Comparative Analysis of Different Attack on Web Log Data using Machine Learning Technique

Diwakar Prasad Nuniya[1], Nisha[2]

[1]Research Scholar, [2]Assistant Professor
Department of Computer Science,
PKG Group of Institutions, Panipat

***Abstract*: It is proposed in this research that a multi-stage filter be designed based on the analysis and distribution of various types of network assaults in web log datasets. An extended GOA algorithm with a decision tree algorithm is used in the first stage of the filter to detect frequent attacks, and an enhanced GOA algorithm with a genetic algorithm is used in the second stage to detect moderate attacks. An upgraded GOA algorithm using Nave Bayes as a base learner has been utilized in the final stage of the filter to detect the rare attacks. Many features are included in benchmark datasets used to test and evaluate intrusion detection systems. These massive datasets, on the other hand, necessitate more computational power and time. Intrusion detection relies heavily on the identification of relevant and irrelevant features in high-dimensional datasets. This research provides a strategy for reducing the number of features in an ensemble in order to better classify web-attacks. Information security policy breaches are known as intrusions. Intrusion detection (ID) is a set of actions for detecting and recognizing suspicious behaviors that make the expedient acceptance of standards of secrecy, quality, consistency, and availability of a computer-based network system more difficult. Using the GOA technique, we describe a new approach to feature selection and classification for the NSL-KDD cup 99 intrusion detection dataset. As a primary goal, it is to reduce the number of features used in training data for intrusion classification. Features are selected and eliminated in supervised learning in order to improve classification accuracy by focusing on the most significant input training features and eliminating those that are less important. Several input feature subsets of the training dataset of NSL-KDD cup 99 dataset were used in the experiment to test the classifier.**

***Keywords*: Intrusion Detection, Ensemble Learning, GA Algorithm, Features Selection, GAO Algorithm.**

I. INTRODUCTION

Web mining is the process of generation of human readable information from web log data. The website's server is the primary source for web log data.. Web mining converts the server web log data into easily understandable information. This systematic human readable information is very useful for security, improvement and maintenance purposes of websites.

Web mining uses nearly all mathematical models of data mining. So, web mining may be defined as the branch of data mining in which web server logs are used as data. Web mining is the analysis of web server log data for the sake of generation of valuable information. Analysis of server log data gives marvelous insights about visitor's navigational behavior, development of personalization systems, accessing website security, target marketing, improvement/development of websites, improving server's performance and many more. Web logs contain structured and unstructured data. Depending on the data type, web mining is further categorized in three parts [Figure 1.1].

- Web-Content-Mining
- Web-Structure-Mining
- Web-Log/Usage-Mining

Web-Content-Mining

This mining is related to the generation of useful patterns from elements of a website like images, tables, text, audio, video, pdf, graphics etc. There are two approaches to web content mining. First one related to the mining of the contents of web pages. It gives mining results based on the type of content. Second approach based on the target of mining or improving web search results of search engines. The classification of websites based on contents on web pages is an example of the second approach of web content mining.

Web Structure Mining

Web structure mining deals with the hyperlink connectivity within the webpage and in between the webpages. Web Structure Mining explains the hyperlink structure of a website to obtain the insight for improvement of the website. Websites link structure analyzed in web structure mining. The relationship between websites can be analyzed by organizing link structure in the form of topology. Web page ranking and website reorganization are two direct applications of web structure mining.

Web Log/Usage Mining

The analysis of web log data for discovery of navigational patterns and analysis of these discovered patterns for generation of standard rules comes under web log mining. Web usage mining refers to extraction of valuable usage patterns from the analysis of web log data. Thus 'web log mining' and 'web usage mining' represent the same meaning. Web usage mining is the analysis of web log data for generation of standard rules and usage patterns.

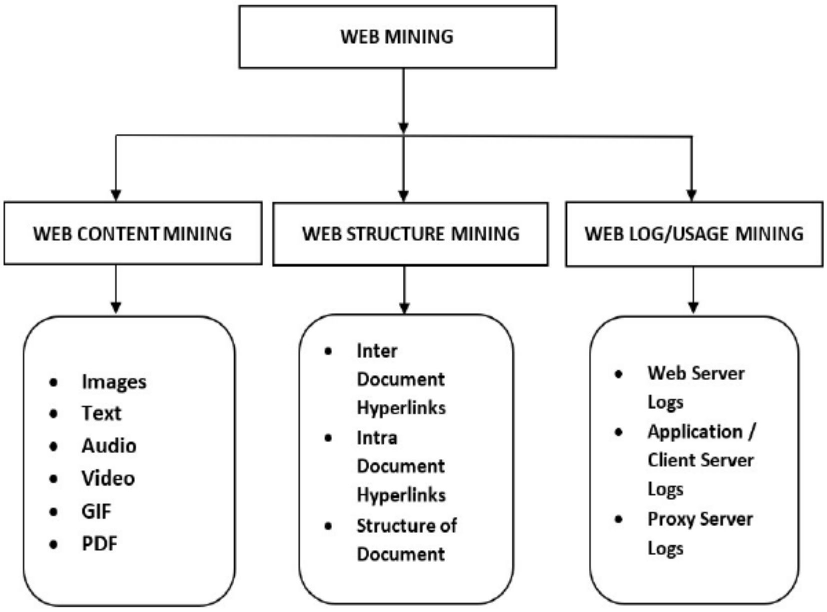


Figure 1: Types of Web mining.

