Evaluation of Genetic Algorithm and Decision Tree Optimizations for Anomaly Detection IDS

Ali Gharaee[1] and Hossein Gharaee Garakani[2*†]

[1]Psychological Sciences, Azad University, South Tehran Branch,
Damavand, Tehran, 17117-34353, Tehran, Iran.
[2]Network Security, ICT Research Institute(ITRC), North Karegar,
Tehran, 1439845739, Tehran, Iran.

*Corresponding author(s). E-mail(s): gharaee@itrc.ac.ir;
Contributing authors: al.gharaee@gmail.com;
†These authors contributed equally to this work.

Abstract

In this study, we evaluate and compare two feature optimization methods for enhancing Intrusion Detection Systems (IDS): Genetic Algorithm (GA)-based and Decision Tree (DT)-based approaches. Both methods aim to select optimal feature subsets to improve classification performance and reduce computational cost. The GA-based approach integrates a novel fitness function with Least Squares Support Vector Machine (LSSVM) to simultaneously maximize the True Positive Rate (TPR), minimize the False Positive Rate (FPR), and optimize features. The DT-based method leverages a hybrid filter-wrapper strategy using C5.0 decision trees and LSSVM, employing gain ratio and pruning for initial selection, followed by predictor importance ranking for refined optimization. Experimental evaluations using the KDD CUP 99 and UNSW-NB15 datasets demonstrate that the DT-based feature selection consistently outperforms the GA-based method across most attack categories in terms of accuracy, TPR, FPR, and Area Under the Receiver Operating Characteristic (ROC) Curve (AUC). The results show that DT has better performance than GA with respect to feature optimization.

Keywords: Intrusion Detection System, Anomaly, Genetic Algorithm, Decision Tree, Support Vector Machine

1 Introduction

As the internet becomes increasingly embedded in every aspect of modern life—from banking and healthcare to smart homes and national infrastructure—the need for strong cybersecurity has never been more urgent. Although the conventional methods like firewalls, encryption and antivirus software packages adopted by organizations play a significant role in securing network infrastructure, still, these methods provide the first level of defense and cannot completely protect the networks and systems from progressive attacks and malware[1]. As a result, organizations use intrusion detection systems (IDSs), which Denning proposed in 1987, as an additional security technique for securing their networks [2].

1.1 Intrusion Detection Systems (IDS)

IDSs play a critical role in safeguarding computer networks by identifying unauthorized access, attacks, and abnormal activities in real time.

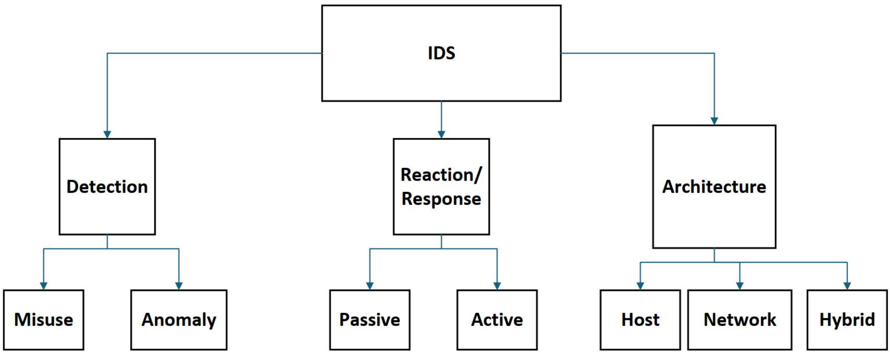

Fig. 1 IDSs Classification

As depicted in Fig. 1, based on the detection method, IDSs can be classified in-to two categories: Misuse-based or Signature-based, and Anomaly-based. Before diving into the details of anomaly-based intrusion detection, it is helpful to briefly explain the concept of anomaly detection itself. In general, anomaly detection is an important data analysis task that detects anomalous or abnormal data from a given dataset. It is an interesting area of data mining research as it involves discovering enthralling and rare patterns in data. It has been widely studied in statistics and machine learning, and also synonymously termed as outlier detection, novelty detection, deviation detection and exception mining. Although an anomaly is defined by researchers in various ways based on its application do-main, one widely accepted definition is that of Hawkins: 'An anomaly is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism' [3]. Here, anomaly detection involves profiling user behavior. In this approach, a model of normal user activity is established, and any significant deviation from this model is considered anomalous [2]. Anomaly-based IDSs detect both network and computer intrusions by monitoring the system. Instead of relying on known patterns or signatures of attacks (which signature-based IDSs use), anomaly-based IDSs analyze system or network activity using a set of heuristics or rules. These rules are based on what is considered ”normal” behavior. The system monitors activity and flags anything that deviates significantly from the norm as anomalous, which could indicate a potential security threat or attack — even if it is previously unknown[4]. While signature-based or misuse-based IDSs just can only detect attacks that happened earlier and whose signatures exist, anomaly-based ones can detect zero-day attacks[5]. However, establishing the rule set in anomaly-based approaches can be a complex and challenging process, and such a type of IDS might generate a huge number of false positives[1, 3].

