb

have the capability to effectively carry out non-linear classification by utilizing the kernel trick, a technique that transforms input data into high-dimensional feature spaces [4].

The Support Vector Machine (SVM) is a highly popular machine learning technique that categorizes data by utilizing support vectors derived from training data samples.   The main goal is to identify the optimal hyperplane for fresh data points.   SVM classifiers have the ability to utilize different kernel functions, such as linear, polynomial, Gaussian radial basis function, and sigmoid, to handle non-linear data samples[5].

Genetic algorithms are extensively employed in the field of computers for the purpose of resolving intricate issues, offering solutions that are robust, adaptable, and optimum. Genetic algorithms, drawing inspiration from biological phenomena like as natural selection, evolution, mutation theory, and genetic inheritance, employ selection and crossover modules to identify the optimal solution for certain problem domains. These algorithms facilitate the interchange of parameters in order to generate novel solutions. The mutation module is utilized to improve parameters inside genetic algorithms, which are a commonly employed approach in the field of network security. Specifically, genetic algorithms are vital in the design and proposal of IDSs. These algorithms aid in the classification of security threats and the generation of tailored rules for diverse types of attacks [6].

2. RELATED WORKED

The proliferation of digitalization and (IoT) has resulted in a significant surge in security events, encompassing unauthorized access, malware assaults, zero-day attacks, data breaches, (DoS), social engineering, and phishing. In the year 2010, the number of distinct malware executables was recorded to be fewer than 50 million. Cybercrime and assaults have the potential to result in substantial financial ramifications, It is imperative for organizations to implement a robust cybersecurity strategy in order to effectively reduce these financial losses. The preservation of a nation's security is contingent upon the availability of secure apps and solutions for enterprises, governments, and individuals, together with their capacity to promptly identify and eradicate cyber threats. The speedy resolution of the vital matter of accurately identifying and safeguarding essential systems from cyber events is of utmost importance. *Cybersecurity*: encompasses a range of technologies and procedures that are specifically devised to safeguard computers, networks, software applications, and sensitive information against potential threats, such as illegal access, damage, or malicious attacks. In recent years, the field of cybersecurity has experienced notable transformations, primarily influenced by the advancements in (DS). (ML) is an integral component of (AI) and serves a crucial function in the extraction of valuable insights from data. Machine learning (ML) has the potential to bring about substantial transformations in the field of cybersecurity, therefore ushering in a novel scientific paradigm [7].

2.1 Types of cyber security

1. **Critical infrastructure cyber security:** including physical and digital systems providing essential services for a country's economy, poses significant economic or public health and safety risks, while cyber security protects these systems from potential cyber-related attacks [8]. Critical infrastructure comprises16 interconnected sectors that provide essential functions for our way of life. Any threat to these sectors could potentially harm national security, economic growth, public health, and safety [9].
2. **Network security:** The field of network security is of paramount importance within the realm of cybersecurity, since its primary objective is to safeguard computer networks against various forms of cyber attacks. The primary goals encompassed by this initiative involve the prevention of illegal access to network resources, the identification and cessation of cyberattacks, and the provision of safe access for those with permitted permissions [10].
3. **Cloud Security:** It is of utmost importance for enterprises and governments, hence emphasizing the significance of cloud security. The process encompasses methodologies and technological measures aimed at safeguarding cloud computing environments, therefore thwarting unlawful entry and guaranteeing the provision of internet-based information technology services [11].
4. **IoT (Internet of Things) security:** The (IoT) is a network of web-enabled objects that can connect and exchange information, including personal fitness trackers, TVs, thermostats, and connected cars. With over 30 billion IoT connections projected by 2025, it's crucial to understand how to securely use IoT devices in an organization, as the number of connected devices is increasing [12].
5. **Application security:** involves using security solutions, tools, and processes to protect applications throughout their life cycle. With modern development speed, organizations must ensure security before an application is live.

2.2 Challenges in Cybersecurity

1. Evolving threat landscape: Cyber threats evolve with new techniques, requiring continuous monitoring, intelligence, and proactive defense strategies to stay ahead.
2. Sophistication of cyberattacks: Cybercriminals employ advanced techniques, including zero-day exploits, social engineering, and polymorphic malware, to bypass traditional security measures.
3. Insider threats: Insider threats pose cybersecurity challenges; organizations must implement access control mechanisms, employee training, and continuous monitoring to mitigate risks.
4. Compliance and regulatory requirements: Organizations face complex compliance challenges in data protection, privacy, and cybersecurity, requiring demanding and resource-intensive compliance efforts [13].

2.3 Approaches to Cybersecurity

1. Defense in depth: A robust defense system combines network, endpoint, access controls, encryption, and user awareness training to create multiple barriers deterring attackers.
2. Threat intelligence: Threat intelligence gathers information on potential threats, including vulnerabilities, malware signatures, and IOCs, enabling organizations to proactively identify and mitigate emerging threats.
3. IDS and IPS monitor network traffic, detect suspicious activities, and block threats using techniques like signature-based, anomaly detection, and machine learning algorithms.
4. Security awareness and training: User awareness and training programs are crucial in cybersecurity, educating users on common threats, phishing techniques, password hygiene, and safe browsing practices to reduce attack likelihood.
5. Regular updates and patch management: Regular patch management is essential for maintaining systems, software, and applications, addressing vulnerabilities and reducing attacker exploitation risk [14].

