Journal of Computer Science

Original Research Paper

An Ensemble of Gaussian Mixture Model and Support Vector Machines for Network Intrusion Detection

[1]Olujimi Daniel Alao, [2]Sheriff Alimi, [2]Shade Oluwakemi Kuyoro, [2]Ruth Chinkata Amanze,
[3]Adesina Kamorudeen Adio and [2]Michael Oluwagbenga Agbaje

[1]Department of Information Technology, Babcock University, Illishan, Nigeria
[2]Department of Computer Science, Babcock University, Illishan, Nigeria
[3]Department of Basic Sciences, Babcock University, Illishan, Nigeria

Article history
Received: 12-04-2022
Revised: 17-08-2022
Accepted: 26-08-2022

Corresponding Author:
Olujimi Daniel Alao
Department of Information Technology, Babcock University, Illishan, Nigeria
Email: alaool@babcock.edu.ng

Abstract: Network Intrusion Detection Systems (NIDS) can protect computer networks and computer systems by detecting abnormal network packets and taking agreed action plans, such as notifying an administrator or rejecting the network packets. In this study, the aim is the implementation of NIDS with improved performance using an ensemble of Support Vector Machines (SVMs) and the Gaussian Mixture Model (GMM). Four SVMs with Radial Basis Function (RBF), linear, polynomial, and sigmoid kernel functions, and a GMM were trained with the same portion with Knowledge Discovery and Data Mining Tools Competition (KDD 99) dataset, and another portion of the dataset was used to evaluate the performance of the respective NIDS models. Finally, the five models were integrated to form an ensemble Intrusion Detection System (IDS) model and the same test dataset was used to validate its performance. The IDS model of SVM with RBF kernel function has the best performance with precision, recall, f1 score, accuracy, false acceptance rate, and false rejection rate of 99.88, 99.67, 99.77, 99.82, 0.08, and 0.33% respectively. The ensemble model built by combining the five trained models where each of them has equal voting rights yields state-of-art performance, precision, recall, f1-score, accuracy, false acceptance rate, and false rejection rate of 99.7, 99.4, 99.55, 99.65, 0.18 and 0.59% respectively though it is below the performance of the SVM-RBF and the SVM-polynomial models. Ensemble models are expected to have better performance than a single classifier, but the result of this research shows that this is not applicable in all cases as the SVM with RBF kernel outperformed the ensemble classifier.

Keywords: Network Intrusion Detection, Gaussian Mixture Model, Support Vector Machines, Performance Metrics

Introduction

Internet or cyber security is concerned with technologies and procedures applied to both networks and computer systems to guarantee the availability, confidentiality/privacy, and integrity of information assets and computer services by protecting them from vulnerabilities and threats. Networks today has extended beyond connected computers to include Internet-of-things and vehicular ad hoc network (Bangui and Buhnova, 2021; Sarker *et al*., 2020).

A successful attack on an organization has a far-reaching effect that could lead to disruption of business operation which implies revenue loss. Such an attack can also result in reputation or brand damage, loss of customers, and diminishing organization goodwill. According to an IBM report, the cost of the data breach between August 2019 and April 2020 in the United States is estimated to be $3.86 million and 52% of the breach was caused by malicious attacks (IBM, 2020).

Threats come in form of various attacks such as probing attacks, denial-of-service attacks, Remote-to-Local (R2L) attacks, and User-to-Root (U2R) attacks. Other forms of attack are viruses, trojans, and worms.

With a probing attack, the hacker sends a well-crafted packet to scan the port of the target system to reveal vulnerabilities while Denial-of-service is primarily concerned with flooding the destination systems with requests more than it can handle which makes the system unable to service genuine requests and, in some cases, shut down the computer. A user-to-root attack is a situation where a hacker has gained access to a system and attempts to gain privileged access like a root user so that more damaging havoc can be done. The use of invalid users by password guessing to gain access to a system is also a form of attack and it is called a remote-to-local attack (Chen *et al*., 2016).

Virus attacks can be dealt with irrespective of viral strategies adopted ranging from polymorphic viruses to sparse infection viruses with the use of anti-virus software which will be more effective by regular updates with the on-access scanner option.