In web logs each clickstream of a visitor's traversal is recorded. So, web log data is an immense source of information about visitors of websites. It contains information of visitors like IP address, user's identification details, date and time of visit, bytes transferred, status code of request, user's system information (user agent), resource requested, referrer websites, protocol used for request, Cookie etc. Analysis of this information is vital for web administrators for website development, website improvement, personalization, security and other visitors related issues.

Srivastava J et al described the complete web mining process for mining web log data, giving special attention to commercial applications [1]. Agarwal et al initially presented a most researched pattern discovery association rule mining algorithm [2]. Cooley R et al discussed different pattern discovery methods for web log data [3]. Researchers [4] given detailed analysis of pre-processing methods for web usage mining. They provided the WEBMINER system for the complete web mining process. WEBMINER incorporated pre-processing, knowledge discovery and pattern analysis phases in the system architecture. According to Cooley et al [5] if a unique IP accesses a web page without using a hyperlink, then the user will be counted as a new user. Martin Arlitt and Carey Williamson used the data from server logs of NASA that was collected by Jim Dumoulin of the Kennedy Space Center [9]. They explained the various features of workloads of the web server.

II. Related Work

The logs were analysed. Log analysis has been used to increase software system dependability[35] in a variety of ways, including anomaly detection[10],[28],[47], failure diagnosis[17],[31],[38], programme verification[11],[42], and act prediction[16]. The majority of these log analysis approaches are divided into two steps: log parsing and log mining, both of which have received a lot of attention in current years. He et al.[24] compare the efficiency of four non-system source code offline log parsing methods: SLCT[45], IPLOM[29], LogSig[44], and LKE[20]. [34] proposes an offline log parsing solution which requires linear time and space. Using system sources, Xu et al.[47] offer an online log processing method. Xu et al[47] employ PCA to find abnormalities, with the input being a matrix built from logs.

Beschastnikh et al.[11] create a finite state machine that defines system runtime behaviour using system logs. Unlike these articles, which use log analysis to resolve a range of complications, we focus on log-based anomaly detection methods.

Anomalies Detection

Anomaly detection is the process of looking for out-of-the-ordinary behaviour that can be stated to manual examination and debugging engineers. Bovenzi et al.[13] present an operating system-level method for detecting abnormalities that is suited for mission-critical systems. Venkatakrishnan et.al[46] identify safety vulnerabilities before a system is compromised.

In contrast to past efforts that concentrated on discovering individual anomalies, this study analyses the efficiency of anomaly detection strategies for generic irregularities in large-scale systems. Babenko et al.[9] offer an algorithm for automatically creating explanations from anomaly-detected failures.

Empirical research

Since empirical research may often provide practical insights to both academics and developers, there has been a lot of empirical research on software dependability in recent years. Yuan et al.[48] investigate open-source logging practises and offer advice to developers.

Fu et al.[21],[49] investigate the logging industry empirically. Pecchia and colleagues [37] look into the goals and difficulties of logging in industrial settings. The use of decision tree approaches to detect smells in code is investigated by Amorim and colleagues [7]. Lanzaro and his colleagues [25] look on how library code flaws emerge as interface issues. [40] Take a look at long-living bugs from five different angles. Milenkoski and colleagues[33] investigate and organise typical computer intrusion detection approaches. Take, for example, Chandola. [14] Survey anomaly detection methods that employ machine learning practices in a range of domains, but this research focuses on assessing and evaluating existing work that employs log analysis to discover system anomalies.

Review of Log Anomalies and Deep Learning

To identify suspect business-specific activity and user profile behaviour, T.F. Yen et al. [29] used SIEM log data composed from over 1.4 billion logs each day. Scalability, data noise, and a lack of ground truth were all challenges for this project. The suggested solution demands the generation of a feature vector based on historical data for each internet host. To detect potential security problems, they utilise unsupervised clustering using data-specific characteristics. Manual labelling experts must be aware of the absence of ground-based reality. The technique is rule-based, and historical log processing requires subject-matter expertise. Min Du et al. [2] proposed an architecture for detecting anomalies in log data that does not need any former knowledge of the domain. The proposed method includes a process for diagnosing log key and parameter value abnormalities, as well as a mechanism for identifying log key and parameter value abnormalities from logs. The probability of the next log key is predicted using a neural network-based method.

A log parameter sequence abnormality can similarly be detected using a comparable LSTM neural network. The software also uses false-positive manual feedback to improve future accuracy. The LSTM considers the log series to be a natural language sequence that may be processed accordingly. Using datasets from BGL, Thunderbird, Open Stack, and IMDB, Amir Farzad et al. [6] suggested a deep learning model for detecting log message abnormalities and compared these models to boost efficiency. The IMDB dataset is used to demonstrate how their method can be used to a range of classification challenges.

Natural Language Processing techniques were used by Mengying Wang et al. [1] to discover abnormal log messages. In the research, word2vec and TF-IDF feature extraction methods are applied, and the activity is finished with a classification LSTM deep learning algorithm. They discovered that word2vec beats TF-IDF in log message identification jobs.