3. INTRUSION DETECTION SYSTEM

IDS are crucial in modern cybersecurity infrastructure, identifying and responding to security breaches and unauthorized activities. They can be classified into network-based (NIDS) and host-based (HIDS) systems [15].
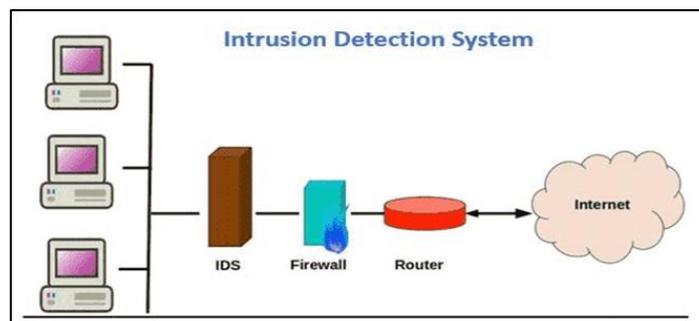

Fig1: Intrusion.Detection.System

1. **Network-Based.Intrusion.Detection.System.(NIDS):** NIDS are designed to examine network packets and employ various detection algorithms in order to identify and detect instances of unauthorized intrusions or malicious actions. These systems have the capability to be strategically implemented at several locations within the network architecture, including the perimeter or designated network segments [16].
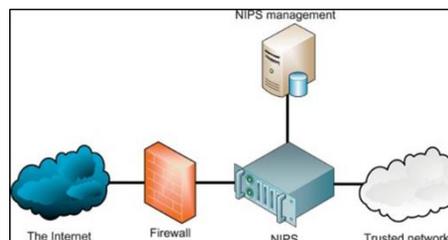

Fig.2. Network-Based.Intrusion.Detection.Systems.(NIDS)

2. **Host-Based.Intrusion.Detection.System.(HIDS):** HIDS is designed to monitor and analyze changes in host activity and events on specific hosts or devices. The software examines application logs, system calls, file-system updates, and packets. In the event that any potentially suspicious behavior is detected, a notification is triggered in order to safeguard the system from any malevolent attacks. The majority of industries exhibit a preference for (HIDS) as opposed to (NIDS), mostly due to the former's heavy reliance on system log analysis [17].
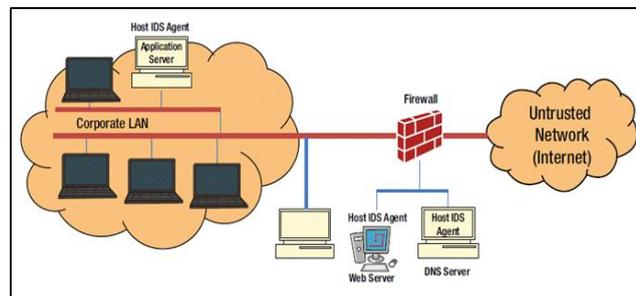

Fig.3. Host-Based.Intrusion.Detection.System.(HIDS)

3.1 Intrusion Detection Techniques

Intrusion Detection Systems (IDS) utilize a range of approaches to identify and detect unauthorized access attempts and abnormal behaviors. The strategies encompass signature-based detection, anomaly-based detection, and hybrid methodologies that integrate both approaches. Signature-based detection is a method that depends on pre-established patterns or signatures of recognized attacks. Conversely, anomaly-based detection operates by modeling the typical behavior of a system and generating warnings when departures from this norm are detected. Hybrid methodologies exploit the respective advantages of many techniques in order to augment the precision of detection [18].

3.2 The Benefits of Using an IDS for Network Security

IDS offers numerous benefits for network security, including enhanced network visibility, improved security, and enhanced organizational efficiency the top four pluses to utilising an IDS:
1. Detects malicious activity : IDS detects suspicious activity and alerts system administrator before significant damage occurs.
2. Improves network performance : IDS detects network performance issues, aiding in improving network performance.
3. Compliance needs : IDS assists organizations in compliance by generating reports and monitoring network activity.
4. Provides insights : IDS provides valuable network traffic insights for identifying vulnerabilities and enhancing security [19].

4   MACHINE LEARNING IN IDS

IDS is a monitoring tool, which can be implemented as either hardware or software, that performs analysis on data in order to identify and detect potential assaults on a system or network. Traditional methodologies are characterized by their intricate nature and the significant amount of time they need, resulting in reduced efficiency when confronted with large volumes of data, sometimes referred to as Big Data. The utilization of Big Data tools and approaches has the potential to decrease computation and training time. According to existing research, the utilization of machine learning algorithms in the context of (IDS) has demonstrated the potential to effectively mitigate false positive rates while concurrently enhancing the accuracy of the IDS [20].
ML, a subset of AI, improves systems' automatic abilities without explicit programming, enhancing detection accuracy in Intrusion Detection Systems (IDS) by analyzing large amounts of data. Typically, ML algorithms can be classified into three categories: [21].
1. **Supervised ML Algorithm:** Supervised algorithms are utilized to study datasets that are completely labeled with class information. These algorithms aim to identify and establish connections between the data and its corresponding

class through the application of classification or regression techniques. Some commonly used classification techniques in machine learning include Support Vector Machines (SVM), Discriminant Analysis, Naïve Bayes, Neural Networks, and Logistic Regression. Regression techniques often employed in data analysis and machine learning including Linear Regression, Support Vector Regression (SVR), Ensemble Methods, Decision Tree, and Random Forest. [21].

2. **Unsupervised ML Algorithm:** Unsupervised learning algorithms detect hidden structures in unlabeled data without training data using clustering, association analysis, or dimensionality reduction. Examples include K-Means, K-Mediods, C-Means, SvD, PCA, and genetic algorithms [21].

3. **Semi-Supervised ML algorithm:** Semi-supervised ML algorithm, a hybrid of unsupervised and supervised learning, uses unlabeled data for training and small amounts of labelled data for large datasets. It improves classifier diversity for accurate network intrusion detection [21].

