Comparison of Anomaly Detection Accuracy of Host-based Intrusion Detection Systems based on Different Machine Learning Algorithms

Yukyung Shin[1], Kangseok Kim[1, 2*]
Department of Data Science, Graduate School of Ajou University, Suwon, Korea[1]
Department of Cyber Security, Ajou University, Suwon, Korea[2]

***Abstract*—Among the different host-based intrusion detection systems, an anomaly-based intrusion detection system detects attacks based on deviations from normal behavior; however, such a system has a low detection rate. Therefore, several studies have been conducted to increase the accurate detection rate of anomaly-based intrusion detection systems; recently, some of these studies involved the development of intrusion detection models using machine learning algorithms to overcome the limitations of existing anomaly-based intrusion detection methodologies as well as signature-based intrusion detection methodologies. In a similar vein, in this study, we propose a method for improving the intrusion detection accuracy of anomaly-based intrusion detection systems by applying various machine learning algorithms for classification of normal and attack data. To verify the effectiveness of the proposed intrusion detection models, we use the ADFA Linux Dataset which consists of system call traces for attacks on the latest operating systems. Further, for verification, we develop models and perform simulations for host-based intrusion detection systems based on machine learning algorithms to detect and classify anomalies using the Arena simulation tool.**

***Keywords*—Anomaly detection; host based intrusion detection system; system calls; cyber security; machine learning; simulation**

I. Introduction

Owing to the recent developments in the fields of software, hardware, and mobile networks, as well as the proliferation of information services, such as social network services (SNS), people are now more closely connected to the Internet than ever before. However, this extensive use of information systems over the Internet has exposed us to many threats, including hacking and malicious software (malware), such as ransomware. To mitigate such threats, a firewall, which forms an essential part of any Internet and network security system, prevents intrusions from external networks to internal networks or devices on those networks; nevertheless, these networks are still considerably vulnerable to other attacks, such as Denial of Services (DoS) attacks that cannot be prevented by a firewall [1]. Furthermore, another disadvantage of firewalls is that they block only some of the hacking attacks that are made against a system or network. Considering this drawback and owing to the emergence of intelligent cyberattacks, the importance of attack detection and security on systems and networks has significantly increased in recent times. Thus, intrusion detection systems (IDS) [2], which have been studied for a considerable time, have been developed as next-generation security systems against hacking methods.

In general, network packets pass through the IDS after passing through the firewall, and the IDS generates an alarm if it detects malicious activities or determines anomalies in the incoming data [3]. Therefore, an IDS has a role similar to that of a firewall, but it also detects internal hacking and malicious codes that the firewall cannot detect and defend against. In addition, the IDS detects and responds to unauthorized activities against target systems that are not certified [4]. Thus, an IDS is an important tool for detecting security violations in real time. An IDS can be classified into two types: a host-based IDS (HIDS) and network IDS (NIDS) based on the position and purpose of the detection area according to datasource-based classification [2]. In order to detect malicious behaviors such as DoS attacks and port scans, an HIDS analyzes information collected from specific host systems, while an NIDS monitors network traffic [5]. Unlike the NIDS which detects attack vectors based on network traffic, the HIDS focuses on monitoring and analyzing the internal system, instead of the external network.

A HIDS can further be classified according to the type of model used for intrusion detection, namely misuse detection method and anomaly (or behavior) detection method. Both use information extracted from the analysis target to determine if an intrusion has occurred [6, 7, 8]. The misuse detection method, which is used in a signature-based (or knowledge-based) HIDS, is effective in detecting known attack vectors; nevertheless, it is vulnerable to attacks from unknown attack vectors. Therefore, there is a need for the anomaly detection methods [8, 9]. In particular, anomaly detection methods define and detect any anomalies that deviate from normal behavior patterns based on existing network usage scenarios, internal system calls, and so on.

In order to define normal behavior patterns, it is, therefore, necessary to extract normal behavior and anomaly patterns in HIDS. Then, machine learning algorithms based on iterative learning or data mining can be used to develop intrusion detection models using mathematical and statistical methods on these extracted patterns. Extensive research has been performed on applying data mining techniques on the new dataset to develop models for HIDS [8] as well as on network traffic data to develop models for NIDS [7]. Furthermore, the existing HIDS design suffers from the problem of a high false alarm

[*] Corresponding Author.

rate, thereby increasing the detection rate [8]. Since the approach suggested by [10], the works to reduce the false alarm rate based on system calls (which are interactions between programs and kernel) patterns in HIDS have resulted in a lot of researches [8]. However, it should be noted that the accuracy of the methods is not sufficiently high still.

Therefore, the objective of this study focuses on improving the accuracy of attack detection by applying the three different machine learning approaches to data preprocessed from system call sequence dataset released by [11]. Then the N-gram [12] method, which is one of data representation techniques, is used to preprocess the system call sequence dataset. In addition, after applying and comparing the results of various machine learning algorithms with the preprocessed data, we propose the most suitable machine learning algorithm model that improves the intrusion detection accuracy of HIDS. Furthermore, we verify the anomaly detection accuracy of the proposed HIDS models by performing simulations using the Arena simulation tool [13].

The remainder of the paper is organized as follows. Section II discusses the experimental datasets which are the system call sequence data released by [11] as well as previous studies that integrate machine learning algorithms and intrusion detection systems. The data preprocessing using N-gram method and the machine learning algorithms to be applied are explained in Section III. Section IV describes the experiments conducted in our study using the preprocessed datasets with various machine learning algorithms. Section V presents information on the verification tasks performed via simulations as well as the experimental results. Finally, Section VI provides our conclusions and directions for future research.