W Meng et al. 2019 [4] created an attention-based LSTM model that could simultaneously detect both successive and computable irregularities. It uses FT-Tree to analyse logs and has developed template2vec, a new word representation method that uses synonyms and antonyms to effectively discover anomalies. When only the log template index is evaluated in [2], and the semantic log connection cannot be provided, this solution tackles the issue of losing key log information. Xiaojuan Wang et al. [3] used NetEngine40E to collect router logs and analyse behaviour type, attributes, and rank.

The projected model is an LSTM neural network that analyses the amount of logs over time to forecast log spikes. The Aspect syntax forest is also utilised for attribute data semantic analysis. The work has been extended to identify logs that are the cause of log spikes based on attribute information and value. "Robust Log," one of the most current log anomaly detection approaches, was suggested by Xu Zhang et al. [8]. They built a BiLSTM classification model based on the vector demonstration of each log event and the semantic information included in the log's semantic vector. FastText [36] word vectorization and TF-IDF-based aggregation are utilised to generate log event vectors. As demonstrated by the development of a synthetic HDFS log dataset, the robust-log appears to operate well in unstable log events. The issues presented by academics, as well as the Deep Learning models, datasets, and approaches used in various log analysis research projects, are summarized in Table 1.

Table 1 “Summary of Challenges Addressed and methods Used By Various Authors on Log Anomaly Detection”

Year	Citation	Challenges Addressed	DL Model	Data Set	NLP /Other Method	Pre-Processing / Parsing
2019	Xiaojuan Wang [3]	Time period extracted Semantics were also expressed by anomalies and reasons of Log surge.	LSTM	Netengine 40E Router Log	Directed Graph	Parsed on Behavior type
2019	Xu Zhang [8]	Relevant knowledge of log sequences was aided by log data instability.	Bi-LSTM with Attention	HDFS, Other security system of Microsoft	FastText	Drain
2019	WeibinMeng[4]	When only logtemplates are used, it is possible to discover both sequential and quantitative anomalies at the same time.	LSTM	BGL, HDFS	Template2Vec	FT-Tree
2019	Amir Farzad [6]	In log message classification and anomaly detection, LSTM and Bi-LSTM models with autoencoders are utilised.	Auto-LSTM, Auto-BLSTM,Auto-GRU	BGL, IMDB, Open stack, Thunderbird	Word Frequency	-
2018	Siyang Lu [5]	The performance of CNN with LSTM and Multilayer Perceptron for log anomaly detection was compared (MLP)	CNN based model	HDFS	LogKey2Vec	Logs-Key Sequences and session key
2018	Andy Brown[9]	Concentration has an effect on sequence modelling.	LSTM with 5 attention mechanism	LANL, cybersecurity dataset	-	Language Modeling and Tokenization
2018	Mengying Wang [1]	For log anomaly detection, NLP approaches such as word2vec and TFIDF are used.	LSTM	Thunderbird	Word2Vec , TF-IDF	Data Cleaning of Logs
2017	Min Du [2]	Workflows are used to detect and analyze anomalies based on "Log Key" and "Parameter Value."	LSTM	HDFS, Openstack	Log Key, Parameter value and Workflow	Spell

III. Attacks in Web Log Data

DARPA and the Air Force Research Laboratory (AFRL) have funded MIT Lincoln Laboratory to collect and distribute datasets for the evaluation of computer network intrusion detection system research under their sponsorship.

Data from the KDDCup99 dataset are derived from the DARPA benchmark [11]. KDDCup99 is a four-gigabyte dataset of compressed binary TCP dump data culled from seven weeks of network traffic and broken down into around five million records, each containing about 100 bytes of information on the connections that were made. The two weeks of test data include roughly two million records of connections. To differentiate between a normal and an attack connection, each KDDCup'99 training connection record has 41 attributes and is designated as such [11].

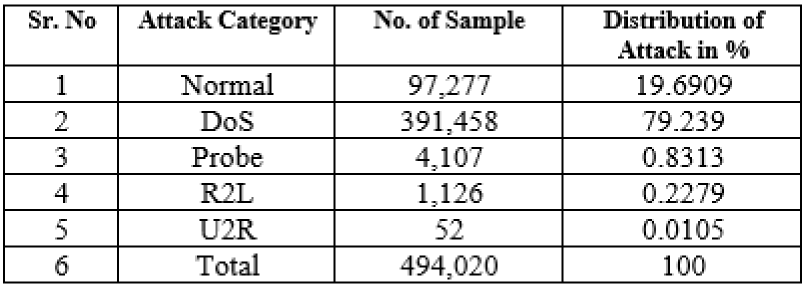
The KDDCup'99 training set has 494,020 records, whereas the KDDCup'99 test set has 311,029 records. As a result of this categorization of the dataset's numerous attack types, the detection rate of comparable attacks can be improved. The training set includes 24 attack kinds, while the test set includes 38 attack types, 14 of which are innovative attacks. Dos, Probing, Remote to Local (R2L) and User-to-Root are the four main categories of attacks in the dataset (U2R). KDDCup'99 training set attacks are shown in Table.1, along with additional attacks found in KDDCup'99 test set.