5. SUPPORT VECTORS MACHINE (SVM) ALGORITHM

SVM is a widely used supervised machine learning algorithm for classification and regression, excelling in complex datasets and handling limited data, making it a powerful choice for various tasks [22].SVM algorithm is designed to identify an ideal hyperplane inside a feature space of large dimensionality, with the objective of effectively separating distinct data points. The hyperplane possesses the maximum margin, which is established by measuring the distance between the hyperplane and the nearest data points. The support vectors, which are the data points that are closest to the hyperplane, have a significant impact on establishing the decision boundary and exerting influence on the performance of the model [22].
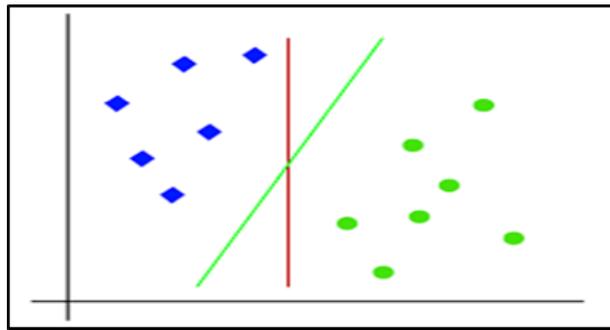

Fig.4. Support Vectors Machine (SVM) Algorithm

5.1 Types of Support Vector Machines

SVMs can handle linearly and non-linearly separable data, using a linear hyperplane for linear separability and a kernel function for non-linear separability.

1. **Linear SVM:** Linear SVM finds the hyperplane with the largest margin for linearly separable data using a straight line decision boundary.

$$w^T x + b = 0$$

where w is the weight vector, x is the input vector, and b is the bias term.

2. **Non-linear SVM:** Non-linear SVM is used for non-linearly separable data, mapping input data into higher-dimensional feature spaces using kernel functions and hyperplanes.

$$f(x) = w^T \phi(x) + b = 0$$

where $\phi(x)$ is the feature vector in the high-dimensional space, w is the weight vector, and b is the bias term [23].

5.2 How SVM Works

Support Vector Machines (SVM) are capable of effectively categorizing data points within a feature space of large dimensionality, even in cases when linear separability is not present. A separator is identified, and the data is converted into a hyperplane. Novel data attributes are employed for the purpose of forecasting the category to which a certain data entry pertains.

For instance, the above illustration depicts a scenario whereby the data points are segregated into two distinct groups.
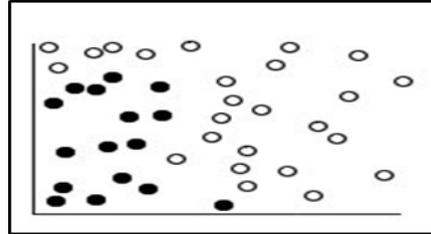

Fig.5.   Original.dataset

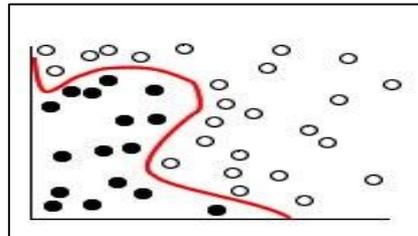The two groups can be delineated by use of a curve, as seen in the subsequent image.

Fig.6. Data with separator added

Upon undergoing the aforementioned transformation, the demarcation line between the two categories can be precisely delineated by a hyperplane, as visually depicted in the subsequent diagram. [24]
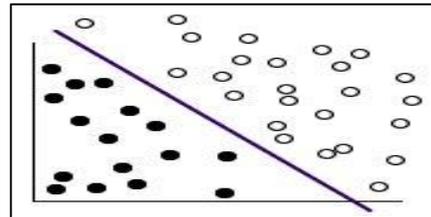

Fig.7. Transformed.data

6.   GENETIC ALGORITHM (GA)

The (GA) is an optimization algorithm that emulates the process of evolution in nature and employs techniques of natural selection to identify optimal solutions. The utilization of this method is prevalent in optimization and search issues with the purpose of identifying the optimal global solution. The primary processes employed by genetic algorithms (GAs) encompass selection, crossover, and mutation. The starting population is stochastically produced, consisting of a group of individuals that are assessed using the fitness function. The selection process for the subsequent generation involves the identification and inclusion of individuals with the highest fitness values. [25].

The objective function, also known as the fitness function, plays a critical role in genetic algorithms (GAs) since it assesses the quality of individuals and determines the selection of the most favorable ones to advance to the subsequent generation. The crossover operator involves the generation of new offspring by combining characteristics inherited from both parents. The mutation operator involves the use of random swaps in order to generate new solutions. The algorithm terminates upon the fulfillment of a predetermined condition [25].
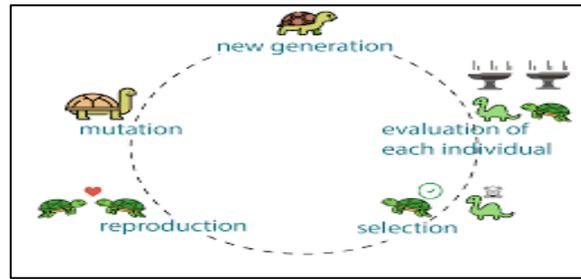
Fig.8. Genetic Algorithm

6.1 How the Genetic Algorithm Works

The genetic algorithm commences by generating an initial population by a random process, followed by the creation of subsequent populations in a sequential manner. The process involves utilizing the members of the present generation to procreate and give rise to the subsequent generation. The algorithm evaluates the fitness value of each member, normalizes them to generate expectation values, chooses parents based on their expectation values, selects elite individuals with lower fitness, generates offspring from parents through mutation or crossover, and replaces the current population with the offspring to form the subsequent generation. The algorithm terminates when any of the specified halting criteria are satisfied. [26].