The firewall sits at the edge of the network to protect the computers behind it from attack by inspecting the packets and filtering inappropriate ones such as saturation packets. On occasions when the attacks were undetected by the firewall, an additional security layer introduced is the IDS. The intrusion detection system can protect both the computer network and computer system by detecting abnormal network packets or abnormal activities on the systems and taking an agreed action plan, such as notifying an administrator, and in the case of network packets, such packets can be rejected as seen in a variant of IDS called Intrusion Prevention System (IPS). It can protect against DoS attacks, probing attacks, U2R attacks, and R2L attacks. There are two types of IDS, the network-based IDS for network security protection and the host-based IDS installed on a system to monitor the system audit log for abnormal activity detection. Figure 1 below shows the position of IDS in an organization's edge network.

The two methods for implementing IDS are anomaly-based and misuse-based approaches, combining these two leads to a hybrid method. The misuse-based detection uses the characteristics/signature of past attacks to detect new ones while the anomaly-based established a model of what is normal behavior and any deviation from such is considered as abnormal; abnormal in terms of the network packet seen as an attack or intrusion while in case of host-based implementation, it is seen as activity perpetrated by an intruder. The anomaly-based detection uses machine learning algorithms that are trained with relevant datasets and are more effective compared to signature-based counterpart that does not perform well with new attack signatures.

This study is focused on exploring anomaly-based network intrusion detection systems with an ensemble of Support Vector Machine (SVM) and Gaussian Mixture Model (GMM) using the KDD 99 dataset. The ensemble model comprises a Gaussian mixture model and four SVM models based on RBF, polynomial, linear and sigmoid functions; the main objective is to examine if the ensemble can help improve the performance of IDS against individual model performance.

Several machine learning algorithms have been used for implementing anomaly intrusion detection systems for network traffic and a couple of datasets such as KDD 99 and Information of Excellent (ISCX) were used to evaluate the various models and many of them yielded convincing results.

The volume of network data generated to be processed by IDS is quite high creating a bottleneck for the server infrastructure, Chen *et al*. (2016) in their work used compressed sensing to achieve dimensional reduction to reduce the volume of network traffic data to eliminate the infrastructure bottleneck and support vector machine was then used for the IDS model. Similarly, Sarker *et al.* (2020) achieved a reduction in the dimension of the features used in the training tree-based intrusion detection model by focusing on important features of the security data.

Bangui and Buhnova (2021) examined the performance of various machine learning algorithms for detecting intrusion, the result shows that Decision-tree had the best accuracy over K-Nearest Neighbor, Logistic Regression, K-means, Stochastic Gradient Descent while Gaussian Naive Bayes recorded the worst performance across all types of attack.

In an implementation of anomaly-based intrusion detection, Aldwairi *et al*. (2018) explored the Restricted Boltzmann Machine (RBM) model for discriminating between normal and anomaly network traffic and the Information of Excellent (ISCX) dataset was used because it is a realistic dataset. The best performance accuracy recorded was 89.7% and the corresponding True Positive Rate and True Negative Rate at that instance are 89.2 and 93.9% respectively.

Resende and Drummond (2018) in their adaptive anomaly-based intrusion detection system used a genetic algorithm to select features for profiling and the fitness function is dependent on both True Positive and False Positive, overall performance recorded for detection rate and false detection rate are 92.85 and 0.69% respectively with the Intrusion detection evaluation dataset CICIDS2017 dataset. Similar work by Kalavadekar and Sane (2018) also confirmed the effectiveness of the genetic algorithm in the selection of features for anomaly intrusion detection systems.

Shin and Kim (2020) in their work, explored the performance of SVM, logistic regression, and K-Nearest Neighbour (KNN) in the implementation of host-based anomaly intrusion detection systems, the result shows that the SVM had the best Area Under the Curve (AUC) and second-best accuracy performance, especially with the use of RBF kernel function.

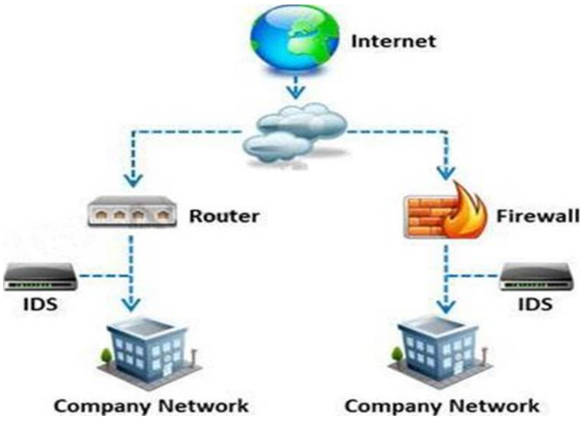

Fig. 1: Position of IDS in an organization's edge network

The use of autoencoder in the pre-training stage to achieve dimensional reduction was considered by Mennour and Mostefai (2020) in their network-based intrusion detection system and the output was then used to train Deep Neural Network (DNN). Mennour and Mostefai (2020) made use of CICIDS2017 datasets in their work and the result was better than two state-of-art IDS.

