Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.8 No.2, 2019, pp.25-31
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajcst-2019.8.2.2145

IDSFS: A Signature Based Intrusion Detection System with High Pertinent Feature Selection Method

S. Latha[1] and Sinthu Janita Prakash[2]
[1]Assistant Professor, [2]Head & Professor, PG & Research
[1&2]Department of Computer Science, Cauvery College for Women, Tiruchirapalli, Tamil Nadu, India
E-Mail: slathanagaraj@gmail.com

(Received 21 February 2019; Revised 5 March 2019; Accepted 12 April 2019; Available online 20 April 2019)

Abstract **- Securing a network from the attackers is a challenging task at present as many users involve in variety of computer networks. To protect any individual host in a network or the entire network, some security system must be implemented. In this case, the Intrusion Detection System (IDS) is essential to protect the network from the intruders. The IDS have to deal with a lot of network packets with different characteristics. A signature-based IDS is a potential tool to understand former attacks and to define suitable method to conquest it in variety of applications. This research article elucidates the objective of IDS with a mechanism which combines the network and host-based IDS. The benchmark dataset for DARPA is considered to generate the IDS mechanism. In this paper, a frame work IDSFS - a signature-based IDS with high pertinent feature selection method is framed. This frame work consists of earlier proposed Feature Selection method (HPFSM), Artificial Neural Network for classification of nodes or packets in the network, then the signatures or attack rules are configured by implementing Association Rule mining algorithm and finally the rules are restructured using a pattern matching algorithm-Aho-Corasick to ease the rule checking. The metrics like number of features, classification accuracy, False Positive Rate (FPR), Precision, Number of rules, Running Time and Memory consumption are checked and proved the proposed frame work's efficiency.**

Keywords: Feature Selection, Intrusion Detection System, Association Rule Mining, Apriori Algorithm, Artificial Neural Network, Aho-Corasick Pattern Matching Algorithm, Gain Ratio, Chi-Square Analysis

I. INTRODUCTION

The system which involves in action of identifying attacks in a network may be either host-based IDS (HIDS) or network-based IDS (NIDS). The nodes with suspicious intent can be identified from the patterns or attack signatures which are observed from the previous history. These patterns can be observed from the log files or from the network traffic. When this pattern checking is for network traffic, then it is network-based IDS. The IDS can work dynamically when it consists of both HIDS and NIDS.

The data source for NIDS [1] is the raw packets in a network. A NIDS must assess the entire network traffic flow continuously. Its attack recognition module has four collective methods to diagnose the attack signature: (i) Pattern, expression or byte code matching [2], (ii) Frequency or threshold crossing [3], (iii) Correlation of lesser events, (iv) Statistical anomaly detection [4].

When an attack is noticed, the IDS provide range of possibilities to identify, attentive and take action in response to the attack. The response by an IDS may be a procedural notification, action to termination and/or session recording for forensic study and evidence collection.

Host-based intrusion detection systems [5] continue as a potent tool for understanding prior attacks and determining proper methods to conquest their imminent application. Host-based IDS still custom the audit logs, however they are much more mechanical, having changed with refined and receptive detection techniques. Host based IDS [6] stereotypically monitor system, event, and security logs on Windows NT and syslog in UNIX environments. When any of these files undergone any modification, the IDS check the new log entry with attack signatures for a match. If so, the system reacts with administrator alerts and other calls to action.

HIDS have developed to comprise other technologies. The popular method for perceiving intrusions checks key system files and executable via checksums at steady intervals for unpredicted fluctuations. The appropriateness of the response is in direct connection to the regularity of the system. Finally, some products hang on to port movement and alert administrators when explicit ports are identified. This sort of detection fetches the fundamental level of network-based intrusion detection into the host-based setting.

A survey about techniques like feature selection, Classification, Rule generation and Pattern matching algorithms is presented in section 2. The proposed methodology IDSFS is explained in section 3 with its framework, algorithms and the method of working. The experimental results and discussion are in section 4 and finally this paper is concluded in section 5.

II. RELATED WORKS

As per Jianglong Song, *et al.*, [7], presented the redundant and irrelevant features cause high resource consumption and similarly worsen the performance of IDS, that too primarily with big data. Accordingly, they presented a novel technique wherein, the principal phase conducts an initial quest for an ideal subset of features using chi-square feature selection. Then those selected features are augmented using the Random Forest (RF).