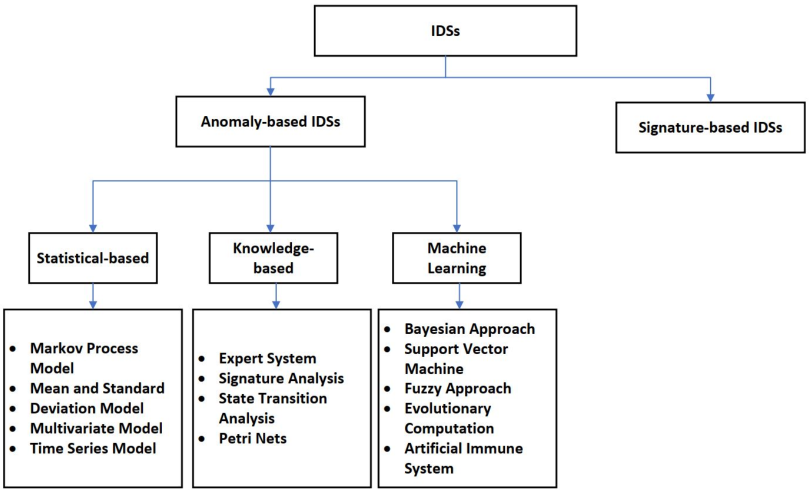

Fig. 2 Anomaly-based IDSs

This research focuses on anomaly-based intrusion detection approaches, which are generally categorized into three types: knowledge-based, statistical, and ma-chine learning methods[2]. Among these, as illustrated in Fig. 2, the current re-search specifically concentrates on machine learning-based anomaly detection systems. Many research ideas have been proposed pertaining to the IDS using machine learning (ML) techniques, deep learning (DL) techniques, and swarm and evolutionary algorithms (SWEVO). Machine learning techniques are particularly well-suited for anomaly-based intrusion detection because they can learn complex patterns of normal and abnormal behavior from large datasets. Unlike traditional signature-based methods, ML models can generalize from historical data and detect previously unseen (zero-day) attacks, making them effective in dynamic and evolving threat environments. This adaptability is crucial in modern cybersecurity contexts, where attack methods are constantly changing. These methods have been tested on the datasets such as DARPA, KDD CUP 99, and NSL-KDD using network features to classify attack types[3].