7. METHODOLOGY

The methodology shown in the following figure19 shows the necessary steps to implement training the machine learning model to detect network intrusion using the algorithm (genetic algorithm and SVM algorithm), and also explains the characteristics of the kddcup99 dataset, implementing data pre-processing, and evaluating the model.
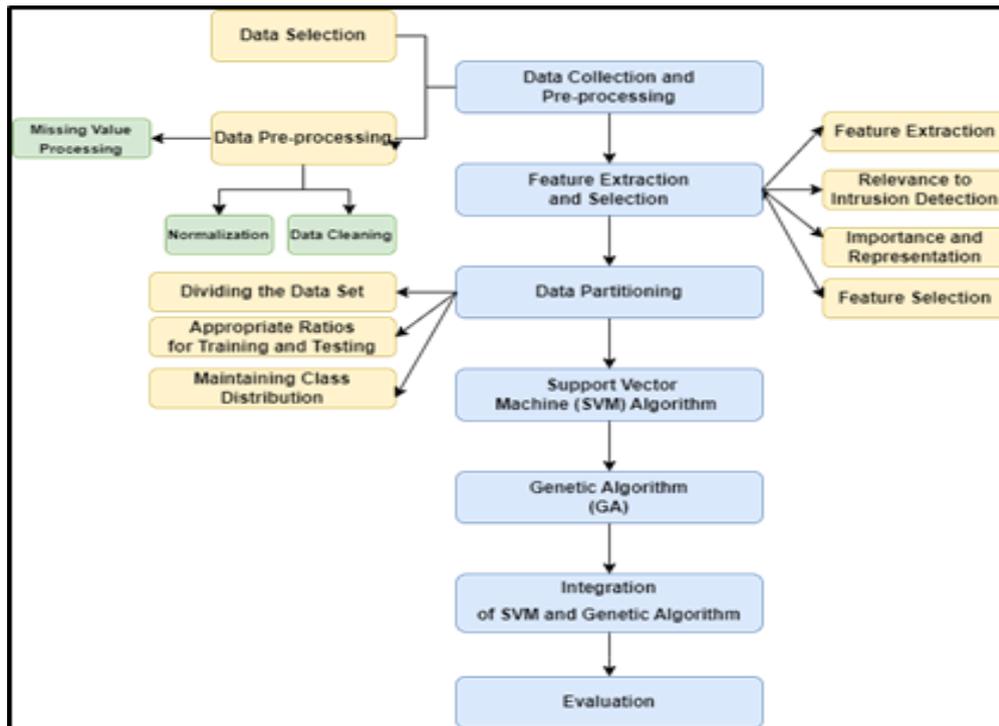

Fig.9. System Architecture

7.1 Data Collection and Pre-processing

The "Data collection and pre-processing" stage selection of suitable datasets that contain network traffic data for the purposes of training and testing an Intrusion Detection System (IDS). In this case, the "kddcup99" dataset is chosen as the source of network traffic data.

1. **Data Selection:** The "kddcup99" dataset is a well-known dataset widely used for evaluating intrusion detection systems. It includes a large collection of network connection records, categorized as either normal or intrusive connections. The dataset is appropriate for our research as it simulates real-world network traffic scenarios.

2. **Data Pre-processing:** Once the dataset is selected, data pre-processing steps are applied to ensure its quality and usability for building the IDS. This includes:

    *   **Data Cleaning:** In order to rectify any inconsistencies or flaws present in the dataset, it is important to undertake certain measures such as eliminating duplicate entries and effectively managing outliers. The purpose of this stage is to improve the precision and dependability of the analysis.
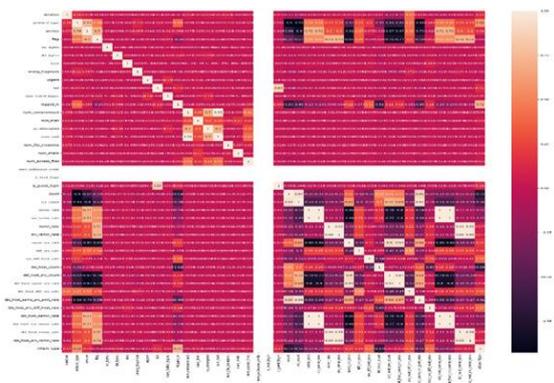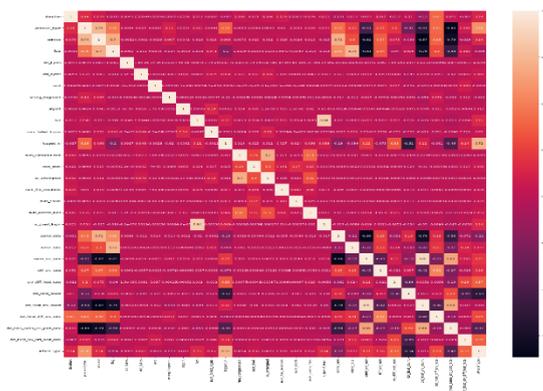

Fig.10. Show the correlation between data   Fig.11. remove the columns have correlation more than 90 %

    *   **Normalization:** Standardizing the characteristics in the dataset to a uniform scale. This guarantees that features with varying units of measurement are given equal treatment while training the model and prevents specific characteristics from overpowering others.