According to the work done by Tama *et al.* (2019), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and genetic algorithm were used to extract features from NSL-KDD and UNSW-NB15 datasets, and a two-stage meta-classifier was used to achieve anomaly-based intrusion detection system.

Gaussian Mixture Model and Support Vector Machine Theoretical Principles

This section covers the theoretical principles guiding the workings of both the Gaussian Mixture Model and the Support Vector Machine.

A. Gaussian Mixture Model

This can be considered as an extension of k-means in which clusters are modeled based on Gaussian distribution. It makes use of the mean and also the covariance of the features which describe their ellipsoidal shape. The fitting of the model is done by maximizing the likelihood of the data with Expectation-Maximization (EM) which is like K-means except that data is assigned to a cluster by soft probability.

For dimension vector $x,\left(where\, x=\left\{x^1,x^2,...,x^d\right\}\right)$, the gaussian probability distribution function is defined by Eq. (1) below:

$$\mathrm{N}(x|u,\varepsilon)=\frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{\varepsilon}}e^{-\frac{(x-u)_T}{2}\ \varepsilon^{-1}(x-u)} \tag{1}$$

Here, u, ε are the mean and covariance matrix of the Gaussian.

The Gaussian Mixture Model (GMM) is the weighted sum of the number of probability distribution functions and weight π is determined by distribution. GMM is expressed with Eq. (2) and (3) below:

$$\begin{aligned}p(x)&=\pi_0\mathrm{N}\left(x\,|\,u_{0,}\varepsilon_0\right)+\pi_1\mathrm{N}\left(x\middle|u_{1,}\,\varepsilon_1\right)\\&\quad...\ \pi_{k-1}\mathrm{N}\left(x\middle|u_{k-1,}\varepsilon_{k-1}\right)+\pi_k\mathrm{N}\left(x\middle|u_{k,}\varepsilon_k\right)\end{aligned} \tag{2}$$

and $\sum_{k=0}^{k}\pi_k=1$

Hence:

$$p(x)=\sum_{k=0}^{k}\ \pi_k\mathrm{N}\left(x\,|\,u_{k,}\varepsilon_k\right) \tag{3}$$

The training of GMM is done by using EM which involves two iterative steps which are the algorithm Expectation or E-step and Maximization or M steps.

B. EM Algorithm

Start with the clusters: Mean u_k, covariance, ε_k and the size π_k.

Start with an assignment of π_k.

E Steps (Expectation)

For each of the data point x_i, compute the probability of being a member of the cluster k as seen in Eq. (4):

- Compute the probability
- Normalize to sum 1 over the k clusters:

$$r_{ik}=\frac{\pi_k\mathrm{N}(x|u_{k,}\varepsilon_k),}{\sum_{i'}\pi_{k'}\mathrm{N}(x|u_{k',}\,\varepsilon_{k'})} \tag{4}$$

M Steps (Maximization)

For each of the clusters (Gaussian), its parameters are updated using the weighted data points

- $m_k=\sum\ r_{ik}$ (this is the total probability allocations to the cluster)
- $\pi_k=\frac{m_k}{m}$ (this is the fraction of the total data point assigned to cluster k)
- $u_k=\frac{1}{m_k}\sum\ r_{ik}\ x_i$ (updated means of the cluster)
- $\varepsilon_k=\frac{1}{m_k}\sum\ r_{ik}(x_i-u_k)^T(x_i-u_k)$ which represents the updated covariance of the cluster

Log-Likelihood Computation

For each iteration, the GMM likelihood is computed (with the Eq. (5) below) and the iterations of the EM algorithm are stopped when the value converges. Each iteration increases the log-likelihood of the model:

$$\log\mathrm{P}(x)=\sum_i\log\left[\sum_{k=0}^{k}\pi_k\mathrm{N}(x|u_{k,}\,\varepsilon_k)\right] \tag{5}$$

C. Support Vector Machine

A support vector machine is a supervised learning model that focuses on maximizing the distance between data points at the boundaries of two different classes and that explains why it is referred to as a large-margin classifier (Parikh and Shah, 2016).