Table 2: Attacks in KDD Dataset

Sr. No.	Attack Category	Attacks in KDDCup'99 Training set	Additional attacks in KDD Test set
1	Dos	back, neptune, smurf, teardrop, land, pod.	apache2, mail bomb, process table.
2	Probe	satan, portsweep, ipsweep, nmap.	mscan, saint.
3	R2L	warezmaster, warezclient, ftpwrite, guess password, imap, multihop, phf, spy	sendmail, named, snmpgetattack, snmpguess, xlock, xsnoop, worm.
4	U2R	rootkit, buffer overflow, load module, perl.	httptunnel, ps, sqlattack, xterm

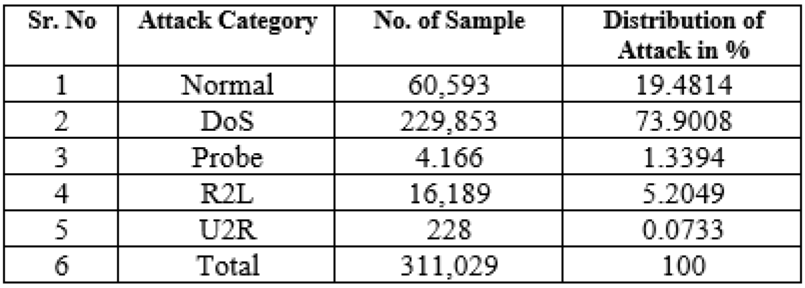
Table.3 and Table.4 shows the number of records for each attack category in the training and testing datasets respectively.

Table 3: Sampling & % Distribution of Attack



Sr. No	Attack Category	No. of Sample	Distribution of Attack in %
1	Normal	97,277	19.6909
2	DoS	391,458	79.239
3	Probe	4,107	0.8313
4	R2L	1,126	0.2279
5	U2R	52	0.0105
6	Total	494,020	100

Table 4: Sampling & % Distribution of Attack



Sr. No	Attack Category	No. of Sample	Distribution of Attack in %
1	Normal	60,593	19.4814
2	DoS	229,853	73.9008
3	Probe	4.166	1.3394
4	R2L	16,189	5.2049
5	U2R	228	0.0733
6	Total	311,029	100

In both the training and test sets, the attack percentage distribution is not the same. Both the test set and training set have a different chance of attack distribution. For example, there are only 0.2 percent R2L attacks in the training set, but there are 5.2 percent in the testing set. This is one of the most difficult parts of the assault classification process.

IV. Proposed Algorithm

4.1 Genetic Algorithms

It is the crossover and mutation operators that are utilized to preserve population diversity and avoid local optima in genetic optimization. To make matters worse, as populations evolve, crossover and mutation probabilities remain constant, delaying algorithm convergence until much later in the process, which in turn leads to the lengthy training time of GOA. Since the crossover and mutation probabilities of GA are influenced by the fitness value of the population, this paper's strategy develops a population that speeds up search in early evolution and increases convergence in later evolution by adjusting these probabilities.

A. Selection Operators

GA's selection process is designed to ensure that the population is diverse and that the best people are recruited. The offspring population chooses the person based on their fitness value, which gives the better individuals a better chance of being selected. Operators like as roulette are widely available , wheel, elitist and tournament selection are all examples of this.

Roulette wheel selection: The selection of a chromosome in the population is inversely related to its fitness value, as demonstrated by the roulette wheel. A "roulette wheel" is assigned to each member of the population "and the wheel goes around N times, where N is the individual's fitness value (N is the number of individuals in the population). The chromosome under the wheel mark is chosen for the following generation in each spin.

Elitist selection: The individual with the highest fitness value in the population does not participate in crossover or mutation and is utilized to replace the individual with the lowest fitness value following crossover and mutation. Elitist selection prevents the crossover or mutation operator from destroying the best individual.

Tournament selection: Selection is done by a series of "tournaments." "Two people are randomly selected, and only one of them has a higher fitness value than the other, and that person is passed down to the next generation. The tournament selection method was used in this study to choose the best 60% of chromosomes.

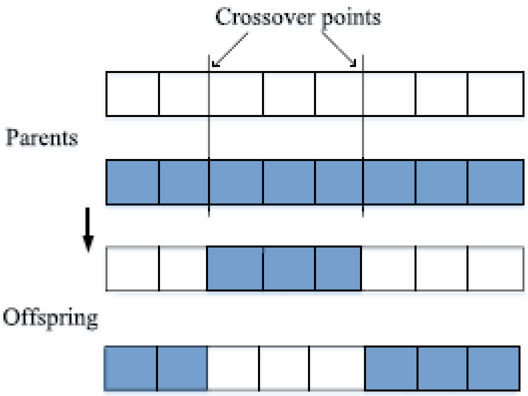


Fig 2 Crossover Operators and Their Effects in Generation of Offspring.

B. Optimized Crossover Probability

In order to create a new individual, the parent chromosomes' partial structure is replaced and recombined, and this procedure is known as the crossover operation.

The population is getting closer to the optimal solution set as the population's evolutionary algebra increases, and this is why we want to emphasize the following points.

Individual crossovers must be increased in the early stages of the population's evolution in order to quickly search across the whole definition space. It is necessary to decrease the number of crossovers at this stage of population evolution in order to keep excellent genes from being lost, which will speed up the convergence of GAs.