M.S Irfan Ahmed, *et al.*, [8], stated that data preprocessing before categorizing detections would certainly progress the results in different dimensions. The authors evidenced the usage of Information Gain technique for pre-processing the NSL-KDD dataset and also applied the J48 classification technique on it.

Longjie Li, *et al.*, in [9] presented a novel hybrid model with the purpose of detecting network intrusion effectively. In the proposed model, Gini index is used to select the optimal subset of features, the gradient boosted decision tree (GBDT) algorithm is adopted to detect network attacks, and the particle swarm optimization (PSO) algorithm is utilized to optimize the parameters of GBDT.

M.R. Gauthama Raman [10] presented a novel approach, based on Helly property of Hyper graph and Arithmetic Residual based Probabilistic Neural Network (HG AR - PNN) to address the classification problem in IDS.

Yu Wang, *et al.*, [11] with the advent of fog computing, they proposed a privacy-preserving framework for signature-based intrusion detection in a distributed network based on fog devices.

Yehonatan Cohen, *et al.*, [12] showed that malicious webmail attachments are unique in the manner in which they propagate through the network. The authors leveraged these findings for defining novel features of malware propagation patterns.

III. IDSFS: A SIGNATURE BASED INTRUSION DETECTION SYSTEM WITH HIGH PERTINENT FEATURE SELECTION METHOD

The proposed framework for Signature Based Intrusion Detection System with High Pertinent Feature Selection Method (IDSFS) is presented in Figure 1. It has four stages as follows:

Stage 1:

Pre-Processing by Proposed HPFSM

A novel High Pertinent Feature Selection method is proposed by hybridizing the Chi-Squared analysis and Gain Ratio feature selection methods as first step. This proposed HPFSM is used to lessen the dimension of the DARPA dataset. Proposed HPFSM picks only the strongly relevant and low redundant structures from the dataset for advance processing [13].

Stage 2:

Classification

Here the reduced dataset from the above stage is taken as the input for the classification. The Artificial Neural Network (Back Propagation) is utilized for the classification of features into three categories i) Normal, ii) Abnormal and iii) Unknown.

Stage 3:

Rule Structure Generation

In this stage, the direction for identifying the unknown attacks' rule structures is generated using Association Rule Mining algorithm with the Abnormal and Unknown dataset as input.

Stage 4:

Signature Updation

This stage is used to create the unknown attacks' rule structures which can be efficient by using Pattern Matching algorithm. Aho-Corasick pattern matching algorithm is deployed to probe and examine for the pattern in the HIDS. Using this algorithm, the signature for the unknown attacks is updated.

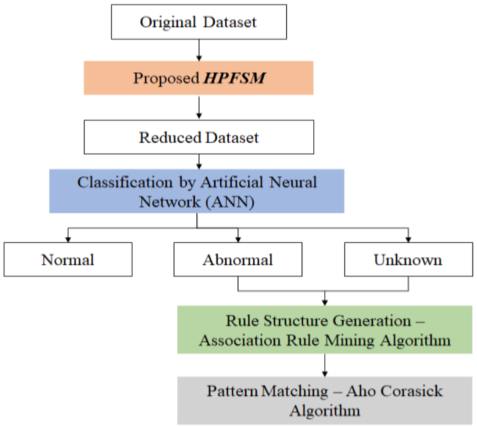
Original Dataset
Proposed ***HPFSM***
Reduced Dataset
Classification by Artificial Neural Network (ANN)
Normal
Abnormal
Unknown
Rule Structure Generation – Association Rule Mining Algorithm
Pattern Matching – Aho Corasick Algorithm

Fig. 1 IDSFS – Framework

A. Proposed Hybrid High Pertinent Feature Selection Method

A Novel Feature Selection mechanism (HPFSM) is used here. This phase was published in [13]. The proposed algorithm acts in connection with Gain Ratio (GR) and Chi-Square (CS). The relevancy among features (RF_m) is considered using Gain Ratio and Chi-Square which contributes information energy values in both the approaches. Cumulative information energy (RE_m) of (RF_m) is calculated by conjoining the information energy standards attained from the above-said methods. Then the max_value and its Feature_max_index are attained from the RE_m and settled in order. Subsequently, the merit value which is a relevancy between the attributes for the recently arrived structures is measured. This merit valued features are taken as input to the next phase-classification method. Hence, the result is a new characteristic set with new rating.