The two classes are +1 and -1 and x is d dimension feature vector with class label y where vector instance x_i has a corresponding y_i class label. Equation (6) and (7) below are the hyperplane equations of the two classes:

$$y_i = +1; wx^{+}_{i} + b \ge 1 \tag{6}$$

$$y_i = -1; wx^{-}_{i} + b \ge -1 \tag{7}$$

The *w* and *b* are the weight and constant of the equations. Simplifying Eq. (6) and (7) yields:

$$y_i(wx_i + b) \ge 1 \tag{8}$$

Subtracting Eq. (7) from (6) gives (9) below:

$$x^{+} - x^{-} = \frac{2}{w} \tag{9}$$

Further refinement of Eq. (8) and (9) result into constrain optimization problem below:

$$\min imize\ f(w,b) = \frac{\|w\|^2}{2}$$

$$subject\ to: g(w,b) = y_i(wx_i + b) \ge 1 \tag{10}$$

Application of Lagrange multiplier to the constrained optimization problem of Eq. (10) yields in Eq. (6) and (7) is best suited to solve the constrained optimization problem:

$$L(w,b,\alpha) = \frac{\|w\|^2}{2} - \sum_{i=1}^{m} \alpha\left[y_i\left(wx_i + b\right)\right] \tag{11}$$

$$\nabla L(w,b,\alpha) = \nabla\frac{\|w\|^2}{2} - \nabla\left(\sum_{i=1}^{m} \alpha_i\left[y_i(wx_i + b)\right]\right) = 0 \tag{12}$$

From the above we can deduce:

$$W = \sum_{i=1}^{m} \alpha_i y_i x_i$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$b = y_i - wx_i$$

The training of the SVM model ensures that the best values of *w* and *b* are obtained in realizing the binary classifier.

An important learning parameter in an SVM classifier is the kernel function and the choice of kernel function used determines the performance of the model. Some of the functions are linear, polynomial, and RBF kernel functions (Huang *et al*., 2017).

Materials and Methods

In this study, KDD 99 dataset was used, the dataset comprises 9-week Transmission Control Protocol (TCP) dump connections and system audit, simulating different types of users, natural traffic, and attack techniques (Chen *et al*., 2016) and the dataset has been used extensively in intrusion detection system research (Özgür and Erdem, 2016).

The methodology consists of three stages:

A. The preprocessing stage and dividing datasets into training and validation sets
B. Training of each of the models and evaluation of their performance
C. Formation of ensemble network using the five models and evaluation of its performance using the validation dataset
D. Figure 2 below represents the first two stages of the methodology

A. Preprocessing

The dataset contains 42 fields that represent features. All the fields are numeric except for field 1, 2, 3, and 41. Field 1 is the protocol and it has the following unique values:

['tcp', 'udp', 'icmp']

Field 2 is the service type and it has the following unique values:

['http', 'smtp', 'finger', 'domain_u', 'auth', 'telnet', 'ftp', 'eco_i', 'ntp_u', 'ecr_i', 'other', 'private', 'pop_3', 'ftp_data', 'rje', 'time', 'mtp', 'link', 'remote_job', 'gopher', 'ssh', 'name', 'whois', 'domain', 'login', 'imap4', 'daytime', 'ctf', 'nntp', 'shell', 'IRC', 'nnsp', 'http_443', 'exec', 'printer', 'efs', 'courier', 'uucp', 'klogin', 'kshell', 'echo', 'discard', 'systat', 'supdup', 'iso_tsap', 'hostnames', 'csnet_ns', 'pop_2, 'sunrpc', 'uucp_path', 'netbios_ns', 'netbios_ssn', 'netbios m', 'sql_net', 'vmnet', 'bgp', 'Z39_50', 'ldap', 'netstat', 'urh_i', 'X11', 'urp_i', 'pm_dump', 'tftp_u', 'tim_i', 'red_i']

Field 3 has the under-listed possible values:

['SF', 'S1', 'REJ', 'S2', 'S0', 'S3', 'RSTO', 'RSTR', 'RSTOS0', 'OTH', 'SH']

Lastly, field 41 is the attack categories:

['normal.', 'buffer_overflow.', 'loadmodule.', 'perl.', 'neptune.', 'smurf.', 'guess_passwd.', 'pod.', 'teardrop.', 'portsweep.', 'ipsweep.', 'land.', 'ftp_write.', 'back.', 'imap.', 'satan.', 'phf.', 'nmap.', 'multihop.', 'warezmaster.', 'warezclient.', 'spy.', 'rootkit.']

It shows clearly that features or fields 1, 2, 3, and 41 are categorical values which are strings and it is mandatory to convert them to numerical values before they can be useful in building a machine learning model.