An increase in the individual crossover probability boosts the chances of producing great individuals when average fitness values are low. An individual's likelihood of crossing over should decrease as the population's average fitness value gets closer to the ideal solution.

The following is a summary of the adaptive crossover probability in genetics:

$$P_c = \frac{P_{c_0} + P_{c_1}}{2}$$

$$= \frac{\left(\left(\frac{N-n}{N} P_{cmax} + \frac{n}{N} P_{cmin}\right) + \frac{P_{cmax} \cdot f_{min}}{f_{max}}\right)}{2}$$

With the increase of population evolutionary algebra, the value of Pc0 decreases, and Pc1 decreases when the average fitness value of the population tends toward the ideal value. In this case, P_{cmax} is the maximum crossover probability, P_{cmin} is the minimum crossover probability, f_{min} is the minimum fitness value of the cut-off for the current population, f_{max} is the maximum fitness value of the cut-off for the current population, n is the current evolutionary algebra, and N is the evolutionary algebra of the entire population.

When the population's average fitness level approaches the ideal, the value of P_{c0} and P_{c1} decline as population evolutionary algebra increases. An example of this would be if P_{cmax} is the maximum crossover probability and P_{cmin} the minimum, as well as f_{min} and f_{max} being the minimum fitness value cut-offs for current populations, and n and N being current evolutionary algebra and total population evolutionary algebra, respectively, as well as the fitness value cut-offs for the current populations.

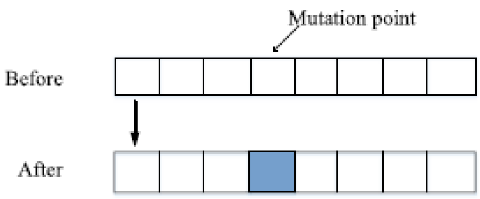


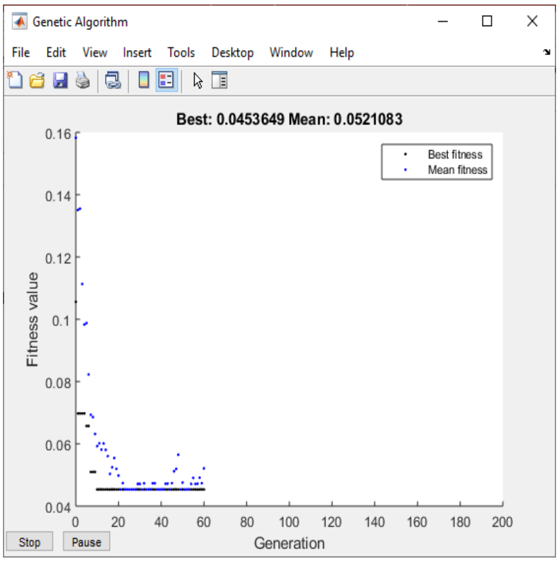
Fig 3. Mutation Operators and their Effects in Generation of the Offspring.

We carried out two different sets of experiments to compare the two different hypotheses. In the first experiment, we tested the detection rate of the DoS assaults that occur often in the network using all of the dataset's 41 attributes. Initially, a Decision Tree is built, and then an updated GOA algorithm is utilized to improve its classification accuracy. To replicate Table 4's experiment, we use Enhanced GOA with Decision Tree as its base learner to select the 15 characteristics. Table 5 displays the findings of the research.

Table 5: The attack detection rate of Dos Attack

No. of features	% of detection rate	Training Time(sec)	Test Time(sec)
41	97.8	9.7	0.43
15	98.9	6.2	0.26

When 15 features are taken into account, the system takes only 6.2 seconds to train. Additionally, the incoming network connection is tested in just 0.26 seconds.



(a)