    *   **Missing Value Processing:** Identifying and addressing missing values in the dataset. Depending on the characteristics of the data, missing values can be imputed (replaced with estimated values) or eliminated, in order to ensure that the dataset is complete and appropriate for analysis.

Finding missing values of all features:

TABLE I. MISSING VALUES OF ALL FEATURES

index	features	Missing_counts	Missing_percent
0	duration	0	0.0
1	protocol_type	0	0.0
2	service	0	0.0
3	flag	0	0.0
4	src_bytes	0	0.0
5	dst_bytes	0	0.0
6	land	0	0.0
7	wrong_fragment	0	0.0
8	urgent	0	0.0
9	hot	0	0.0
10	num_failed_logins	0	0.0
11	logged_in	0	0.0
12	num_compromised	0	0.0
13	root_shell	0	0.0
14	su_attempted	0	0.0

15	num_root	0	0.0
16	num_file_creations	0	0.0
17	num_shells	0	0.0
18	num_access_files	0	0.0
19	num_outbound_cmds	0	0.0
20	is_host_login	0	0.0
21	is_guest_login	0	0.0
22	count	0	0.0
23	srv_count	0	0.0
24	serror_rate	0	0.0
25	Srv_serror_rate	0	0.0
26	rerror_rate	0	0.0
27	srv_rerror_rate	0	0.0
28	same_srv_rate	0	0.0
29	diff_srv_rate	0	0.0
30	srv_diff_host_rate	0	0.0
31	dst_host_count	0	0.0
32	dst_host_srv_count	0	0.0
33	dst_host_same_srv_rate	0	0.0
34	dst_host_diff_srv_rate	0	0.0
35	dst_host_same_src_port_rate	0	0.0
36	dst_host_srv_diff_host_rate	0	0.0
37	dst_host_serror_rate	0	0.0
38	dst_host_srv_serror_rate	0	0.0
39	dst_host_rerror_rate	0	0.0
40	dst_host_srv_rerror_rate	0	0.0
41	target	0	0.0

No missing value found, so we can further proceed to our next step.

To summarize, the "kddcup99" dataset requires the "Data collection and pre-processing" phase, which includes choosing the suitable network traffic dataset and carrying out necessary pre-processing tasks such as cleaning, normalization, and handling missing values. These stages guarantee the accuracy, consistency, and readiness of the dataset for training and testing the intrusion detection system using machine learning techniques such as SVM and Genetic Algorithm.

7.2 Feature Extraction and Selection

The method of "Feature extraction and selection" entails obtaining significant attributes from the network traffic data of the "kddcup99" dataset. These attributes, referred to as features, serve a vital role in facilitating the identification of network intrusions within the dataset.

1. **Feature Extraction:** Feature extraction involves the identification and isolation of particular properties from network traffic data, which can offer valuable insights into the behavior of network connections. The dataset "kddcup99" is being referred to.The properties encompassed in this context consist of information such as source and destination IP addresses, protocol kinds, service types, and various statistical measurements associated with data transfer.
2. **Relevance to Intrusion Detection:** The chosen characteristics are not arbitrary; instead, they are picked based on their possible relevance to the job of identifying network intrusions. Features that exhibit substantial differences between normal and intrusive connections are particularly important in identifying suspicious or malicious activity within network data.
3. **Importance and Representation:** Features are categorized according to their significance in detecting patterns of network activity linked to intrusions. Examples of separating typical network activity from potentially hazardous ones include the number of failed login attempts, the volume of data sent, and the connection duration.
4. **Feature Selection:** Due to the abundance of available features, it is crucial to choose a subset that offers the most pertinent information while avoiding needless complexity or noise.
   The selection of these features was based on the following criteria:
   - All of these attributes are numerical, which simplifies their utilization in machine learning models.
   - They exhibit a substantial correlation with the target variable, indicating their potential for predicting the target variable.
   - The variables have a low correlation, indicating that they do not offer duplicate information.

This stage involves evaluating the impact of each feature on the overall effectiveness of the intrusion detection system.

7.3 Data Partitioning

The technique of data splitting is essential in the building of an efficient intrusion detection system (IDS) using the kddcup99 dataset. The process entails dividing the dataset into separate subsets in order to enable the training and testing of the Intrusion Detection System (IDS).

1. **Dividing the Data Set:** The dataset "kddcup99" comprises a significant amount of network traffic records. In order to assure precise evaluation of the Intrusion Detection System's performance, the dataset is partitioned into two main subsets: one for training and the other for testing.
2. **Appropriate Ratios for Training and Testing:** The allocation of data between the training and testing subsets is determined by appropriate ratios. It is common practice to assign a larger portion of the data for training, typically around 80%, and a smaller portion for testing, usually around 20%. This division ensures that the Intrusion Detection System (IDS) is effectively trained while still having enough data for evaluation.
3. **Maintaining Class Distribution:** It is crucial to maintain the same class distribution in both the training and testing subgroups in order to prevent bias. Within the framework of the "kddcup99" dataset, it is crucial to maintain similar proportions of normal and intrusive network connections in both subsets. This aids the Intrusion Detection System (IDS) in acquiring knowledge from a well-rounded portrayal of network activities and then assessing its effectiveness on a varied range of connections.