1.2 Feature Selection via Genetic Algorithm and Decision Tree

The high dimensionality of network data is indeed a significant challenge for ML-based anomaly detection IDSs, and feature selection is considered a crucial step to address this challenge. It helps to simplify the data, improve the efficiency and accuracy of the IDS, and reduce the risk of the overfitting. In the current paper, two widely used feature selection methods—Genetic Algorithm and Decision Tree—are evaluated and compared in the context of intrusion detection systems[5]. Whether using GA or DT, feature selection methods generally fall into two categories: filter-based and wrapper-based. Wrapper-based feature selection is a more involved approach to choosing the best features for a machine learning model[6]. Instead of just looking at the characteristics of the data itself (like in filter methods), wrapper methods actually use the machine learning model that we are trying to build to evaluate different sets of features [5]. It can be thought of like trying to pick the best ingredients for a cake by actually baking several small cakes with different combinations of ingredients and then tasting which one is the best. It is called a "wrapper" method because the feature selection process is "wrapped around" the machine learning model. The process of wrapper-based selection generally begins by selecting a specific machine learning model intended for the final task, which are, in the current research, SVM and LSSVM for GA and DT, respectively. Next, a search strategy is defined to explore different combinations of features. This search may involve techniques such as for-ward selection, which starts with no features and adds the most relevant ones; backward elimination, which begins with all features and removes the least useful; or Recursive Feature Elimination (RFE), where the model is built multiple times, features are ranked by importance, and the lowest-ranked ones are iteratively removed. More advanced strategies, such as genetic algorithms, can also be used to explore the space of feature combinations in a more randomized and evolutionary way. For each subset generated by the chosen strategy, the model is trained using only those features, and its performance is evaluated on a separate validation dataset using metrics such as accuracy, precision, recall, or F1-score. Finally, the feature subset that results in the best model performance is selected for use in the final model. On the other hand, while the model acts as the evaluator for the feature subsets in the wrapper-based approach, the filter-based method's estimation of classification performance is done with indirect assessment and does not rely on classifier performance [5]. Filter methods evaluate the relevance of features based on their intrinsic properties and their relationship to the target variable—such as distinguishing between anomalies and normal activity in an IDS—independently of any specific machine learning algorithm. Used as an optimizing feature selection, Genetic Algorithm is a powerful technique inspired by the evolutionary principles of natural genetics. It generates a population of chromosomes, each representing a potential solution to the problem [5]. The vital component of it is the fitness function, which evaluates the quality of the chromosomes. A well-designed fitness function helps the algorithm identify subsets of features that are closer to the desired outcome. Previous IDS models based on GAs have typically used two evaluation factors: classification accuracy and the number of selected features. Nevertheless, a major limitation of earlier models was that they evaluated feature chromosomes based solely on accuracy or true positive rate, overlooking one of the vital challenges in IDS: the high false alarm rate. This issue was often not addressed in prior approaches [7]. Considering feature selection, another popular and widely used ML method is Decision Tree. Decision trees follow a ”divide and conquer” approach, which means they repeatedly break down a large problem into smaller sub-problems. At each step, the data is split into groups based on the value of a specific feature (like ”protocol type” or ”packet size”). The choice of which feature to split on is made using a specific criterion (like information gain or gain ratio) that measures how well a feature separates the data into useful categories [8]. At the final stage of the proposed IDSs, SVM is used for classification. This model not only evaluates the effectiveness of the feature selection process but also acts as the final component in the IDS, which classifies network traffic as normal or an attack. Support Vector Machine (SVM) is an outstanding and well-known machine learning classification method for this final stage that can be used to design an IDS. The advantages of the SVM are its mathematical tractability, clear geometric interpretation, low-cost function, and fast simulation [5, 8]. SVM can be considered as an empirical risk minimization method and consequently, defines the theoretical bounds on its performance [9]. On the other hand, Least Squares SVM (LSSVM) is a reformulation version of SVM that solves the SVM convex quadratic programming problem faster and easier, with higher performance [8].

2 Related Works

In the current research, it is aimed to concentrate on the two proposed methods, which are: 1. Genetic Algorithm-based Feature Optimization 2. Decision Tree-based Feature Optimization It is noteworthy that both IDSs approaches share common stages, which are depicted in the following figures. Additionally, the proposed methods have been evaluated on the same datasets, which will be introduced in the following sections.

2.1 Genetic Algorithm Feature Optimization

In [5], Hossein Gharaee et al. proposed a Genetic Algorithm (GA)-based feature selection method for Intrusion Detection Systems (IDS), aimed at addressing limitations in earlier models by identifying optimal feature subsets. A key contribution of their work is a custom-designed fitness function that incorporates True Positive Rate (TPR), False Positive Rate (FPR), and the number of selected features (NumF). This multi-objective formulation seeks to maximize detection performance (high TPR), minimize false alarms (low FPR), and reduce feature dimensionality to improve computational efficiency. Their method integrates GA with a Support Vector Machine (SVM) to evaluate candidate feature subsets, leveraging SVM’s scalability and effectiveness. Additionally, they employed Least Squares SVM (LSSVM) with a Radial Basis Function (RBF) kernel, which has been shown to outperform standard SVM in terms of accuracy and convergence speed [10]. Building upon this foundation, our study extends the fitness function and applies it in conjunction with LSSVM to evaluate and compare GA-based and Decision Tree-based feature optimization strategies for IDS. It is important to clarify that the fitness function is not part of the Support Vector Machine (SVM) or classification process itself. Rather, it is a core component of the Genetic Algorithm and is used exclusively during the optimization phase to evaluate the quality of different feature subsets. The classification model (LSSVM) is only used to compute performance metrics—such as TPR and FPR—that serve as inputs to the fitness function. Therefore, the fitness function guides the evolutionary search in the GA, while the classifier remains a separate down-stream component.