B. Artificial Neural Network Classification

Artificial Neural Network (ANN) [14] is used as eminent administered learning of neural network architecture known as Multi-Layer-Perceptron (MLP) with Back-Propagation (BP) gradient-descent. In order to custom a feed-forward multi-layer in MLP, the pool of non-linear neurons is associated to one another. As a result, this method is branded to be suitable for forecast and classification issues. On the whole, the training of the MLP initiated from a small number of neurons, and with only one hidden layer, which processes the error ratio of the trained BP on holdout samples, gradually aggregate the number of neurons at the hidden layer in which the performance of the trained phase on holdout samples is arisen due to the tricky of overtraining. In consequence, the best number of neurons for the hidden layer of the ANN is found. The network training is ruined with the Least-Square Error (E) between the desired input y_i and actual output d_i is less than E_{max}. The assumed value for $E_{max}= 1*10^{-7}$

$$E = \frac{1}{2p}\sum_{P=1}^{p}\sum_{i=1}^{m}(y_i - d_i)^2$$

where p is the total number of training patterns, and m total number of output nodes:

$$d_i = \begin{bmatrix} 1 & \mathit{If\ the\ training\ pattern} \in i^{th}\ \mathit{cluster} \\ -1 & \mathit{otherwise} \end{bmatrix}$$

For all experiments, the learning rate α is fixed to $1 * 10^{-7}$. So in this work, the input layer is composed of 16 neurons, the hidden layer composed of 13 neurons, the output layer gives 3 classes of output as Normal, Abnormal and Unknown.

C. Pattern Generation – Association Rule Mining Algorithm

Association Rule Mining (ARM) [15] is a method to regulate the manageable association rules for predictabilities among the items in comprehensive swapping information recorded. Let $I=I_1, I_2, \ldots I_m$ be a set of m targeted attributes and T be a transaction that contains a group of objects such that T→I. D is a database with exclusive transaction files. An association rule is a outcome of type *X*→*Y* where *X* and *Y* are attributes and $X \cap Y = \emptyset$. X is known as the antecedent event and Y is known as the consequent. So, the two important principles for association rule mining are *support* (S) and *confidence* (C), which designates how often items are in the database and how many times the item sets are presented, correspondingly. The succeeding includes some key classifications in ARM.

Definition 1: Given a collection of n transactions $T= \{t_1, \ldots t_n\}$ and m items $I = \{i_1,\ldots i_m\}$, an association rule is expressed in the form:

$$X\ (Antecedent) \rightarrow Y\ (Consequent) \qquad (1)$$

where $X,Y \subseteq I, X \cap Y = \Phi$, the left hand and right hand side rules are the antecedents and the consequents respectively.

Definition 2: Support(X) describes the proportion of transactions in T including X.

$$Support\ (X) = \frac{Number\ of\ Transaction\ Containing\ X}{Total\ Number\ of\ Transactions}, X \in T \qquad (2)$$

Definition 3: If $Support\ (S) \geq Min\_support$ then S is known as frequent item set where $Min_support$is a threshold value described by users.

Definition 4: Transactions Count is $N = |T|$

Definition 5: Largest transaction length is *E=Max (|ti|).*

Definition 6: The rule confidence is the proportion of transactions in T including item set X which also include item set Y. Rules with both Support (X→Y) ≥ Min_Support and Confidence (X→Y) ≥ Min_Confidence are called strong rules. These thresholds values are described through customers.

$$Confidence\ (X \rightarrow Y) = \frac{Support\ (X \cup Y)}{|Support\ (x)|} \qquad (3)$$

D. Pattern Matching – Aho-Corasick Algorithm

There are many tactics to identify patterns that comprise by finite automata. The Aho-Corasick (AC) algorithm [16]-[18] is one such typical algorithm. The idea is that a finite automaton is erected using the set of keywords in the course of the pre-computation phase of the algorithm and the corresponding encompasses the automaton scanning the input text string sense every character in y accurately once and taking persistent time for each read of a character. Primarily, it is crucial to cognize finite automata theory to apprehend the AC algorithm's depiction. The scheme which is practiced for the AC automaton is a 7-tuple (Q, q_0, A, Σ, g, f, o), where

1. Q is a finite set of states,
2. $q_0 \in$ Q is the start (initial) state,
3. A ⊆ Q and is the set of accepting states,
4. Σ is the input alphabet accepted,
5. g is a function from Q x § into Q, called the good (or goto) transition function,
6. f is a function from Q into Q, called the fail (or failure) transition function, and
7. is a function from Q into Q, called the output function.