II. Datasets and Related works

A. Data Collection

Among the various experimental datasets used for research on HIDS, the publicly-available knowledge discovery and data mining (KDD) cup 99 datasets [14] has provided a systematic approach to forming intrusion detection system data. However, over time, this dataset has become outdated, and thus, many have criticized its use or are skeptical about applying it to current Internet environments [11]. Since computer and network systems have evolved, new attack vectors and vulnerabilities have emerged. Therefore, HIDS developed on the basis of existing datasets does not properly take into account the features of current attack vectors, so these existing datasets are not suitable for HIDS evaluation and validation [15].

Thus, alternative datasets reflecting current attack vectors have been proposed in [11]; an example of such a dataset is the research dataset provided by the Australian Defense Force Academy (ADFA) [11]. In many recent works, the ADFA dataset along with the latest attack vector features have been used for research on intrusion detection verification. In particular, the ADFA dataset was developed to evaluate a system call based HIDS as well as anomaly detection in signature-based HIDS.

The ADFA dataset is divided into the ADFA Linux dataset (ADFA-LD) and ADFA Windows dataset (ADFA-WD). The ADFA-LD reflects the features of current Linux-based operating systems, compared to many existing datasets used to evaluate the HIDS, and consists of thousands of system call traces collected from Linux local servers for the most recent attacks and vulnerabilities that occur in various applications. Considering this, the ADFA-LD is expected to become a new benchmark for evaluating and verifying HIDS.

Thus, in this study, using ADFA-LD, we extracted the attack patterns against the current HIDS and applied the machine learning and data mining techniques to the patterns to improve the accuracy of the attack and anomaly detection of the HIDS.

As previously mentioned, the ADFA-LD has thousands of normal traces collected from hosts on Linux servers, including abnormal trace files for six new types of cyberattacks, general user behavior and cyberattack path, and audit daemon setup, among others. In particular, during sampling periods for the ADFA-LD, a host captures system call traces that are generated by normally functioning legitimate programs and stores the corresponding data in a file. Among them, 8-20 abnormal call traces are stored as attack data files using call traces generated after a cyberattack is initiated against the test host. As listed in Table I, the ADFA-LD consists of three different data groups, each of which contains their own system call trace files. These data groups include training data master (TDM) and validation data master (VDM) groups, which represent normal data, whereas attack data master (ADM) group consists of call traces representing attack data. Furthermore, the ADM consists of six types of attack data: “Adduser”, “Hydra-FTP”, “Hydra-SSH”, “JavaMeterpreter”, “Meterpreter”, and “Web-Shell”.

B. Related Works

A number of system calls-based anomaly detection models have been designed to increase the accurate detection rate and to reduce the false alarm rate in HIDS. The paper [16] provides a survey of the host-based intrusion detection system with system calls, from the viewpoint of algorithms, techniques, datasets, application areas, and future research trends to inspire researchers about system-call-based HIDS in the big data and cloud environment. Also the paper [17] reviewed the research regarding intrusion detection techniques based on the HMM and provided challenges in this field. In this section we discuss existing works which integrate machine learning models and host-based anomaly detection systems.

Many studies have assessed the use of Hidden Markov Model (HMM), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Artificial Neural Network (ANN) algorithms, and so on for attack pattern recognition to improve the HIDS by reducing its false alarm rate and increasing the detection rate. As mentioned in the paper [18], the use of the Sequence Time-Delay Embedding (STIDE) algorithm in [8, 10] was problematic because the STIDE algorithm requires re-training for each particular process, and still has a high false alarm rate to any system call sequence data which do not appear in the training data. In [19], a scalable anomaly detection model was developed for servers in a cloud environment. The research proposed a nested-arc hidden semi-Markov model (NAHSMM) based on HHMM [20] which is a method for detecting anomalies in cloud servers. The anomaly detection algorithm is derived by integrating state summarization and NAHSMM and was evaluated with NGIDS-DS [21] dataset. The dataset is composed of labeled host logs and network packets. The model can work effectively with a smaller number of training samples and less processing time than recurrent neural networks (RNNs). In future work, with the NGIDS-DS data set and other data representations, we will consider comparing the study with various machine learning algorithms and deep learning models such as LSTM (Long Short-Term Memory) model and GRU (Gated Recurrent Unit) model.

In another study [15], considering the advancements in computer systems, as a preliminary work, researchers used the ADFA-LD dataset to evaluate a new host-based anomaly detection system (HADS) instead of the older datasets that were previously used. The common patterns and frequency of attacks were evaluated by the KNN-based HADS with the AFDA-LD dataset. Although acceptable detection results were obtained for some attacks by their proposed HADS, it still had a weakness in that it could not identify the behaviors of some attacks from normal behavior through the KNN algorithm.

In [22], researchers developed a frequency-based misuse detection method using ensemble classification. After preprocessing the raw ADFA-LD system call traces using the N-gram method, patterns were generated by extracting features; in addition, the number of patterns were balanced based on class through the synthetic minority over-sampling technique (SMOTE) [23]. The classification of temporal sequences with data-driven method do not need parameter estimation [12]. Therefore, in future work we will have to consider configuring a N-gram matrix that reflects well the data structure. Furthermore, their classification design was based on a majority voting ensemble technique of Naive Bayes, SVM [24], PART [25], Decision Tree, and Random Forest algorithms as well as Principal Component Analysis (PCA); their proposed misuse intrusion detection method showed good performance in terms of attack detection. Also host-based anomaly intrusion detection by Radial Basis Function neural network and Random Forest was conducted. The simulation study showed good performance in terms of detecting anomalies and normal activities.