2.2 Decision Tree Feature Optimization

Decision tree algorithms generally prefer simpler structures with fewer nodes and shallower depths, as minimizing complexity can enhance accuracy[11]. Methods such as pruning are often applied to streamline the tree structure. In [8], a net-work intrusion detection system (NIDS) was proposed using the C5.0 decision tree for feature selection and LSSVM for classification, resulting in the DTLSSVM model. The feature selection process is built upon the decision tree's inherent simplicity [12] and employs a structured framework that integrates both filter-based and wrapper-based approaches [13]. Additionally, the model incorporates both true negative and false positive rates in the normal class detection. The proposed method is evaluated using KDD CUP 99 and UNSW-NB15, with the latter also having been utilized in prior research. The DTLSSVM model consists of three components: the first two stages focus on hybrid feature selection, while the third stage handles the detection phase. In the initial filter-based stage, features that do not effectively differentiate between training instances are removed using the gain ratio. To further reduce the feature set without sacrificing detection accuracy, error-based pruning based on the pessimistic upper bound error criterion is then applied. Since redundant features may still persist after pruning, a wrapper-based method is employed in the second stage. This component uses the predictor importance metric derived from the C5.0 algorithm to rank and assess feature subsets using the LS-SVM classifier. Notably, this study is the first to apply the predictor importance criterion for feature selection in IDS contexts. The final optimized feature subset is then used in the third component to train the LS-SVM-based NIDS.

3 Proposed Optimization Algorithms

In this section, we want to compare the two methods of IDS, one based on Genetic Algorithm for feature optimization and the other based on Decision Tree for feature optimization. The shared optimization process for both GA-based and DT-based anomaly detection systems is illustrated in the figure below.

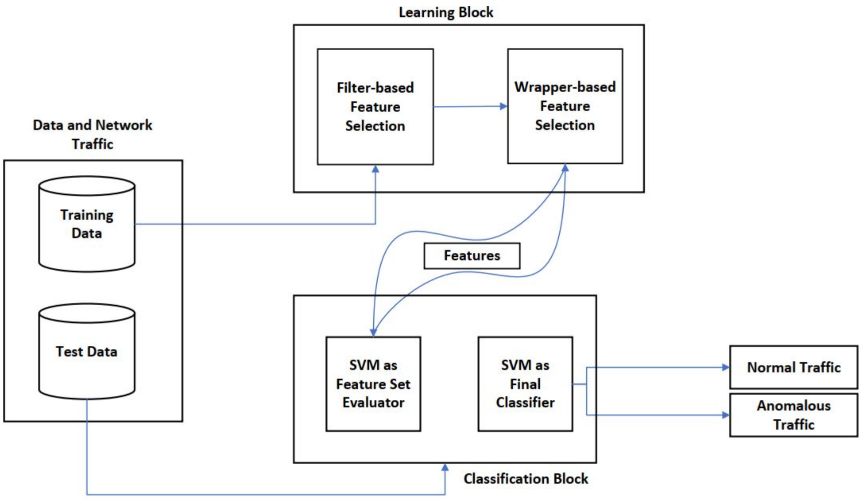

Fig. 3 Generic Feature Selection Optimization Method

In Fig.4, the specific parts of genetic algorithm feature optimization are illustrated.

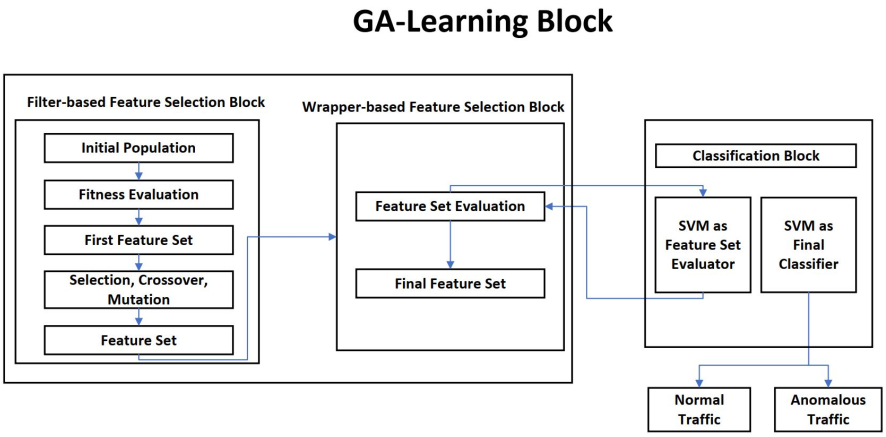

Fig. 4 GA-Feature Selection Optimization Method

On the other hand, the specific part that belongs to decision tree feature optimization can be in Fig. 5.