7.4 Integration of SVM and Genetic Algorithm

The integration of Support Vector Machine (SVM) and Genetic Algorithm (GA) is a pivotal component of the Intrusion Detection System (IDS) developed using the "kddcup99" dataset. This collaboration enhances the performance of the IDS by utilizing GA to optimize SVM parameters, leading to more accurate intrusion detection. The genetic algorithm (GA) is used to find the best value of the hyperparameter C for the SVM that maximizes the accuracy on a given dataset. Let's integrate the SVM and the genetic algorithm step by step:

The following figure shows the working principle of the machine learning model using the hybrid algorithm (GA & SVM) to detect network intrusion:
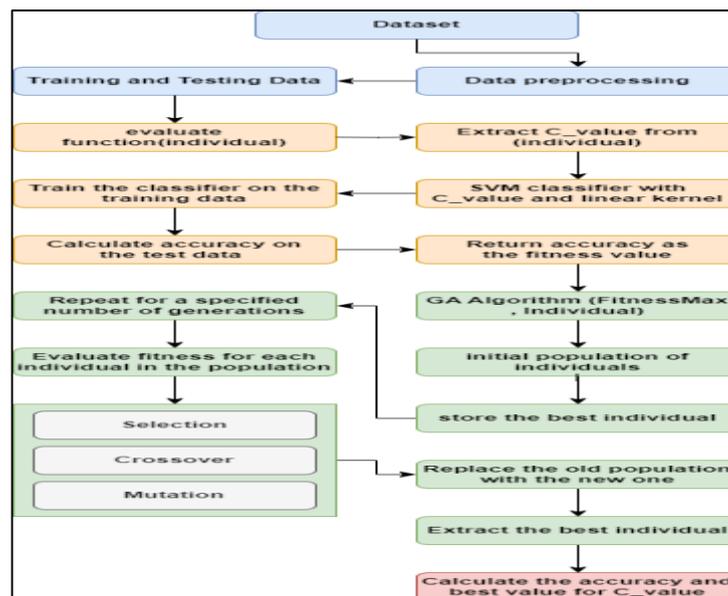

Fig.12. hybrid algorithm (GA & SVM)

1. **SVM Setup:** The SVM is defined and trained within the `evaluate` function. It takes an individual (in this case, an individual is a single value representing the hyperparameter C) and uses that value to create an SVM with the specified C value and a linear kernel. The SVM is then trained on the training data (`X_train`, `y_train`), and its accuracy on the

test data (`X_test`, `y_test`) is computed using the `clf.score` method. This accuracy value is returned as the fitness value for the genetic algorithm to optimize.

2. **GA Setup:** The `creator` module is used to define classes for the fitness and individuals. Here, we are performing a single-objective maximization problem, so we define a `FitnessMax` class. The `toolbox` is created to store various tools required for the genetic algorithm process. Functions for generating attributes (hyperparameter values), individuals, and populations are registered. Also, the functions for evaluating, mating, mutating, and selecting individuals are registered.

3. **Genetic Algorithm Process:** A population of 50 individuals is initialized using the `toolbox population` method. A `HallOfFame` object (`hof`) is used to store the best individuals found during the optimization process. A `Statistics` object (`stats`) is created to collect statistics about the population during the optimization process, such as average, standard deviation, minimum, and maximum fitness values. The `algorithms eaSimple` function is used to run the genetic algorithm. The optimization process is performed over 20 generations (`ngen=1`). Crossover (`cxpb=0.7`) and mutation (`mutpb=0.2`) probabilities are specified, along with the statistics and Hall of Fame objects.

4. Retrieving the Best Individual: After the genetic algorithm completes, the best individual is retrieved from the Hall of Fame (`hof`). The best hyperparameter C value and its corresponding accuracy are extracted from this individual.

In summary, the integration of the SVM and the genetic algorithm involves using the genetic algorithm to optimize the hyperparameter C value for the SVM classifier. The genetic algorithm evolves a population of individuals representing different C values, and the fitness of each individual is determined by training and evaluating an SVM with that C value. The genetic algorithm's goal is to find the C value that maximizes the accuracy of the SVM on the test data.

7.5 Evaluation

After the genetic algorithm converges, apply the optimal hyperparameters to the SVM model whose performance is evaluated on the test dataset using intrusion detection metrics such as accuracy.

8. RESULTS AND COMPARATIONS

Between 1998 and 1999, a project sponsored by DARPA and MIT's Lincoln Laboratory generated standardized network traffic data to evaluate network intrusion detection systems, aiming to review and evaluate research activities in this field. The 99KDD dataset includes 41 features, normalized data, and 22 distinct attack types, classified into R2U, L2R, PROBE, and DOS [27].

TABLE II. FEATURES OF THE KDD CUP 99 DATASET

No	Feature name	No	Feature name
1	duration	22	Is_guest_login
2	protocol type	23	Count
3	service	24	Serror_rate
4	Src_byte	25	Rerror_rate
5	Dst_byte	26	Same_srv_rate
6	flag	27	Diff_srv_rate
7	land	28	Srv_count
8	Wrong_fragment	29	Srv_serror_rate
9	urgent	30	Rv_rerror_rate
10	hot	31	Srv_diff_host_rate
11	Num_failed_logins	32	Dst_host_count
12	Logged_in	33	Dst_host_srv_count
13	Num_compromised	34	Dst_host_same_srv_count
14	Root_shell	35	Dst_host_diff_srv_count
15	Su_attempted	36	Dst_host_same_src_port_rate
16	Num_root	37	Dst_host_srv_diff_host_rate
17	Num_file_creations	38	Dst_host_serror_rate
18	Num_shells	39	Dst_host_srv_serror_rate
19	Num_access_shells	40	Dst_host_rerror_rate
20	Num_outbound_cmds	41	Dst_host_srv_rerror_rate
21	Is_hot_login	42	

The dataset included 494,021 records, some records containing normal connectivity and various attack records[27], according to the following table:

TABLE III. THE NUMBER OF SAMPLES IN THE DATASET

Number of records	percentage	class
97278	19.6911%	Normal
391458	79.2391%	Dos
4107	0.8313%	Probe
1126	0.2279%	R2L
52	0.0105%	U2R
396743	80.3089%	Total attacks
494021	100%	Total records

8.1 Selected Feature Experiments Results

All experiments were performed on a 1.10 GHZ, Celeron(R) N4000 CPU, 4.0 GB RAM and Windows 11 operating system. The google colab Python Open Source, was used to make the experiments. Table3 presents the selected features that are considered important based on the proposed model rules for detecting the attacks.