The researchers in [26] used various machine learning classification algorithms to extract patterns from labelled new generation system call traces for modern exploits and attacks because they considered anomaly detection in ongoing processes using system call traces as a typical pattern recognition problem for machine learning. They evaluated the performance of the enhanced vector space representation technique for the ADFA-LD and their results showed good performance in distinguishing process behavior from exploits and attacks by using system calls.

TABLE. I. DATA GROUPS IN THE ADFA-LD DATASET

Data Groups	Type of Traces	Number of Traces
TDM	Normal	833
VDM	Normal	4372
ADM	Adduser	91
	Hydra-FTP	162
	Hydra-SSH	176
	JavaMeterpreter	124
	Meterpreter	75
	Web-Shell	118

III. PROPOSED HIDS DETECTION METHOD

Although the performance of an intrusion detection method based on misuse detection has been verified in a previous work [19, 24], to the best of our knowledge, there is a lack of machine learning approach on anomaly intrusion detection methods. Therefore, we study the classification of extracted system call sequence data into normal or malicious behaviour using machine learning algorithms. Section A describes data preprocessing using the N-gram method, and Section B presents the various machine learning algorithms used in the study. Fig. 1 shows a methodology of anomaly detection system using various machine learning approaches conducted in our study with the ADFA-LD dataset.

A. Data Preprocessing

The extracted system call trace data [11] consists of a series of numbers corresponding to system calls made on the Linux operating system. We apply machine learning algorithms to the system call trace data and then classify the process operation into normal behaviour or six specific attack types. First, we used the N-gram technique to extract attribute vectors from the system call trace dataset. The N-gram method involves cutting a sample text into a contiguous sequence of N characters or words. For an N-gram of size 1, i.e., N = 1, the N-gram is referred to as Uni-gram (1-gram), while for an N-gram of size 2, i.e., N=2, the N-gram is referred to as Bi-gram (2-gram). In this study for N-gram, a word units consist of system call numbers, and the number of system call sequence attributes is derived by creating an array of N words according to the given word order. By doing this step repetitively, the call attributes of the system call traces can be obtained. Fig. 2 shows an example of applying the N-gram technique on system call trace data.

In particular, N-gram data is expressed as a two-dimensional matrix; the columns of this matrix consist of the attribute values by matching the entire word belonging to each gram according N, while the rows represent instances that belong to each trace. The value corresponding to the row and column of the data represents the number of occurrences of N-gram in each trace as shown in Table II. As the value of N increases, the model becomes more complicated and requires considerably more storage space, thereby increasing processing time. Therefore, in this study, we limited N to 1 to 5. Furthermore, in order to extract those instances that occur most frequently in an entire trace, we extracted and used only instances that they were used more than once in the entire trace and had more than 30% (0.3) of all the instances of in the entire trace because the used instances were small.

B. Applied Machine Learning Algorithms

After pre-processing the data using the N-gram technique, we detected anomalies in the system call trace data using machine learning algorithms, based on which, classified and predicted normal data and six types of attack data from new system call trace files. In order to apply machine learning algorithms, we divided the dataset into training and test data. Then, the training data was used to model the algorithm, and the test data was used to validate the algorithm to ensure accuracy. In our study, the ADM dataset, which is the attack data to be detected, was used in a 7: 3 ratio for the training to test data.

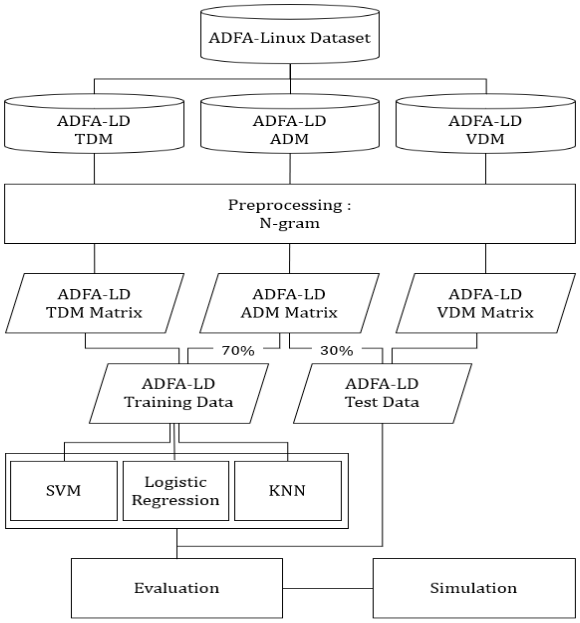


Fig. 1. Methodology Flow of Evaluation (Training/Testing) and Simulation Performed for the Proposed Anomaly Detection.

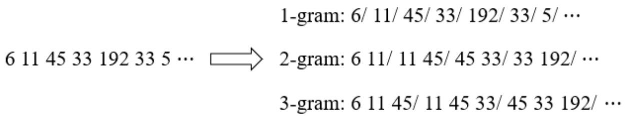

1-gram: 6/ 11/ 45/ 33/ 192/ 33/ 5/ ···
6 11 45 33 192 33 5 ···
2-gram: 6 11/ 11 45/ 45 33/ 33 192/ ···
3-gram: 6 11 45/ 11 45 33/ 45 33 192/ ···

Fig. 2. An Example of N-Gram units.