The following are the conversion performed on each of the fields in a dictionary data structure:

protocol = {'tcp':1, 'udp':2, 'icmp':3}

service = {'http':1, 'smtp':2, 'finger':3, 'domain_u':4, 'auth':5, 'telnet':6, 'ftp':7,'eco_i':8, 'ntp_u':9, 'ecr_i':10, 'other':11, 'private':12, 'pop_3':13, 'ftp_data':14,'rje':15, 'time':16, 'mtp':17, 'link':18, 'remote_job':19, 'gopher':20, 'ssh':21,'name':22, 'whois':23, 'domain':24, 'login':25, 'imap4':26, 'daytime':27, 'ctf':28,'nntp':29, 'shell':30, 'IRC':31, 'nnsp':32, 'http_443':33, 'exec':34, 'printer':35,'efs':36, 'courier':37, 'uucp':38, 'klogin':39, 'kshell':40, 'echo':41, 'discard':42,'systat':43, 'supdup':44, 'iso_tsap':45, 'hostnames':46, 'csnet_ns':47, 'pop_2':48,'sunrpc':49, 'uucp_path':50,'netbios_ns':51,'netbios_ssn':52,'netbios_dgm':53,'sql_net':54, 'vmnet':55, 'bgp':56, 'Z39_50':57, 'ldap':58, 'netstat':59, 'urh_i':60,'X11':61, 'urp_i':62, 'pm_dump':63, 'tftp_u':64, 'tim_i':65, 'red_i':66}

flag= {'SF':1, 'S1':2, 'REJ':3, 'S2':4, 'S0':5, 'S3':6, 'RSTO':7, 'RSTR':8, 'RSTOS0':8,'OTH':10, 'SH':11}

Regarding the record classification, the class tagged 'normal' is assigned value '0' while others that represent different attacks are assigned '1'.

After the conversion of the categorical fields to numerical equivalent based on the previous steps, the dataset is normalized and broken into training and validation sets in ratios of 70 and 30% respectively.

B. Training Stage and Evaluation

Four SVMs with RBF, polynomial, linear, and sigmoid kernel functions were trained respectively with the training dataset. Similarly, the GMM was also trained with the same dataset.

Each of the five models was evaluated based on their respective predictions against the validation dataset. An ensemble prediction system was then built using the Bagging technique and the problem at hand being a classification problem, the class group is predicted by taking the mode of the classifiers' predictions with each classifier having equal voting contributions.

The performance of the ensemble-based IDS is obtained by evaluation of its classification capabilities against the validation/test dataset and the result is compared with that of GMM, SVM-RBF, SVM-POL, SVM-LINEAR, and SVM-SIG classifiers. The validation approach is depicted in Fig. 3.

E. Evaluation Metrics

Standard machine learning metrics are deployed to evaluate the performance of the various models and the metrics are accuracy, precision, recall, f1-score, False Acceptance Rate (FAR), and False Rejection Rate (FRR).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

where, *TP* = True Positive, *TN* = Time Negative, *FP* = False Positive, *FN* = False Negative.

False Acceptance Rate (FAR) is the ratio of the number of times that attacks were classified as non-attacks while False Rejection Rate (FRR) is the ratio of the number of times non-attacks were classified as attacks.

$$FAR = \frac{FP}{FP + TN}$$

$$FRR = \frac{FN}{FN + TP}$$

Algorithm Steps

The entire process can be represented with the algorithm below:

1) Dataset is divided into training and test sets in ratios of 70 to 30
2) Five classifiers (GMM, SVM-RBF, SVM-POL, SVM-LINEAR, and SVM-SIG) were separately trained with the training dataset
3) The test dataset was applied to the five classifiers to obtain their respective predictions
4) The predictions of the five classifiers are aggregated to produce a single output using the mode function i.e., returns the most frequent prediction (this is a Bagging ensemble method)
5) Evaluation of the aggregated model's or ensemble classifier's prediction with the test dataset label reference point using metrics such as accuracy, F1-score, precision, False acceptance rate, etc
6) Evaluation of the performance of individual classifiers separately