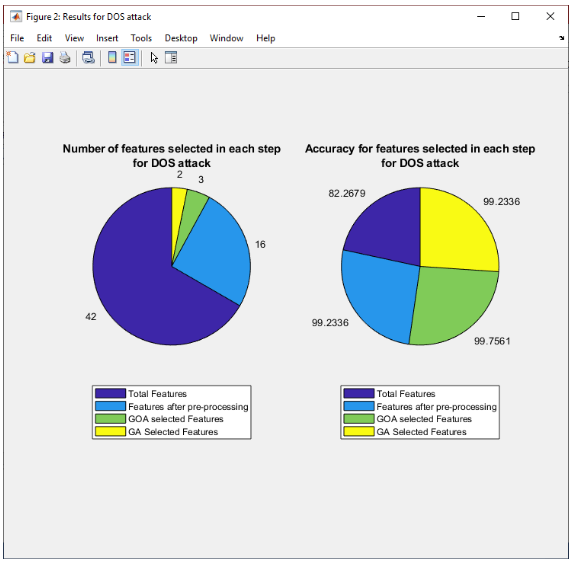
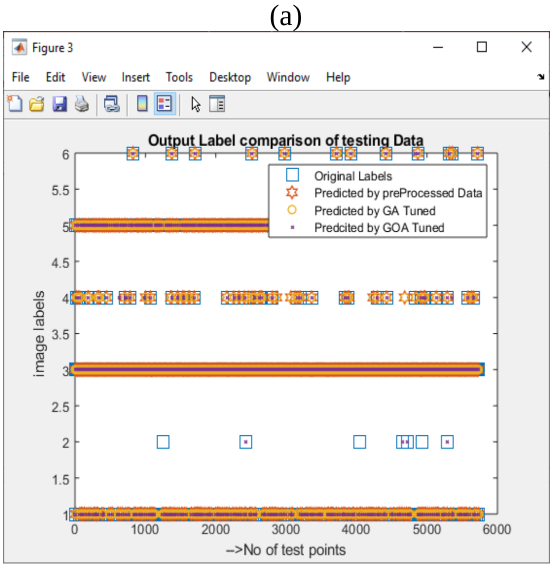
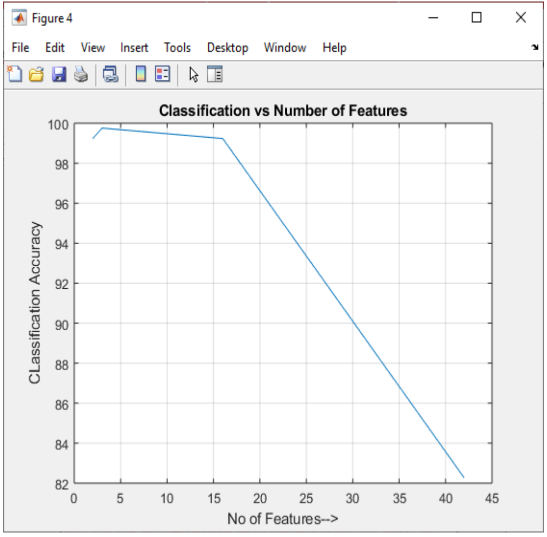


Figure 2: Results for DOS attack
File Edit View Insert Tools Desktop Window Help
Number of features selected in each step for DOS attack
2
3
16
42
Accuracy for features selected in each step for DOS attack
82.2679
99.2336
99.2336
99.7561
Total Features
Features after pre-processing
GOA selected Features
GA Selected Features
Total Features
Features after pre-processing
GOA selected Features
GA Selected Features

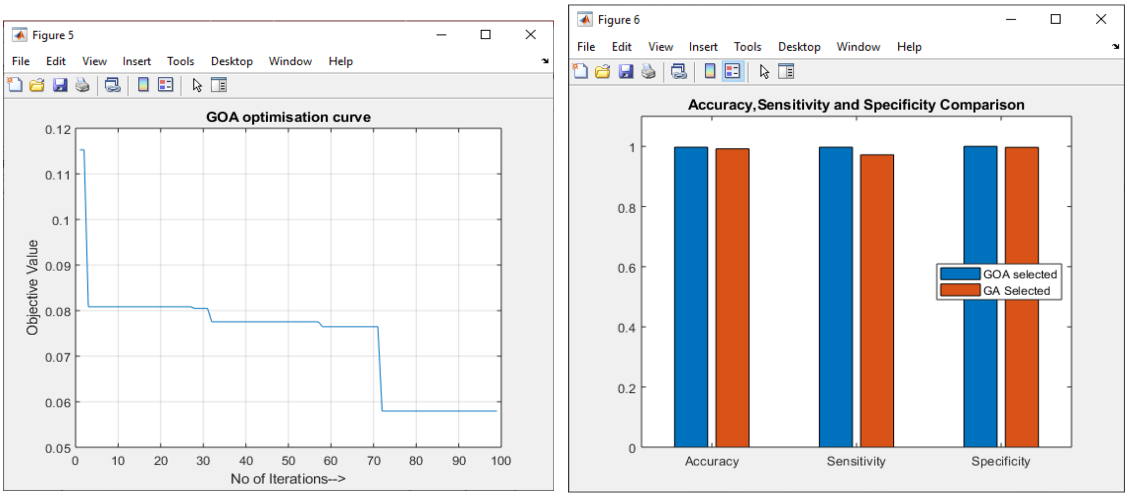
(b)



(c)



(d)



(e) (f)

Fig 4 (a) Genetic Algorithm Best Value (b) No. of features & Accuracy for each step for DoS Attack (c) Outpul Label (d) Classification Accuracy (e) GOA Optimization Curve (f) Comparative Analysis of Accuracy , Sensitivity & Specificity

Table 6 Comparative Analysis of Genetic Algorithm and Grasshopper Optimization Algorithm

Sr. No.	Parameter	GA Selected	GOA Selected
1	Accuracy	0.9923	0.9976
2	Sensitivity	0.9721	0.9972
3	Specificity	0.9968	0.9999

V. CONCLUSION

An ensemble feature reduction method is proposed in this paper for the detection of web attacks. Additionally, the method uses an ensemble of filter-based feature selection algorithms to identify a reduced feature subset from the original feature set. According to this new approach to feature selection, there are a total of 16 decreased feature sets from the original 78. Research reveals that an ensemble feature reduction method is more accurate than other methods. In comparison to current state-of-the-art systems with limited feature subset, the implemented system with J48 produces promising results. Filter and wrapper approaches can be combined in this study to provide an ensemble method for the discovery of the optimal feature subset.