TABLE. II. THE NUMBER OF OCCURRENCES OF N-GRAM IN EACH TRACE

N-gram / System Call Trace	1-gram				···	5-gram			
	x_1	x_2	···	x_α	···	$x_{1'}$	$x_{2'}$	···	$x_{\alpha'}$
d_1	$N_{1,1}$	$N_{1,2}$	···	$N_{1,\alpha}$	···	$N_{1,1'}$	$N_{1,2'}$	···	$N_{1,\alpha'}$
d_2	$N_{2,1}$	$N_{1,2}$	···	$N_{1,\alpha}$	···	$N_{2,1'}$	$N_{1,2'}$	···	$N_{1,\alpha'}$
···	···	···	···	···	···	···	···	···	···
d_m	$N_{m,1}$	$N_{m,2}$	···	$N_{m,\alpha}$	···	$N_{m,1'}$	$N_{m,2'}$	···	$N_{m,\alpha'}$

The SVM algorithm proposed by Boser in 1992 is one of the most successful classification algorithms in the field of data mining. In particular, the SVM algorithm is a method of finding a hyperplane that maximizes the margins that are farthest from data among the hyperplanes that dichotomically divide data based on training data [27]. The SVM algorithm is based on a kernel function, which can solve large dimension issues, i.e., SVM does not suffer from problems associated with high dimensionality; in addition, the generalization ability of the SVM method can be enhanced by increasing margins during the training process [28]. Considering these features of the SVM algorithm, we considered it suitable to classify the data which was represented using a large-sized matrix preprocessed by the N-gram technique with the SVM algorithm for experiments.

Furthermore, the logistic regression algorithm uses the predictive model by representing the relationship between dependent and independent variables as a function. It is similar to the linear regression algorithm; however, unlike the linear regression algorithm, which uses continuous data, the dependent variables in the logistic regression algorithm are categorical data [29]. Thus, the logistic regression algorithm is considered useful for classifying categorical data and labeling the normal data and the six types of attack data considered in this study. Therefore, the logistic regression algorithm was used as another machine learning algorithm for experiments.

The third machine learning algorithm used in the study is the KNN algorithm which is a pattern recognition algorithm widely used for classification and regression. In this method, for classification, the input consists of the nearest K training data within a feature space and provides the class membership as output [30]. The KNN algorithm was used as a machine learning algorithm to detect anomalies in the system call trace data because it is effective in classifying data by labeling it as normal and attack data and specifying K.

In next section we will show the experimental results obtained from using the preprocessed datasets with various machine learning algorithms mentioned in this section.

IV. EXPERIMENTS WITH THREE DIFFERENT MACHINE LEARNING APPROACHES

In this study, we conducted experiments using the three machine learning algorithms: SVM, Logistic Regression, and KNN. The TDM and VDM data groups are the normal data, while the ADM group consists of the six types of attack data, which are listed along with their labels in Table III. After modeling the training data and the label of the training data using the machine learning algorithms, the label of the test data can be predicted with the test data.

For the SVM algorithm, we conducted experiments using the Linear function, Polynomial function, Sigmoid function, and Radial Basis Function (RBF) as kernel functions. The label prediction accuracy for the six types of attack and normal data as well as the time taken after applying each kernel of the SVM algorithm are listed in Table IV. By setting hyperparameter C, we can confirm the label prediction accuracy of the SVM algorithm based on C.

Furthermore, the prediction accuracy of labeling and the time taken after applying each kernel for the logistic regression algorithm are listed in Table V. In a manner similar to the SVM approach, by setting parameter C, we can confirm the label prediction accuracy of logistic regression based on C.

TABLE. III. ADFA-LD DATA LABELING

Data Groups	Type of Traces		Labeling
TDM	Normal		0
VDM	Normal		0
ADM	Attack	Adduser	1
		Hydra-FTP	2
		Hydra-SSH	3
		JavaMeterpreter	4
		Meterpreter	5
		Web-Shell	6

TABLE. IV. ACCURACY AND TIME TAKEN FOR LABELING DATA USING SVM

Kernel	Parameter C	Accuracy (%)	Time (sec)
Linear	0.1	79.5648	27.08
	1	79.0917	33.33
Polynomial	0.1	78.8079	24.79
	1	79.1864	26.15
Sigmoid	0.1	78.8079	22.48
	1	78.5241	22.78
RBF	100	80.9839	24.72
	1,000	82.6869	26.21
	10,000	80.5109	35.19

TABLE. V. ACCURACY AND TIME TAKEN FOR DATA LABELING USING LOGISTIC REGRESSION

Parameter C	**Accuracy (%)**	**Time (sec)**
0.1	76.4427	1.44
1	78.9025	1.49
10	78.1457	1.84
100	78.7133	2.98

TABLE. VI. ACCURACY AND TIME TAKEN FOR DATA LABELING USING KNN (K=7)

KNN type	**Accuracy (%)**	**Time (sec)**
BallTree	83.4437	3.35
KDTree	82.5922	2.44
Brute-force	84.7682	0.55

We also conducted experiments by applying the KNN algorithm for labeling data using three different KNN approaches, namely BallTree, KDTree, and Brute-force Search. The prediction accuracy and time taken for labeling each of these KNN algorithm types are listed in Table VI.

We evaluated the performance of each model using the AUROC curve (Area Under the ROC curve) in order to compare the predictions of each machine learning algorithm for applying to simulation described in Section V. Fig. 3 shows the AUROC curves for the SVM, Logistic Regression, and KNN machine learning algorithms. Table VII shows the summary of the highest prediction accuracies of the different models. In particular, the SVM with an RBF kernel and C= 1,000 has the highest AUC of 0.95 among the SVM kernels (Table IV); similarly, the Logistic Regression model with C=1 (Table V) and KNN Brute-force model (Table VI) have the corresponding highest accuracy based on the AUC model performance.