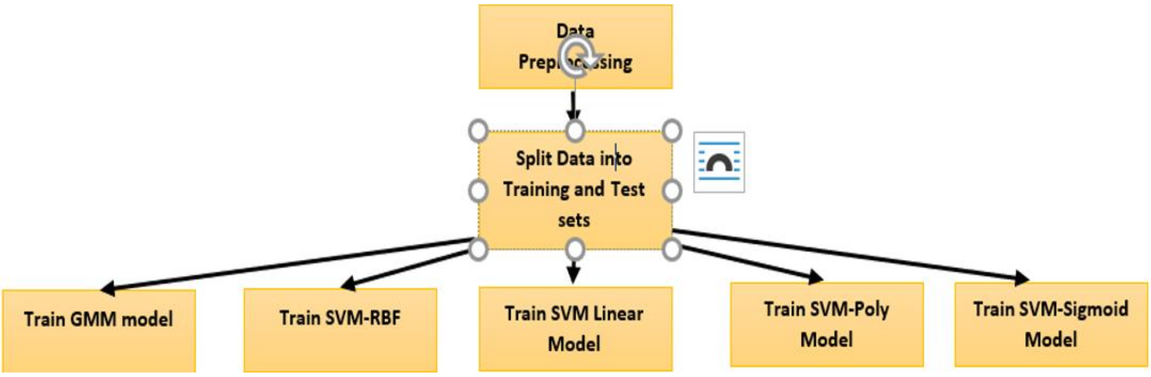

Fig. 2: The first two stages of the methodology

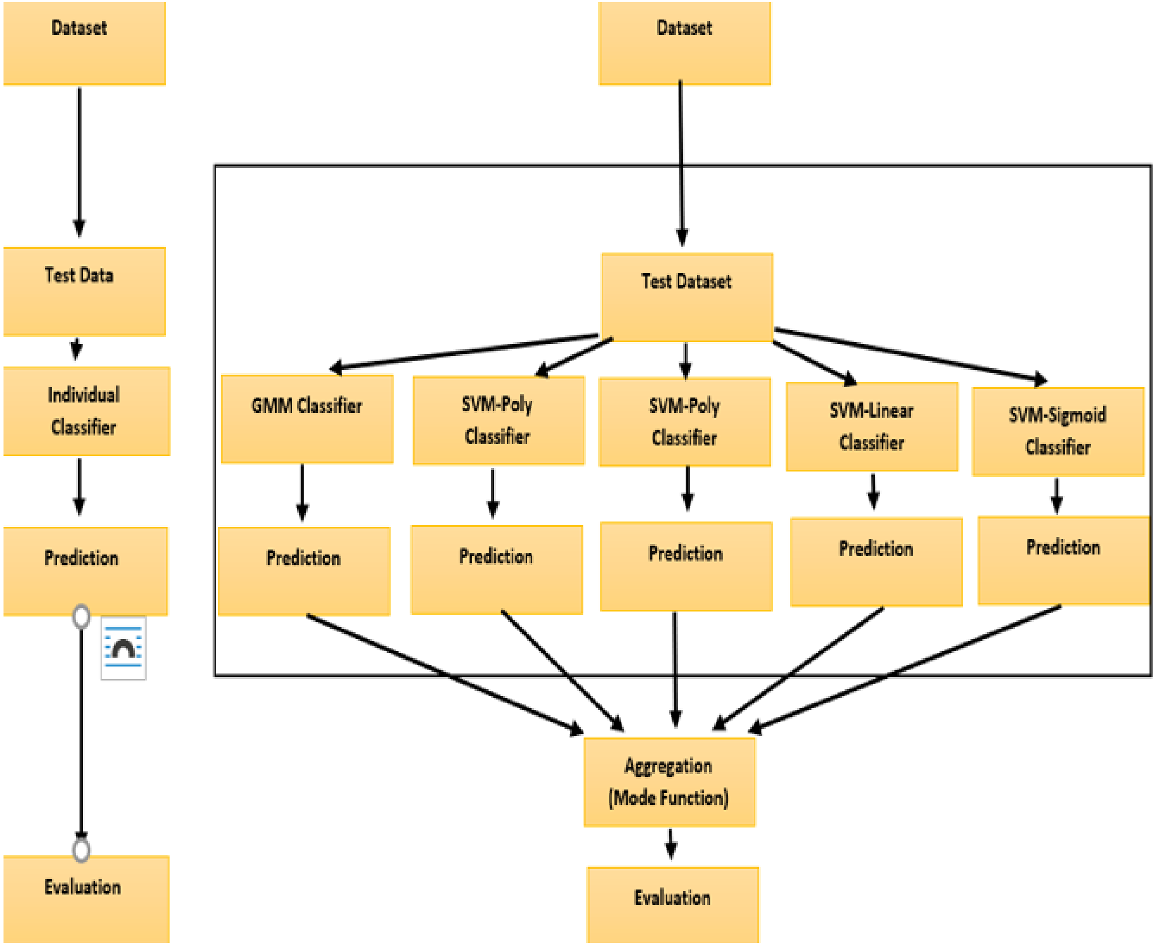

Fig. 3: Evaluation of Individual classifiers and the bagging ensemble model

Results and Discussion

The original KDD 99 dataset contains 494,021, after the removal of duplicates; the records were reduced to 145585. 70% of the records were used for training each of the models and the remaining 30% were used as a validation dataset.