For a human-centered smart IDS, this study presents an alarm intrusion detection algorithm (GA-GOA), which is built on the GA and GOA algorithms. Using the GA population search technique and the capacity of individuals to exchange information by maximizing the crossover and mutation probabilities of GA, this research first and foremost makes efficient use of these features. Convergence of the algorithm and GOA training speed are both improved as a result of the new algorithm. The GOA error rate can be reduced while the true positive rate can be increased using a novel fitness function that has been developed. As a result, the accuracy of SVM is enhanced while also optimizing the kernel parameter, penalty parameter C, and feature weights all at once An improvement in intrusion detection based on genetic algorithms and Grasshopper Optimization is presented in this paper, which increases detection rates, accuracy, and true positives while decreasing false positives and shortening SVM training time. These findings are supported by simulations and experiments.

A multi-core processor with each step of the filter assigned to an individual core of the processor could be a future subject of research. This would increase the utilization of the processor and also enhance the detection time of the attraction.

References

[1] Xuan Dau Hoang, Jiankun Hu, and Peter Bertok, “A program-based anomaly intrusion etection scheme using multiple detection engines and fuzzy inference,” Journal of Network and Computer Applications, Vol. 32, pp. 1219-1228, 2009.

[2] 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, pp. 18-28, 2009.

[3] Weiming Hu, Wei Hu and Steve Maybank, “AdaBoost-Based Algorithm for Network Intrusion Detection,”

IEEE Transactions on Systems, Man and Cybernetics, Vol. 38, pp. 577-583, April-2008.

[4] N. B. Amor, S. Benferhat, and Z. Elouedi, “Naïve Bayes vs. decision trees in intruison detection systems,” In Proc. of the 2004 ACM Symposium on Applied Computing, New York, pp. 420-424, 2004.

[5] Dewan Md. Farid, Nouria Harbi and Mohammad Zahidur Rahman, “Combining naive bayes and decision tree for adaptive intrusion detection”, International Journal of Network Security & its Applications, Vol. 2, No. 2, pp. 12 - 25, 2010.

[6] Natesan P, Balasubramanie P and Gowrison G, "Adaboost with Single and Compound weak classifier in Network Intrusion Detection”, In Proceedings of International conference on Advanced computing, Networking & Security” , Vol. 1, pp 282-290, Dec-2011.

[7] Xiang C, Yong PC, Meng, “Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees”. Pattern Recognition Letters 29(7):918–924, 2008.

[8] Gupta KK, Nath B (2010) Layered approach using conditional random fields for intrusion detection. IEEE Trans Dependable Secure Computing 7(1):35–49, 2010.

[9] Kok-Chin Khor, Choo-Yee Ting and Somnuk Phon-Amnuaisuk A, "cascaded classifier approach for improving detection rates on rare attack categories in network intrusion detection", Journal of Applied Intelligence, DOI 10.1007/s10489-010-0263-y.

[10] Natesan P, Balasubramanie P and Gowrison G, "Design of two stage filter using enhanced Adaboost for improving attack detection rates in network intrusion detection”, Journal of Computer science, Information technology and Security, Vol.2, No.2, pp. 349-357.

[11] KDDCup99 Dataset, http://kdd.ics.uci.edu/databases/kddcup99/ kddcup99.html. 1999.

[12] Z.Pawlak, “Rough sets”, International Journal of Computer and Information Sciences, Vol.11, No. 5, pp. 341-356, 1982.

[13] Lei Shi, Li Zhang, Xinming Ma and Xiaohong Hu, “Rough set Based Personalized

[14] Recommendation in Mobile Commerce. 2009 International conference on Active Media Technology”, Lecture Notes in Computer Science, 370-375, 2009.

[15] Sabhnani, M. R., & Serpen, G. Application of machine learning algorithms to KDD intrusion detection dataset with in misuse detection context. In Proceedings of the international conference on machine learning: Models, technologies, and applications. pp. 209–215, 2003.

[16] Xuren, W., Famei, H., & Rongsheng, X. Modeling intrusion detection system by discovering association rule in rough set theory framework. In Proceedings of the international conference on computational intelligence for modeling control and automation, and international conference on intelligent agents. Web Technologies and Internet Commerce (CIMCAIAWTIC’ 06), 2006.

[17] J.W. Han and M.Kamber, Data Mining: Concepts and Techniques, 2nd edition. Morgan Kaufmann, pp. 310-318, 2006.

[18] N.Friedman, D.Geiger and M.Goldsmidt, “Bayesian Network Classifiers,” Machine Learning, Vol.29, pp 131-163, Nov 1997.

[19] Yoav Freund, Robert E.Schapire. “A Decision theoretic generalization of on-line learning and an application to boosting,” Journal of Computer and System Sciences. Vol. 55, 119-139, 1997.

[20] P.N. Tan, Introduction to Data Mining. Reading, MA:Addison-Wesley, 2006.

[21] Shi-Jinn Horng, Ming-Yang Su, Yuan-Hsin Chen, Tzong-Wann kao, Rong-Jian Chen, Jui-Lin Lai, Citra Dwi Perkasa, “A novel intrusion detection system based on hierarchical clustering and support vector machines”, Expert Systems with Applications, Vol.38, pp.306-313, 2010, DOI:10.1016/j.eswa.2010.06.066.