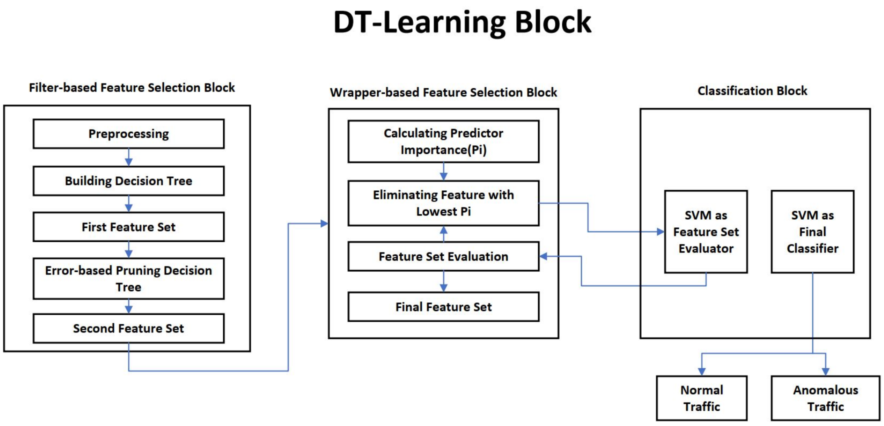
DT-Learning Block
Filter-based Feature Selection Block
Preprocessing
Building Decision Tree
First Feature Set
Error-based Pruning Decision Tree
Second Feature Set
Wrapper-based Feature Selection Block
Calculating Predictor Importance(Pi)
Eliminating Feature with Lowest Pi
Feature Set Evaluation
Final Feature Set
Classification Block
SVM as Feature Set Evaluator
SVM as Final Classifier
Normal Traffic
Anomalous Traffic

Fig. 5 DT-Feature Selection Optimization Method

4 Experiments

Here, first, the utilized datasets, their descriptions, and characteristics are presented, and then the performance metrics are defined. Finally, the experiments are conducted and the corresponding results are reported.

4.1 Utilized Datasets

The proposed IDSs are evaluated using the widely recognized datasets, namely KDD CUP 99 and UNSW-NB15. The KDD CUP 99 dataset was created by MIT Lincoln Laboratory as part of the DARPA 1998 intrusion detection evaluation program [8]. It was designed to support the study and assessment of intrusion detection research. Over a nine-week period, raw TCP traffic data from a local area network (LAN) was collected by the Lincoln Laboratory. Approximately five million connection records were included, with each record representing a TCP/IP connection described by 41 features. These features were classified into four groups: substantial features, content features, time-based traffic features, and host-based traffic features [14]. The attacks present in the dataset were also categorized into five groups, namely DoS, Probe, R2L (Remote to Local), and U2R (User to Root). Each category was composed of a specific sort of attack. It was stated by Moustafa and Slay that the KDD CUP 99 and NSL-KDD datasets were incapable of supporting the detection of new attacks by IDSs. As a result, a new dataset was provided based on the UNSW-NB15 network [15, 16]. In this dataset, a total of 2,540,044 records were included, each consisting of 49 features. Nine types of attacks—namely reconnaissance, shellcode, exploit, fuzzers, worm, DoS, backdoor, analysis, and generic—along with normal traffic, were represented [15]. The 49 features were grouped into six categories: flow features, basic features, content features, time features, additional generated features, and labeled features. The additional generated feature group was further subdivided into general-purpose features, which were from features 36 to 40, and connection features, which were from features 41 to 47 [17].

4.2 Performance Metrics

For our proposed IDS approaches, we have three metrics to assess, which are True Positive Rate (TPR), also termed as Detection Rate, False Positive Rate (FPR), and finally Accuracy.