Table 1 through to Table 6 below are the confusion matrixes for SVM-RBF (RBF kernel), SVM-POLY (Polynomial kernel), SVM-Linear, SVM-SIG (Sigmoid kernel function), GMM (Gaussian Mixture Model), and the ensemble model that comprises the five models.

Each of the Table 2 to 5 contains the values of TP, FP, TN, and FN which are the input for computing accuracy, precision, f1-score, recall, FAR, and FRR for the models.

Table 7 shows that the IDS built using SVM with RBF kernel function has the best performance with precision, recall, f1-score, accuracy, false acceptance rate, and false rejection rate of 99.88, 99.67, 99.77, 99.82, 0.08, and 0.33% respective.

The SVM with polynomial function has performance metrics that are almost at par with that of the RBF function. The Gaussian Mixture Model (GMM) had the worst performance, especially for the false acceptance rate and false rejection rate of 20.04 and 18.43%.

The ensemble model built by combining the five trained models where each of them has equal voting rights yields state-of-art performance, precision, recall, f1 score, accuracy, false acceptance rate, and false rejection rate of 99.7,99.4,99.55,99.65,0.18 and 0.59% respectively though it is below the performance of the SVM-RBF and the SVM-polynomial models.

The best classification accuracy recorded in this research surpasses results from similar work. Chen *et al*. (2016) IDS based on compressed sensing and SVM achieved an accuracy of 99.01% on KDD CUP 99 dataset. An accuracy of 85% was also achieved on the NSL-KDD dataset by Tama *et al*. (2019) using a two-level classifier ensemble.

To compare the result of this study with the other two related works, the True Positive Rate (Sensitivity) must be computed for the SVM classifier (with RBF kernel). The expression for True Positive Rate (TPR) is shown below:

$$TPR = \frac{TP}{TP + FN}$$

The TPR of the SVM-RBF classifier is 99.67%, which is the best compared with the two related works in Table 8 below.

Table 1: SVM-RBF confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	17,301	58
	Negative	20	26,297

Table 2: SVM-POLY confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	17,302	57
	Negative	22	26,295

Table 3: SVM-Linear confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	17,158	201
	Negative	110	26,207

Table 4: SVM-SIGMOID confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	15,280	2,079
	Negative	2,162	24,155

Table 5: GMM confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	14,160	3,199
	Negative	5,274	21,043

Table 6: Ensemble model confusion matrix

		Prediction	
		Positive	Negative
Actual	Positive	17,256	103
	Negative	48	26,269

Table 7: Performance metrics for all models

Models	Precision	Recall	F1 Score	Accuracy	FAR	FRR
SVM-RBF	99.88%	99.67%	99.77%	99.82%	0.08%	0.33%
SVM-SIGMOID	87.60%	88.02%	87.81%	90.30%	8.22%	11.98%
SVM-POLYNOMIAL	99.87%	99.67%	99.10%	99.82%	0.08%	0.33%
SVM-LINEAR	99.36%	98.84%	99.10%	99.29%	0.42%	1.16%
GMM	72.80%	81.57%	76.95%	80.60%	20.04%	18.43%
SVMs(RBF+SIG+POLY+LINEAR) + GMM	99.70%	99.40%	99.55%	99.65%	0.18%	0.59%

Table 8: Comparing the TRP of current work with related works

Methods	Related works	Dataset	TPR
SVM with RBF	Current work	KDD 99	99.67%
Genetic Algorithm with Gaussian Distribution/K-means	Resende and Drummond (2018)	CICIDS2017	92.85%
Restricted Boltzmann Machine (RBM) model	Aldwairi *et al*. (2018)	ISCX	89.25

Conclusion

The performance results show that network intrusion detection systems built with Support Vector Machine using RBF kernel function (SVM-RBF) produced superior performance and it is closely followed by SVM with polynomial function (SVM-POLY). The performance of IDS based on the ensemble of the five models is below that of the SVM-RBF model across all the metrics. The SVM-RBF and SVM-Polynomial-based IDS yielded an encouraging performance considering various metrics used for evaluation in this study. Ensemble models are expected to have better performance than a single classifier, but the result of this research shows that this is not applicable in all cases as the SVM with RBF kernel outperformed the ensemble classifier.