TABLE IV. SELECTED FEATURE

No	No	Feature name	No	No	Feature name
1	1	duration	16	17	Num_file_creations
2	2	protocol type	17	18	Num_shells
3	3	service	18	19	Num_access_shells
4	4	Src_byte	19	22	Is_guest_login
5	5	Dst_byte	20	24	Serror_rate
6	6	flag	21	25	Rerror_rate
7	7	land	22	26	Same_srv_rate
8	8	Wrong_fragment	23	27	Diff_srv_rate
9	9	urgent	24	31	Srv_diff_host_rate
10	10	hot	25	32	Dst_host_count
11	11	Num_failed_logins	26	33	Dst_host_srv_count
12	12	Logged_in	27	35	Dst_host_diff_srv_rate
13	13	Num_compromised	28	36	Dst_host_same_src_port_rate
14	14	Root_shell	29	37	Dst_host_srv_diff_host_rate
15	15	Su_attempted			

8.2 Compare Model(SVM) and hybrid Model (SVM & GA)

The SVM algorithm was applied to the kddcup99 data set after the data was processed and relevant features were selected. The SVM model was compared with the hybrid model (SVM & GA) on ten thousand records from the data set, and the results appeared as shown in the following table:

TABLE V. COMPARE MODEL 9SVM) AND HYBRID MODEL (SVM & GA)

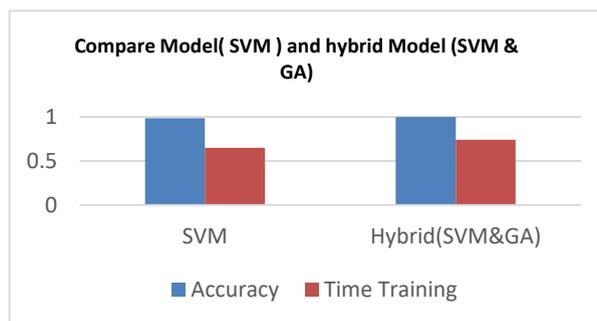| Training set (10000)		
Type Model	**Accuracy**	**Time Training**
SVM (42 features)	0.986	0.65 s
hybrid (SVM&GA) (29 features)	0.999	0.74 s

Fig.13. Compare Model(SVM) and hybrid Model (SVM & GA)

8.3 Performance comparison of improved intrusion detection algorithms based on GA , SVM and SVM algorithm

The selected features from the dataset were run on the SVM algorithm and the hybrid model of (SVM algorithm and genetic algorithm) to find the true positive rate (TPR) which is the percentage of samples that the classifier correctly predicts in all samples with a positive class. According to the following mathematical formula [28]:

$$True\ positive\ rate(TPR)=\ TP/(TP+FN)\ldots\ldots\ldots(1)$$

Likewise, finding the false negative rate (FNR) is the percentage of samples that the classifier incorrectly predicts in all positive samples, and is calculated using a specific formula [28]:

$$False\ negative\ rate(FNR)=\ FN/(TP+FN)\ldots\ldots\ldots(2)$$

The results appeared very close, that is, there are no discrepancies in them, as shown in the following table:

TABLE VI. COMPARISON OF PERFORMANCE BETWEEN (SVM-GA)and SVM ALGORITHM

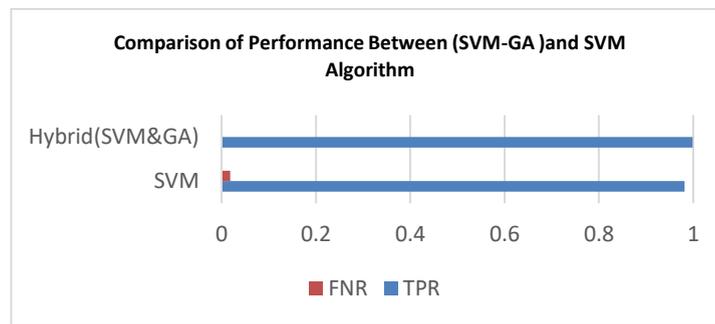| Algorithms	TPR	FNR
SVM	0.9814	0.0185
hybrid (SVM&GA)	0.9987	0.0012

Fig.14. Comparison of Performance Between (SVM-GA)and SVM Algorithm

8.4 Discussion

The experimental results we conducted on this model show that it has excellent intrusion detection performance. The model achieved a high accuracy rate of 99.9%, meaning that it is able to accurately identify most instances that indicate the presence of intrusion. In addition, its true positive rate of 99.8% reflects its ability to effectively identify actual intrusion events.

It is worth noting that the false negative rate was small at 0.2%. This indicates that the model significantly reduces the probability of generating false or false alarms. This low negative error rate enhances confidence in intrusion detection systems that use this model.

However, we should note that this model requires a significant amount of time to train, especially if the data is very large. This aspect can be a practical challenge, especially if detection speed is vitally important in practice. We must keep in mind that directing efforts to improve and improve the training processes may be necessary to improve the efficiency of this model and make it more usable in environments where the data is huge and diverse.

In summary, it can be said that this model has great potential for practical application as an intrusion detection tool, but requires attention to improving training processes and making better use of it in protection and security contexts.