If the automation is in a state q and reads input character (byte) a, it moves (transitions) to state g(q, a) if defined otherwise it moves to state f (q). Similarly if the automaton is in a state q, and q belongs to the set A then q is said to be an acquiescent state. Function o, the output function, returns whether or not any state q ∊A.

Aho and Corasick’s unique algorithm employs a function called output to test this and additionally profits the keyword matched at the compliant state. The AC algorithm’s automaton is such that a transition into an accommodating state specifies a match of one or more keywords. The AC algorithm deploys an enhancement of a tree to pile the set of keywords in a string matching distinctive mechanism.

E. Step by Step Procedure for Aho-Corasick Multiple-Keyword Matching Algorithm

```
Step 1: Input
        y ⟵array of n bytes representing the text input
        n ⟵integer representing the text length
        q0⟵initial state
Step 2: State ⟵q0
Step 3: Matching for i= 1 ⇢ n do
Step 4: g(state, y[i]) is undefined then while g(state, y[i]) = fail do
Step 5: Use the failure function state⟵f(state)
Step 6: end while
Step 7: state ⟵g(state, y[i])
Step 8: if o(state) ≠ ∅ then
Step 9: This an accepting state, i.e. state∊A. Output i.
Step 10: end if
Step 11: end for
```

IV. RESULTS AND DISCUSSIONS

The proposed framework IDSFS gives better result in terms of performance metrics like accuracy, FPR and Precision on classification by Artificial Neural Network.

It reduces number of features by HPFSM feature selection. It produces less number of rules, taking minimum running time and memory utilisation when using Association Rule Mining algorithm and lesser running time for pattern matching algorithm-Aho-Corosick.

A. Feature Selection by HPFSM

The proposed algorithm HPFSM contributes the smallest number of features than in other two techniques, Gain Ratio and Chi-Squared Feature Selection.

The total features in the original DARPA dataset 40 is reduced to 26 in GR, 20 in CS and HPFSM achieves lesser features of 16. It is presented in Table I.

TABLE I NUMBER OF FEATURES SELECTED FROM GR, CS AND HPFSM

Feature Selection Method	Features Selected
Gain Ratio	26
Chi-Square	20
HPFSM	16

B. Performance Evaluation in ANN

Based on the number of features itself, it cannot be said that the proposed HPFSM is better than the existing feature selections. The following evaluation metrics like Classification Accuracy, False Positive Rate (FPR) and Precision are considered for analysing the performance of the proposed HPFSM with other existing feature selection methods. Table II gives the performance metrics values while using features of original dataset, GR, CS and HPFSM by using ANN as the classifier for the classification of nodes in the network. The Fig. 2 to Fig. 4 depict the graphical representation of the Accuracy, Precision and False Positive Rate respectively.

TABLE II PERFORMANCE EVALUATION OF FEATURES OF ORIGINAL DATA SET, GR, CS AND HPFSM IN ANN CLASSIFICATION

Performance Metrics	Original Data set	GR	CS	HPFSM
Accuracy(%)	71.92	77.49	75.48	86.2
FPR	0.37	0.35	0.27	0.24
Precision	0.68	0.7	0.76	0.83

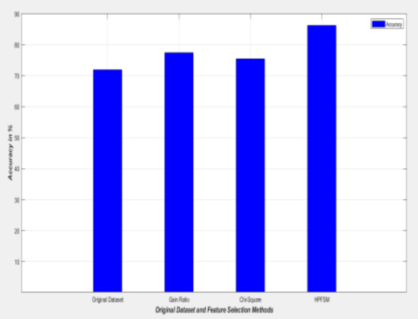

Fig. 2 Accuracy – For Features of original data set, GR, CS and HPFSM in ANN

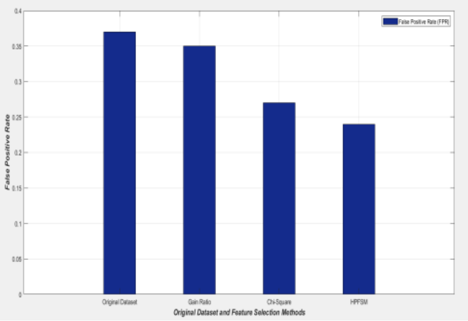

Fig. 3 False Positive Rate – For Features of original data set, GR, CS and HPFSM in ANN