Going forward, it is recommended that similar work should be done using CICDS2017 and ISCX datasets with consideration given to the conversion style adopted in this study to transform protocols, applications, and TCP flag strings to numerical values instead of using one-hot-encoding categorical conversion.

For practical utilization of the research outcome, the SVM-RBF classifier can be embedded into an edge router which is then transformed into an Intrusion Prevention System (IPS). Internally, the router passes the TCP header to the RBF-SVM classifier functioning as an Internal Network Detection (NIDS) engine, and if the packet header is classified as an attack, the entire packet will be rejected, otherwise, it is accepted.

Acknowledgment

We would like to sincerely appreciate the anonymous reviewers for their comments which have improved the work.

Author's Contributions

Olujimi Daniel Alao: Problem formulation, mathematical modeling, literature review, and editing.

Sheriff Alimi: Literature review, mathematical modeling, writing, and supervision.

Shade Oluwakemi Kuyoro: Literature review and data analysis.

Ruth Chinkata Amanze: Writing and mathematical modeling.

Adesina Kamorudeen Adio: Mathematical modeling and data analysis.

Michael Oluwagbenga Agbaje: Literature review, writing, and formatting.

Ethics

This article is an original research paper. There are no conflicts of interest and no ethical issues that may arise after the publication of this manuscript.

References

Aldwairi, T., Perera, D., & Novotny, M. A. (2018). An evaluation of the performance of Restricted Boltzmann Machines as a model for anomaly network intrusion detection. *Computer Networks*, *144*, 111-119. http//:doi.org/10.1016/j.comnet.2018.07.025

Bangui, H., & Buhnova, B. (2021). Recent advances in machine-learning driven intrusion detection in transportation: Survey. *Procedia Computer Science*, *184*, 877-886. http//:doi.org/10.1016/j.procs.2021.04.014

Chen, S., Peng, M., Xiong, H., & Yu, X. (2016). SVM intrusion detection model based on compressed sampling. *Journal of Electrical and Computer Engineering*, *2016*. http//:doi.org/10.1155/2016/3095971

Huang, M. W., Chen, C. W., Lin, W. C., Ke, S. W., & Tsai, C. F. (2017). SVM and SVM ensembles in breast cancer prediction. *PloS one*, *12*(1), e0161501. http//:doi.org/10.1371/journal.pone.0161501

IBM. (2020). "Cost of Data Breach Report 2020 (Highlights)." https://www.ibm.com/security/digital-assets/cost-data-breach-report/#/

Kalavadekar, M. P. N., & Sane, S. S. (2018). Building an effective intrusion detection system using genetic algorithm-based feature selection. *International Journal of Computer Science and Information Security (IJCSIS)*, *16*(7). https://sites.google.com/site/ijcsis/

Mennour, H., & Mostefai, S. (2020, June). A hybrid deep learning strategy for anomaly-based N-ids. In *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1-6). IEEE. http//:DOI.org/10.1109/ISCV49265.2020.9204227

Özgür, A., & Erdem, H. (2016). A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. http//:doi.org/10.7287/peerj.preprints.1954

Parikh, K. S., & Shah, T. P. (2016). Support vector machine–a large margin classifier to diagnose skin illnesses. *Procedia Technology*, *23*, 369-375. http//:doi.org/10.1016/j.protcy.2016.03.039

Resende, P. A. A., & Drummond, A. C. (2018). Adaptive anomaly-based intrusion detection system using genetic algorithm and profiling. *Security and Privacy*, *1*(4), e36. http//:doi.org/10.1002/spy2.36

Sarker, I. H., Abushark, Y. B., Alsolami, F., & Khan, A. I. (2020). Intrudtree: A machine learning-based cyber security intrusion detection model. *Symmetry*, *12*(5), 754. http//:doi.org/10.3390/sym12050754

Shin, Y., & Kim, K. (2020). Comparison of anomaly detection accuracy of host-based intrusion detection systems based on different machine learning algorithms. *International Journal of Advanced Computer Science and Applications*, *11*(2). http//:doi.org/10.14569/ijacsa.2020.0110233

Tama, B. A., Comuzzi, M., & Rhee, K. H. (2019). TSE-IDS: A two-stage classifier ensemble for an intelligent anomaly-based intrusion detection system. *IEEE Access*, *7*, 94497-94507. http//:doi.org/10.1109/access.2019.2928048