Fig. 3 depicts ROC Curve of Class N (where N = 0, 1, 2, 3, 4, 5, 6) expressed according to labels listed in Table III. The first figure in Fig. 3 representing the AUROC curve of the RBF model among the kernels of the SVM algorithm shows that the overall model performance is 95%. The second figure in Fig. 3 representing the AUROC curve when the parameter C of the Logistic Regression algorithm is set to 1 shows that the model performance is 96%. The last figure in Fig. 3 representing the KNN AUROC Curve using the Brute-force approach shows that the model performance is 96%. Overall, the AUC performance of the modeled machine learning algorithms is over 95% in all cases, which indicates that they are suitable for the machine-learning-algorithm-based HIDS model after appropriate data preprocessing and pattern extraction.

TABLE. VII. HIGHEST ACCURACY AND AUC PERFORMANCE OF THE APPLIED MACHINE LEARNING ALGORITHMS

KNN type	**Accuracy (%)**	**AUC**
SVM_RBF (C=1,000)	82.6869	0.95
Logistic Regression (C=1)	78.9025	0.96
KNN_Brute-force	84.7682	0.93

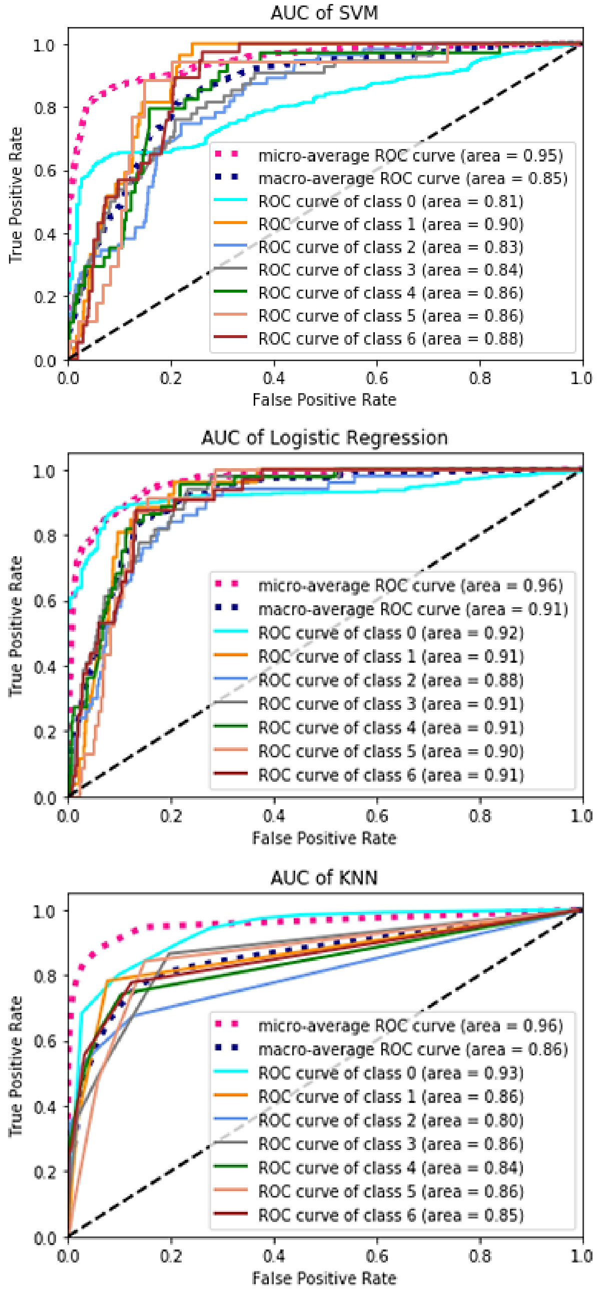


Fig. 3. AUROC of Applied Machine Learning Algorithms.

Our comparison experiments on the three machine learning algorithms - SVM, Logistic Regression, and KNN - indicate that there is significant difference in model performance when the Logistic Regression and KNN algorithms are employed, which can be attributed to their similarity. However, the Logistic Regression algorithm shows the best model performance with the performance values of most labeling (class) models reaching over 90%. Thus, the logistic regression algorithm is the most suitable one in terms of model performance; however, we did not confirm the reason for the model using Brute-force KNN algorithm having a high prediction accuracy.

V. Verification Simulation and Result Analysis

In practice, a HIDS can be installed and operated on different operating systems including on servers as well as clients. However, if simulation experiments for intrusion detection are conducted by directly installing the HIDS on personal computers, several problems need to be considered, including cost incurred because of performance, virus infections, or host malfunctions arising from IDS errors. Considering these possible issues, in this study, we verify the performance of the machine-learning-algorithm-based HIDS via simulations, which are quite similar to performing verification on actual systems. In particular, we constructed the HIDS model using the Arena simulation software, which is a proprietary software [13].