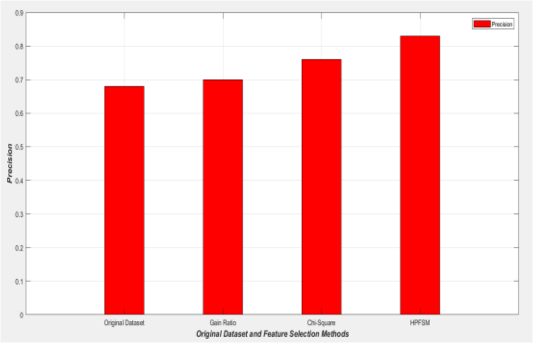

Fig.4 Precision – For Features of original data set, GR, CS and HPFSM in ANN

IV. EFFECT OF HPFSM IN RULE GENERATION BY ARM

A. Framing Number of rules: Table III depicts the number of rules generated by Association Rule Mining algorithm for given original dataset and reduced dataset obtained from proposed HPFSM with various Support and Confidence value of ARM. The same is graphically represented in Fig.5.

TABLE III NUMBER OF RULES GENERATED FOR ORIGINAL DATASET AND HPFSM DATASET BY ARM

Support and Confidence Value	Number of Rules Generated by ARM	
	Original Dataset	HPFSM Dataset
1.0 and 1.0	458	308
0.9 and 0.9	785	527
0.8 and 0.8	899	568
0.7 and 0.7	1021	632
0.6 and 0.6	1148	717
0.5 and 0.5	1259	798
0.4 and 0.4	1388	841
0.3 and 0.3	1465	892
0.2 and 0.2	1551	936
0.1 and 0.1	1675	978

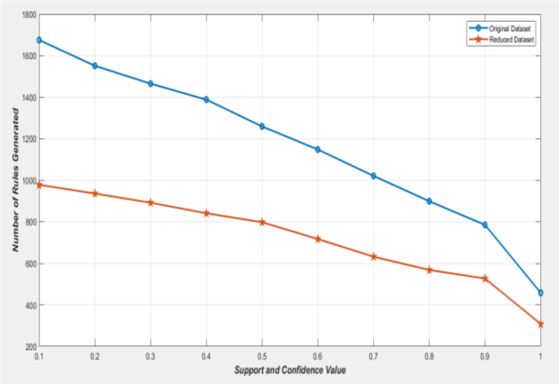

Fig.5 Number of Rules generated for Original dataset and HPFSM dataset by ARM

B. Running time of ARM

The computational time for frequent item set generation measures the amount of time taken for generating the frequent item sets with respect to given Support and Confidence values. It is measured in terms of milliseconds (ms) and mathematically formulated as follows [15],

$$RT = n \times T(n)$$

where *RT* is the running time, *n* represents the number of frequent item sets generated, and *T(n)* represented time taken for frequent item set generations. When the running time for frequent item set generation is low, the method is said to be more efficient.

Table IV and Fig. 6 depict the total execution time (in milliseconds) by Association Rule Mining for original dataset and reduced dataset. The total running time of ARM for original dataset has increased than the reduced dataset.

TABLE IV TOTAL EXECUTION TIME (IN MS) OF ARM FOR ORIGINAL DATASET AND REDUCED DATASET

Support and Confidence Value	Execution Time of ARM in ms	
	Original Dataset	Reduced Dataset
1.0 and 1.0	101	78
0.9 and 0.9	212	151
0.8 and 0.8	358	185
0.7 and 0.7	386	174
0.6 and 0.6	405	216
0.5 and 0.5	418	232
0.4 and 0.4	462	269
0.3 and 0.3	592	312
0.2 and 0.2	627	323
0.1 and 0.1	648	341