TPR is obtained via be below formula:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1}$$

FPR is obtained via the below formula:

$$\text{FPR} = \frac{\text{FP}}{\text{TN} + \text{FP}} \tag{2}$$

Accuracy is obtained via the below formula:

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TN} + \text{TP} + \text{FN} + \text{FP}} \tag{3}$$

In the formulas above, TP (True Positive) refers to the number of correctly detected attacks, while FP (False Positive) denotes the number of normal samples incorrectly identified as attacks. TN (True Negative) indicates the number of correctly detected normal samples, and FN (False Negative) represents the number of attacks that were incorrectly classified as normal.

4.3 Fitness Function

In the GA-based feature selection approach, a custom fitness function was employed to guide the optimization process. The goal was to select feature subsets that simultaneously maximize detection performance and reduce feature dimensionality. The fitness function is defined as:

$$\text{Fitness}(S) = \alpha \cdot \text{TPR} - \beta \cdot \text{FPR} - \gamma \cdot \text{NumF}(S) - \theta \cdot \text{FNR} \tag{4}$$

Where:

- TPR: True Positive Rate
- FPR: False Positive Rate
- NumF(S): Number of selected features in subset S
- α, β, γ, θ: Weight coefficients that balance detection rate, false alarm rate, number of selected features, and false negative rate respectively

This multi-objective fitness formulation ensures that subsets leading to higher detection, fewer false alarms, and lower feature count are favored during the ge-netic search process. The evaluation of each chromosome (feature subset) was carried out using an LS-SVM classifier with RBF kernel.

4.4 Results

According to the illustrated tables and statistics, the DT-based optimization method shows a better performance compare to the GA-based optimization. It is probable that if the fitness function changes, the second GA-based optimization performance increases as well.

Table 1 KDD CUP 99 Compared Evaluation Values

Attack Type	**Model**	**Accuracy (%)**	**TPR (%)**	**FPR (%)**	**AUC**
DoS	DT-LSSVM	99.88	99.8	0.099	0.9985
	GA-LSSVM	99.86	99.8	0.068	0.9997
	[35]	99.81	99.8	0.23	0.9950
Probe	DT-LSSVM	98.60	97.24	0.039	0.9860
	GA-LSSVM	95.16	93.80	3.86	0.9516
	[35]	97.83	96.08	0.44	0.9782
R2L	DT-LSSVM	98.42	92.56	0.074	0.9595
	GA-LSSVM	95.75	84.35	2.59	0.9087
	[35]	95.47	79.07	2.15	0.8846
U2R	DT-LSSVM	99.891	75.00	0.000	0.8750
	GA-LSSVM	99.891	75.00	0.000	0.8750
	[35]	99.891	75.00	0.000	0.8750
Normal	DT-LSSVM	94.35	95.27	6.57	0.9435
	GA-LSSVM	93.19	98.71	12.37	0.9317
	[35]	89.529	86.318	7.235	0.8967

Table 2 UNSW-NB15 Compared Evaluation Values

Attack Type	Model	Accuracy (%)	TPR (%)	FPR (%)	AUC
ShellCode	DT-LSSVM	99.88	99.8	0.099	0.9985
	GA-LSSVM	99.30	100	12.50	0.9375
	[38]	94.41	100	100	0.9960
Reconnaissance	DT-LSSVM	95.37	93.41	2.03	0.9900
	GA-LSSVM	89.54	88.39	8.93	0.9346
	[38]	93.54	90.50	2.45	0.9770
Generic	DT-LSSVM	96.12	98.96	7.42	0.9911
	GA-LSSVM	85.51	99.26	30.17	0.8402
	[38]	94.01	96.77	9.45	0.9960
Fuzzer	DT-LSSVM	98.27	98.60	2.04	0.9900
	GA-LSSVM	96.76	97.38	3.84	0.9803
	[38]	96.19	96.20	3.80	0.9610
Exploit	DT-LSSVM	87.47	87.72	12.38	0.9590
	GA-LSSVM	79.19	67.31	6.23	0.7820
	[38]	83.52	85.09	18.40	0.9500
Analysis	DT-LSSVM	97.69	99.92	27.00	0.8660
	GA-LSSVM	–	–	–	–
	[38]	93.59	99.57	82.29	0.5000
Backdoor	DT-LSSVM	94.67	98.94	59.43	0.7000
	GA-LSSVM	–	–	–	–
	[38]	93.59	99.59	82.29	0.8090
DoS	DT-LSSVM	89.65	90.23	10.86	0.9470
	GA-LSSVM	83.45	92.89	24.91	0.9451
	[38]	90.10	92.17	11.73	0.9110
Worm	DT-LSSVM	99.33	100.00	70.00	0.6285
	GA-LSSVM	–	–	–	0.9960
	[38]	99.05	100.00	100.00	0.5000