Arena simulation provides a simulation and animation environment designed to model discrete / continuous event system. The simulation system is easy to configure because the proprietary code is used to create models consisting of blocks and elements without the need for any additional code. In addition, these blocks are organized in a flow chart format; therefore, it facilitates easy progress monitoring [31]. The manner in which system calls are used in the HIDS is depicted in Fig. 4. At the user application level, the system call, read(), initiates a system call, such as sys_read(), at the kernel level through the HIDS. Furthermore, as indicated in Fig. 4, we can develop simulations by assigning the installation location of the HIDS to the system call interface that is placed from the user application level to the kernel level. The schematic design of the HIDS simulation model is shown in Fig. 5. The model reads the system call patterns of the normal data and attack data using the read-write module "System Call" after being initiated by the user module "user". Then, the sub-model "HIDS" classifies the data based on the accuracy results of SVM, Logistic Regression, and KNN, which are the three machine learning algorithms used in the experiments that are described in Section IV. After classifying in the sub-model, with the six types of attack data and normal data, they are classified as follows: 'Attack' which is classified as attack, 'Normal' that is classified as normal, and 'MissAttack' in which the six types of attack are misclassified as normal.

The schematic design of the sub-model "HIDS" is shown in Fig. 6. The data read through the "System Call" module is classified by the "Classify" module into the six types of attack data and normal data with the names ("IsAdduser," "IsHydraFTP," "IsHydraSSH," "IsJavaMeterpreter," "IsWebshell," "IsMeterpreter") expressed using the "n-way by condition" module. Then, we classify the accuracy results for each algorithm, which are obtained via our experiments, by applying the modules of "IsAdduser," "IsHydraFTP," "IsHydraSSH," "IsJavaMeterpreter," "IsWebshell," "IsMeterpreter," and "IsNormal." In the case of the six types of attack, if the result of the "decide" module is true, the "count" module corresponding to the respective attack increases the count by one. However, if the result is false, the "CountMissAttack" module adds one to the number of misclassified attacks. Furthermore, in the case of normal data, the "CountNormal" and "CountMissNormal" modules store the counts for true and false results similar to the previous case. The classification results obtained through simulation are shown in Fig. 7. In Fig. 7, the "CountMissNormal" and "CountMissAttack" values are important. First, the simulation result of the SVM algorithm shows that the number of misclassifications as indicated by "MissNormal", which is the count for the number of instances when normal data is misclassified, was zero. Therefore, it can be seen that normal data are classified considerably well in simulations compared with the experimental results of SVM_RBF that had an accuracy of only 82%. Furthermore, in the case of the KNN model simulation, the number of "MissNormal" instances is larger than that for the SVM simulation; however, the number of misclassifying the attack data is less than that of "MissAttack". Finally, from the logistic regression simulation results, it is clear that the model based on logistic regression performs worse than the other two machine learning algorithm models considered in this study. The most important feature in an IDS is that the false negative count "MissAttack" should be the least. Consequently, from the results shown in Fig. 7, we can confirm that the KNN-based model provides good results for detecting and classifying attack data, while the SVM-based model performs well in detecting normal data.

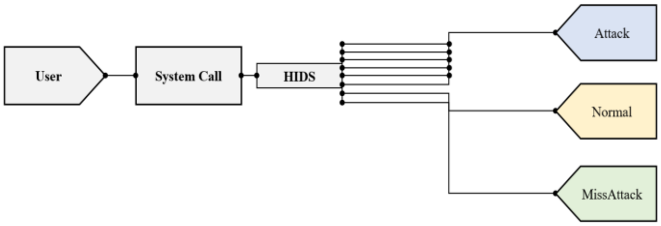


Fig. 4. System Call Process.

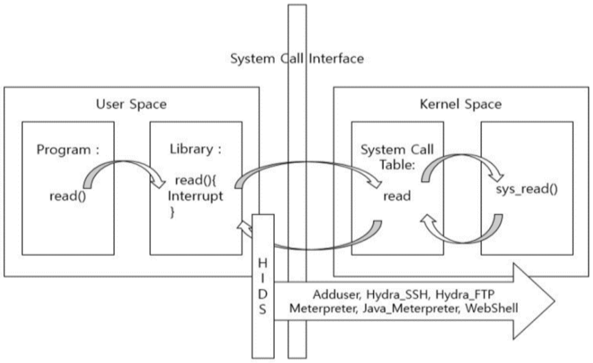


Fig. 5. HIDS Model based on Simulation.