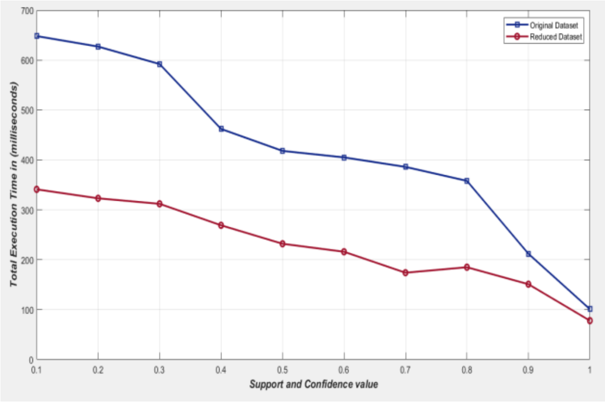

Fig.6 Total Execution time (in ms) of ARM for original dataset and reduced dataset

C. Memory Utilisation by ARM

Table V and Fig. 7 depict the total memory consumption in Mega Bytes (MB) by ARM for original dataset and reduced dataset with various support and confidence value. From this, it is clear that the reduced dataset with ARM consumes less memory than the original dataset.

TABLE V TOTAL MEMORY CONSUMPTION (IN MB) OF ARM FOR ORIGINAL DATASET AND REDUCED DATASET

Support and Confidence Value	**Total Memory Consumption in MB**	
	Original Dataset	**Reduced Dataset**
1.0 and 1.0	12.8	9.5
0.9 and 0.9	23.9	17.9
0.8 and 0.8	31.6	19.4
0.7 and 0.7	37.8	21.1
0.6 and 0.6	42.4	29.5
0.5 and 0.5	40.3	27.8
0.4 and 0.4	51.2	33.1
0.3 and 0.3	54.5	31.2
0.2 and 0.2	57.8	34.1
0.1 and 0.1	61.1	45.0

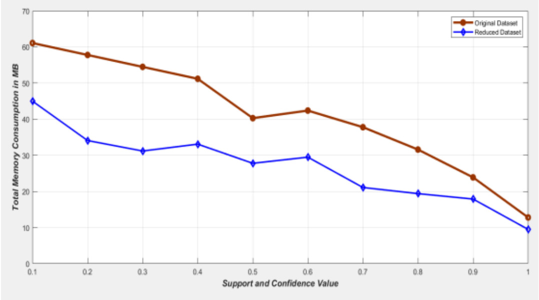

Fig. 7 Graphical representation of the total memory consumption in MegaBytes for original dataset and reduced dataset by using ARM

D. Running time of Aho-Corasick

The performance analysis of the Aho-Corasick Pattern Matching algorithm is analyzed by using running time of the algorithm with input size of the pattern.

TABLE VI RUNNING TIME (IN SECONDS) OF AHO-CORASICK PATTERN MATCHING ALGORITHM WITH ORIGINAL DATASET AND REDUCED DATASET

Input Size	**Running time in seconds**	
	Original Dataset	**Reduced Dataset**
1000	48	14
2000	71	44
3000	98	67
4000	142	90
5000	163	109
6000	185	119
7000	203	125
8000	262	140
9000	298	156

Table VI and Figure 8 depict the running time of the pattern matching algorithm for original dataset and reduced dataset. From this it is clear that the running time (in seconds) of Aho-Corasick pattern matching algorithm requires less time for reduced dataset than the original dataset.

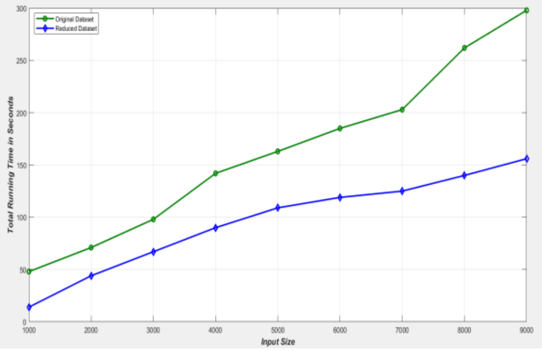

Fig. 8 Graphical representation of the Running time (in seconds) of Aho-Corasick pattern matching algorithm with original dataset and reduced dataset

V. CONCLUSION

From the results obtained, it is clear that the framework (IDSFS) with proposed HPFSM gives better result than using original dataset without using pre-processing step. The detection accuracy and Precision have increased and the False Positive Rate is reduced when the framework utilizes the proposed HPFSM method in the pre-processing stage and Artificial Neural Network as the classifier. The smaller number of rules generated with the pre-processed method than using original dataset. The total running and memory consumption are minimum for the rule generation with proposed HPFSM than using original dataset. The pattern matching algorithm requires less running time with proposed HPFSM than using original dataset. Through this article, it is concluded that the feature selection method-HPFSM plays an important role in designing efficient Intrusion Detection System.