[22] Pfahringer, B. Winning the KDD99 classification cup: Bagged boosting SIGKDD Explorations, 1(2), 65–66, 2000.

[23] Simpson SJ , McCaffery A , HAeGELE BF . A behavioural analysis of phase change in the desert locust. Biol Rev 1999;74:461–80 .

[24] Rogers SM , Matheson T , Despland E , Dodgson T , Burrows M , Simpson SJ . Mechanosensory-induced behavioural gregarization in the desert locust Schis- tocerca gregaria. J Exp Biol 2003;206:3991–4002 .

[25] Topaz CM , BernoffAJ , Logan S , Toolson W . A model for rolling swarms of lo- custs. Eur Phys J Special Top 2008;157:93–109 .

[26] Dewan Md. Farid, Li Zhang, Chowdhury Mofizur Rahman, M.A. Hossain, and Rebecca Strachan, “Hybrid Decision Tree and naıve Bayes classifiers for Multi-class classification Tasks,” Expert Systems with Applications, Vol. 41, Issue 4, Part 2, March 2014, pp. 1937-1946.

[27] Dewan Md. Farid, Li Zhang, Alamgir Hossain, Chowdhury Mofizur Rahman, Rebecca Strachan, Graham Sexton, and Keshav Dahal, “An Adaptive Ensemble classifier for Mining Concept Drifting Data Streams,” Expert Systems with Applications, Vol. 40, Issue 15, November 2013, pp. 5895-5906.

[28] Dewan Md. Farid, and Chowdhury Mofizur Rahman, “Mining Complex Data Streams: Discretization, Attribute Selection and classification,” Journal of Advances in Information Technology, Vol. 4, No. 3, August 2013, pp.

129-135.

[29] Dewan Md. Farid, and Chowdhury Mofizur Rahman, “Assigning Weights to Training Instances Increases classification Accuracy,” International Journal of Data Mining & Knowledge Management Process, Vol. 3, No. 1, January 2013, pp. 13-25.

[30] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, “Mining Complex Network Data for Adaptive Intrusion Detection,” Advances in Data Mining Knowledge Discovery and Applications, ISBN 978-953-51-0748-4, INTECH, September 2012, pp. 327-348.

[31] Amit Biswas, Dewan Md. Farid, and Chowdhury Mofizur Rahman, “A New Decision Tree Learning Approach for Novel Class Detection in Concept Drifting Data Stream classification, Journal of Computer Science and Engineering, Vol. 14, Issues 1, July 2012, pp. 1-8.

[32] Fauzia Yasmeen Tani, Dewan Md. Farid, and Mohammad Zahidur Rahman, “Ensemble of Decision Tree classifiers for Mining Web Data Streams, International Journal of Applied Information Systems, Vol. 1, No. 2, January 2012, pp. 30-36.

[33] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, “An Ensemble Approach to classifier Construction based on Bootstrap Aggregation, International Journal of Computer Applications, Vol. 25, No. 5, July 2011, pp. 30-34.

[34] Dewan Md. Farid, Mohammad Zahidur Rahman, and Chowdhury Mofizur Rahman, “Adaptive Intrusion Detection based on Boosting and Nave Bayesian classifier, International Journal of Computer Applications, Vol. 24, No. 3, June 2011, pp. 12-19.

[35] A. J. M. Abu Afza, Dewan Md. Farid, and Chowdhury Mofizur Rahman, “A Hybrid classifier using Boosting, Clustering, and nave Bayesian classifier, World of Computer Science and Information Technology Journal, Vol. 1, No. 3, April 2011, pp. 105-109.

[36] Dewan Md. Farid, Nouria Harbi, and Mohammad Zahidur Rahman, “Combining Nave Bayes and Decision Tree for Adaptive Intrusion Detection, International Journal of Network Security & Its Applications, Vol. 2, No. 2, April 2010, pp. 12-25.

[37] Dewan Md. Farid, Jerome Darmont, and Mohammad Zahidur Rahman, “Attribute Weighting with Adaptive NBTree for Reducing False Positives in Intrusion Detection, International Journal of Computer Science and Information Security, Vol. 8, No. 1, April 2010, pp. 19-26.

[38] Dewan Md. Farid, and Mohammad Zahidur Rahman, “Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm, Journal of Computers, Vol. 5, No. 1, January 2010, pp. 23-31.

[39] Y. Li, J. L. Wang, Z. H. Tian, T. B. Lu, and C. Young, “Building lightweight intrusion detection system using wrapper-based feature selection mechanisms, Computer Security, vol. 28, No. 6, September 2009, pp. 466-475.

[40] S. Mukkamala and A. H. Sung, “Feature selection for intrusion detection with neural networks and support vector machines, Journal of the Transportation Research Board, Vol. 1822, 2003, pp. 33-39.

[41] A. Frank, and A. Asuncion, “UCI machine learning repository,” University of California, Irvine, 2010, http://archive.ics.uci.edu/ml, Accessed 26.08.2014.

[42] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, “The WEKA Data Mining Software: An Update,” SIGKDD Explorations, Vol. 11, No. 1, 2009, http://www.cs.waikato.ac.nz/ml/weka/, Accessed 26.08.2014.