5 Conclusion and Future Work

In this research, we presented and compared two optimization methods—Genetic Algorithm (GA) and Decision Tree (DT)—for anomaly-based intrusion detection systems (IDS), using LS-SVM as the classifier. Firstly, we revised the fitness function used in the GA-based feature selection process and proposed a new formulation, which led to improved performance compared to previous studies. For the Decision Tree-based method, we implemented a hybrid filter-wrapper strategy using the C5.0 algorithm and applied pruning techniques to remove less relevant features. This also contributed to enhanced classification accuracy. Our experimental evaluations on the KDD CUP 99 and UNSW-NB15 datasets showed that the DT-based approach consistently achieved higher accuracy and lower false positive rates across multiple attack categories. A comparative analysis of the overall optimization performance confirmed that the Decision Tree-based method outperformed the GA-based approach, as reflected in the results presented in the evaluation tables. In future work, we plan to further explore and refine the fitness function by introducing additional evaluation variables. We also intend to assess our models using more recent and diverse datasets, such as CIC-IDS2017, to better reflect modern network traffic and attack scenarios. Finally, we aim to investigate ensemble-based methods, such as Random Forest, to enhance the generalization and robustness of Decision Tree-based optimization.

References

[1] Srinivas, J., Das, A.K., Kumar, N.: Government regulations in cyber security: Framework, standards and recommendations. Future Generation Computer Systems **92**, 178–188 (2019)

[2] Kocher, G., Kumar, G.: Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges. Soft Computing **25**, 9731–9763 (2021)

[3] Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. Journal of Network and Computer Applications **60**, 19–31 (2016)

[4] Farzaneh, B., Montazeri, M.A., Jamali, S.: An anomaly-based ids for detecting attacks in rpl-based internet of things. In: 5th International Conference on Web Research (ICWR), Tehran, Iran, pp. 61–66 (2019). IEEE

[5] Gharaee, H., Fekri, M., Hosseinvand, H.: Intrusion detection system using svm as classifier and ga for optimizing feature vectors. ITRC Journal **10**(1), 26 (2018)

[6] Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., Dai, K.: An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Systems with Applications **39**(1), 424–430 (2012)

[7] Abadeh, M.S., Mohamadi, H., Habibi, J.: Design and analysis of genetic fuzzy systems for intrusion detection in computer networks. Expert Systems with Applications **38**(6), 7067–7075 (2011)

[8] Serkani, E., Gharaee Garakani, H., Mohammadzadeh, N.: Anomaly detection using svm as classifier and decision tree for optimizing feature vectors. The ISC International Journal of Information Security **11**(2), 159–171 (2019)

[9] Zhong, L.L., Zhang, Y.M., Zhang, Y.B.: Network intrusion detection method by least squares support vector machine classifier. In: 3rd International Conference on Computer Science and Information Technology, Chengdu, China, pp. 295–297 (2010). IEEE

[10] Zhao, M., Fu, C., Ji, L., Tang, K., Zhou, M.: Feature selection and parameter optimization for support vector machines: a new approach based on ga with feature chromosomes. Expert Systems with Applications **38**(5), 5197–5204 (2011)

[11] Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks/Cole, Monterey, CA (1984)

[12] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

[13] Serkani, E., Gharaee Garakani, H., Mohammadzadeh, N., Vaezpour, E.: Hybrid anomaly detection using decision tree and support vector machine. International Journal of Electrical, Electronic and Communication Sciences **6** (2018)

[14] Lincoln, M.: KDD Cup 99. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Last accessed 2025/05/11

[15] Moustafa, N., Slay, J.: The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In: 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), pp. 25–31 (2015). IEEE

[16] Moustafa, N.: The UNSW-NB15 data set. https://www.unsw.adfa.edu.au/unsw-canberracyber/cybersecurity/ADFA-NB15-Datasets/. Last accessed 2025/05/11

[17] Janarthanan, T., Zargari, S.: Feature selection in unsw-nb15 and kddcup'99 datasets. In: 26th International Symposium on Industrial Electronics (ISIE), pp. 1881–1886 (2017). IEEE