9. CONCLUSION

This study highlights the significance of employing machine learning models for the development of an intrusion detection system on network traffic. Specifically, the study emphasizes the utilization of machine learning algorithms, such as the

genetic algorithm and SVM algorithm, to mitigate the escalating network attacks. The study examined the model's performance in terms of accuracy, classification time, positive and negative rates for intrusion detection. The findings indicated that the system achieved a high level of accuracy in detecting intrusions and exhibited a significantly low error rate. Moreover, the model outperformed the SVM algorithm when used alone. The evaluation was conducted using the kddcup99 dataset, which is widely utilized in various models for intrusion detection purposes.

Conflicts Of Interest

The author's affiliations, financial relationships, or personal interests do not present any conflicts in the research.

Funding

Acknowledgment

References

[1] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, vol. 36, pp. 16–24, 2013. [Online]. Available: https://doi.org/10.1016/J.JNCA.2012.09.004

[2] V. S and M. S, "Intrusion Detection System – A Study," International Journal of Security, Privacy and Trust Management, vol. 4, no. 1, pp. 31–44, 2015. [Online]. Available: https://doi.org/10.5121/ijsptm.2015.4104

[3] H. Debar, "An Introduction to Intrusion-Detection Systems," [Online]. Available: https://www.researchgate.net/publication/228589845

[4] "Home Entry Topic Review Support Vector Machine," 2014. [Online]. Available: https://handwiki.org/wiki/index.php?curid=2048019

[5] M. A. Almaiah et al., "Performance Investigation of Principal Component Analysis for Intrusion Detection System Using Different Support Vector Machine Kernels," Electronics (Switzerland), vol. 11, 2022. [Online]. Available: https://doi.org/10.3390/electronics11213571

[6] J. Zhang, "Network Security Situational Awareness Based on Genetic Algorithm in Wireless Sensor Networks," Journal of Sensors, 2022. [Online]. Available: https://doi.org/10.1155/2022/8292920

[7] I. H. Sarker et al., "Cybersecurity data science: an overview from machine learning perspective," Journal of Big Data, vol. 7, no. 1, 2020. [Online]. Available: https://doi.org/10.1186/s40537-020-00318-5

[8] J. Livingston, "What is Critical infrastructure cyber security?," Verve, 2022.

[9] "Critical Infrastructure Security and Resilience," Cybersecurity & Infrastructure Security Agency, CISA.gov.

[10] "What is network security?," Network security solutions protect computer systems from internal and external security threats and cyberattacks. www.ibm.com.

[11] "What is Cloud Security?," skyhigh security, www.skyhighsecurity.com.

[12] "Internet of Things (IoT) Security - ITSAP.00.012," 2022. www.cyber.gc.ca.

[13] M. Rouse, "What is cybersecurity? Definition, frameworks, and best practices," SearchSecurity. [Online]. Available: https://searchsecurity.techtarget.com/definition/cybersecurity

[14] McAfee, "Threats Report: November 2020," McAfee Labs. [Online]. Available: https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-nov-2020.pdf

[15] A. Pinto et al., "Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure," Sensors, vol. 23, no. 5, 2023. [Online]. Available: https://doi.org/10.3390/s23052415

[16] M. Rani, "A Review of Intrusion Detection System in Cloud Computing," International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM-2019). [Online]. Available: https://ssrn.com/abstract=3355127

[17] E. Tufan, C. Tezcan, and C. Acartürk, "Anomaly-based intrusion detection by machine learning: A case study on probing attacks to an institutional network," IEEE Access, vol. 9, pp. 50078–50092, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3068961

[18] "2019 International Conference on Computer and Information Sciences (ICCIS)," Jouf University - Aljouf - Kingdom of Saudi Arabia, 03-04 April 2019.

[19] A. Khraisat et al., "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, vol. 2, no. 1, 2019. [Online]. Available: https://doi.org/10.1186/s42400-019-0038-7

[20] T. Saranya et al., "Performance Analysis of Machine Learning Algorithms in Intrusion Detection System: A Review," Procedia Computer Science, vol. 171, pp. 1251–1260, 2020. [Online]. Available: https://doi.org/10.1016/j.procs.2020.04.133

[21] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 297, no. 3, pp. 273–297, 2009. [Online]. Available: Researchgate. DOI: 10.1007/%2FBF00994018

[22] "Applications of Genetic Algorithms in Machine Learning," [Online]. Available: https://www.turing.com/kb/genetic-algorithm-applications-in-ml#applications-of-genetic-algorithm-in-machine-learning

[23] C. M. Bishop, "Pattern Recognition and Machine Learning," Springer, 2006.

[24] M. Al-Shalabi et al., "Energy efficient multi-hop path in wireless sensor networks using an enhanced genetic algorithm," Information Sciences, vol. 500, pp. 259–273, 2019. [Online]. Available: https://doi.org/10.1016/j.ins.2019.05.094

[25] J. Carr, "An Introduction to Genetic Algorithms," 2014.

[26] "Applications of Genetic Algorithms in Machine Learning," [Online]. Available: https://www.turing.com/kb/genetic-algorithm-applications-in-ml#applications-of-genetic-algorithm-in-machine-learning

[27] H. Alahmad and Tishreen University Journal for Research and Scientific Studies -Engineering Sciences Series, "Using Neural Networks to Build an Intrusion Detection System based on Standard Dataset (KDD99)," Issue 93, 2017.

[28] "A Hybrid IDS Using GA-Based Feature Selection Method and Random Forest," International Journal of Machine Learning and Computing, vol. 12, no. 2, 2022. [Online]. Available: https://doi.org/10.18178/ijmlc.2022.12.2.1077.