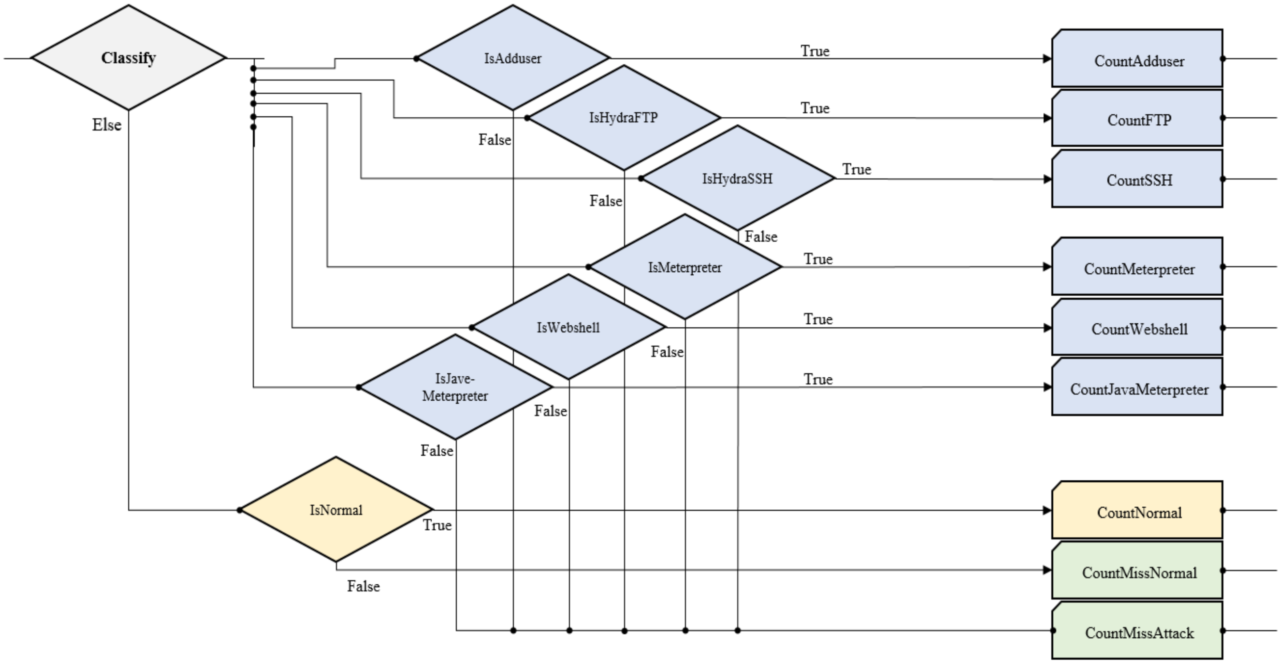
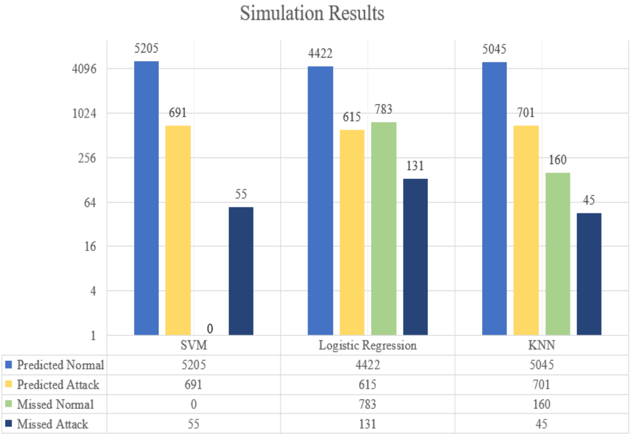


Fig. 6. Schematic Diagram of the HIDS Sub-Model based on Simulation.



	SVM	Logistic Regression	KNN
■ Predicted Normal	5205	4422	5045
■ Predicted Attack	691	615	701
■ Missed Normal	0	783	160
■ Missed Attack	55	131	45

Fig. 7. Simulation Results.

VI. Conclusions

In this study, we propose a method to increase intrusion detection accuracy by applying and comparing various machine learning algorithms that are suitable for intrusion detection models in order to overcome the disadvantages of an anomaly-based intrusion detection method. Using the ADFA-LD, which consists of various system call traces for attacks on the latest operating systems, we preprocessed the data using the N-gram technique and proposed a methodology to overcome the limitations of the STIDE algorithm.

For verification of our proposed methods, we simulated models using the Arena simulation tool to detect and classify anomalies in HIDS based on the machine learning algorithms considered in our study and verified the accuracy of these models. Based on our simulation results, we confirmed that changes in methodology, compared to previous studies, have made progress in improving the accuracy of anomaly detection in HIDS.

In conclusion, the methodology proposed in this study enables the detection of normal data and attack data as well as the classification of each attack data by extracting the patterns and features of anomalies using machine learning algorithms and applying them to anomaly detection in the HIDS, thereby significantly improving the HIDS, and thus, accurate detection rate.

In future work, we will consider to increase the accurate detection rate of anomaly-based intrusion detection systems using a variety of machine learning and deep learning models with a variety of dataset such as the NGIDS-DS dataset as well as ADFA-LD system call sequence dataset. In addition, we will conduct research on the adjustment of parameters and the development of improved machine learning algorithms to overcome the disadvantages of each machine learning algorithm.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT: Ministry of Science and ICT) (No. NRF-2019R1F1A1059036).

References

[1] A. D. Keromytis, V. Misra, and D. Rubenstein, “SOS: Secure Overlay Services," Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM '02), pp. 61-72, Pittsburgh, PA, USA. Aug. 19-23, 2002.

[2] D. Wagner and P. Soto, "Mimicry Attacks on Host-Based Intrusion Detection Systems," Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02), pp. 255-264,

Washington DC, 18-22 Nov. 2002, http://dx.doi.org/10.1145/586110.586145

[3] H. Cavusoglu, B. Mishra, and S. Raghunathan, "A Model for Evaluating IT Security Investments," Communications of the ACM, vol. 47, no. 7, pp. 87-92, July 2004.

[4] K. Richards, "Network based Intrusion Detection: A Review of Technologies," Computers & Security, vol. 18, no. 8, pp. 671-682, 1999.

[5] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A Survey of Intrusion Detection Techniques in Cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42-57. Jan. 2013.

[6] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An Intelligent Intrusion Detection System (IDS) for Anomaly and Misuse Detection in Computer Networks," Expert Systems with Applications, vol. 29, no. 4, pp. 713-722, Nov. 2005.

[7] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, and E. Vazquez, "Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges," Computers & Security, vol. 28, no. 1-2, pp. 18-28, 2009.

[8] G. Creech and J. Hu, "A Semantic Approach to Host-based Intrusion Detection Systems using Contiguous and Discontiguous System Call Patterns," IEEE Transactions on Computers, vol. 63, no. 4, pp. 807-819, Apr. 2014.

[9] A. Torkaman, G. Javadzadeh, and M. Bahrololum, "A Hybrid Intelligent HIDS Model using Two-layer Genetic Algorithm and Neural Network," 5th Conference on Information and Knowledge Technology (IKT), pp. 92-96, 28-30 May 2013.

[10] S. Forrest, S. A. Hofmeyr, A. SoMayaji, and T. A. Longstaff, "A Sense of Self for Unix Processes," Proceedings of IEEE Symposium on Security and Privacy, pp. 120-128, May 1996.

[11] G. Creech and J. Hu, "Generation of a New IDS Test Dataset: Time to Retire the KDD Collection," Wireless Communications and Networking Conference (WCNC 2013), Shanghai, 7-10 April 2013 http://dx.doi.org/10.1109/WCNC.2013.6555301

[12] X. Zhang, Y, Wang, M. Gou, M. Sznaier, and O. Camps, "Efficient Temporal Sequence Comparison and Classification using Gram Matrix Embeddings on a Riemannian Manifold," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4498-4507, June 2016. https://doi.org/10.1109/CVPR.2016.487

[13] Arena Simulation Software, http://www.arenasimulation.com/

[14] KDD Cup 1999. Available on: http://kdd.ics.uci.edu/databases /kddcup99 /kddcup99.html, Feb. 2019.

[15] M. Xie and J. Hu, "Evaluating Host-Based Anomaly Detection Systems: A Preliminary Analysis of ADFA-LD," 6th IEEE International Congress on Image and Signal Processing (CISP '03), pp. 1711-1716, Dec. 2013.

[16] M Liu, Z Xue, X Xu, C Zhong, J Chen, "Host-based Intrusion Detection System with System Calls: Review and Future Trends," ACM Computing Surveys (CSUR), vol. 51, no. 5, pp. 1-36, Nov. 2018. https://doi.org/10.1145/3214304

[17] A. Ahmadian Ramaki, A. Rasoolzadegan and A. Javan Jafari, "A Systematic Review on Intrusion Detection based on the Hidden Markov Model," Statistical Analysis and Data Mining - Wiley Online Library: The ASA Data Science Journal, vol. 11, no. 3, pp. 111-134, Apr. 2018. https://doi.org/10.1002/sam.11377

[18] G. Creech, "Developing a High-accuracy Cross Platform Host-Based Intrusion Detection System Capable of Reliably Detecting Zero-day Attacks," Ph.D. Dissertation, University of New South Wales, Canberra, Australia, 2014.

[19] W. Haider, J. Hu, Y. Xie, X. Yu, and Q. Wu, "Detecting Anomalous Behavior in Cloud Servers by Nested Arc Hidden SEMI-Markov Model with State Summarization," IEEE Transactions on Big Data, vol. 5, no. 3, pp. 305-316, sept. 2019. https://doi.org/10.1109/TBDATA.2017.2736555

[20] S. Fine, Y. Singer, and N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," Machine learning, vol. 32, no. 1, pp. 41–62, 1998. https://doi.org/10.1023/A:1007469218079

[21] W. Haider, J. Hu, J. Slay, B. Turnbull, and Y. Xie, "Generating Realistic Intrusion Detection System Dataset based on Fuzzy Qualitative Modeling," Journal of Network and Computer Applications, vol. 87, pp. 185–192, June 2017. https://doi.org/10.1016/j.jnca.2017.03.018

[22] E. Aghaei, "Machine Learning for Host-based Misuse and Anomaly Detection in UNIX Environment," M.S. Thesis, Computer Science in University of Toledo, May 2017.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321-357, June 2002.

[24] W. Haider, J. Hu, and M. Xie, "Towards Reliable Data Feature Retrieval and Decision Engine in Host-based Anomaly Detection Systems," IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), pp. 513-517, June 2015. https://doi.org/10.1109/ICIEA.2015.7334166

[25] H. Berger, D. Merkl, and M. Dittenbach, "Exploiting Partial Decision Trees for Feature Subset Selection in e-Mail Categorization," Proceedings of the ACM Symposium on Applied Computing, Dijon, France, 2006.

[26] B. Borisaniya and D. Patel, "Evaluation of Modified Vector Space Representation Using ADFA-LD and ADFA-WD Datasets," Journal of Information Security, vol. 6, no. 3, pp. 250-264, 2015.

[27] Y. B. Bhavsar and K. C. Waghmare, "Intrusion Detection System Using Data Mining Technique: Support Vector Machine," International Journal of Emerging Technology and Advanced Engineering, vol. 3, no. 3, pp. 581-586, March 2013.

[28] W. S. Noble, "What is a Support Vector Machine?," Nature Biotechnology, vol. 24, no. 12, pp. 1565-1567, Dec. 2006.

[29] C. J. Peng, K. L. Lee, and G. M. Ingersoll, "An Introduction to Logistic Regression Analysis and Reporting," Journal of Educational Research, vol. 96, no. 1, pp. 3-14, 2002. https://doi.org/10.1080/00220670209598786

[30] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning Intrusion Detection: Supervised or Unsupervised?," International Conference on Image Analysis and Processing, Lecture Notes in Computer Science, vol. 3617, Springer, Berlin, Heidelberg, 2005.

[31] A. Vieira, L. Dias, G. Pereira, and J. Oliveira, "Comparison of SIMIO and ARENA Simulation Tools," 12th Annual Industrial Simulation Conference (ISC2014), University of Skövde, Skövde, Sweden, pp. 5-13, June 2014.