REFERENCES

[1] Sen, Biswaraj, *et al*., "A Trust-Based Intrusion Detection System for Mitigating Blackhole Attacks in MANET", *Advanced Computational and Communication Paradigms*, Springer, Singapore, Vol. 706, pp. 765-775, 2018.

[2] Min, Hong, *et al*., "Pattern Matching Based Sensor Identification Layer for an Android Platform", *Wireless Communications and Mobile Computing*, Vol. 2018, Oct 2018.

[3] Park, Hasil, *et al*., "Hybrid Sensor Network-Based Indoor Surveillance System for Intrusion Detection", *Symmetry*, Vol. 10, No. 6, May 2018.

[4] Moustafa, Nour, Gideon Creech, and Jill Slay, "Anomaly Detection System Using Beta Mixture Models and Outlier Detection", *Progress in Computing, Analytics and Networking*, Springer, Singapore, Vol. 710, pp. 125-135, April 2018.

[5] Deshpande, Prachi, *et al*., "HIDS: A host based intrusion detection system for cloud computing environment", *International Journal of System Assurance Engineering and Management*., Vol. 9, No. 3, pp. 567-576, June 2018.

[6] Kuo, Cheng-Chung, *et al*., "Design and Implementation of a Host-Based Intrusion Detection System for Linux-Based Web Server",

International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Springer, Cham, Vol. 110, Nov. 2018.

[7] Jianglong Song, Wentao Zhao, Qiang Liu and Xin Wang, "Hybrid Feature Selection for Supporting Light Weight Intrusion Detection Systems", IOP Conference Series, *Journal of Physics*, Conference Series: Vol. 887, pp. 1-7, Aug 2017.

[8] M.S. Irfan Ahmed, A.M. Riyad, R.L. Raheemaa Khan, K. Mohamed Jamshad, E. Shamsudeen, "Information based feature selection for intrusion detection systems", *International Journal of Scientific & Engineering Research*., Vol. 8, No. 7, pp. 2362-2366, July 2017.

[9] Li, Longjie, *et al., "*Towards Effective Network Intrusion Detection: A Hybrid Model Integrating Gini Index and GBDT with PSO," *Journal* of *Sensors*, Vol. 20, Mar 2018.

[10] Raman, M.R. Gauthama, *et al*., "A hypergraph and arithmetic residue-based probabilistic neural network for classification in intrusion detection systems", *Neural Networks.,* Vol. 92, pp. 89-97, August 2017.

[11] Yu Wang, *et al.*, "A fog-based privacy-preserving approach for distributed signature-based intrusion detection", *Journal of Parallel and Distributed Computing*., Vol. 122, pp. 26-35, Dec 2018.

[12] Cohen, Yehonatan, Danny Hendler and Amir Rubin, "Detection of malicious webmail attachments based on propagation patterns", *Knowledge-Based Systems*, Vol. 141, pp. 67-79, February 2018.

[13] S. Latha and S.J. Prakash, "HPFSM-A high pertinent feature selection mechanism for intrusion detection system", *International Journal of Pure and Applied Mathematics*., Vol. 118, No. 9, pp. 77-83, 2018.

[14] Shah, Bhavin, and Bhushan H. Trivedi, "Artificial neural network-based intrusion detection system: A survey", *International Journal of Computer Applications,* Vol. 39, No. 6, pp. 13-18, Feb 2012.

[15] M. Sathya and K. Thangadurai, "Association Rule Generation Using E-ACO Algorithm", *International Journal of Control Theory and Applications,* Vol. 27, No. 9, pp. 513-521, 2016.

[16] Shim, Kyu-Seok, *et al*., "Effective behavior signature extraction method using sequence pattern algorithm for traffic identification", *International Journal of Network Management*., Vol. 28, No. 2, pp. 1-7, August 2017.

[17] Santosh Kumar Sahu, "A Detail Analysis on Intrusion Detection Datasets", *IEEE Internationals Advance Computing Conference (IACC)*, pp. 1348-1353, Feb. 2014.

[18] ZibusisoDewa and Leandros A. Maglaras, "Data Mining and Intrusion Detection Systems", *(IJACSA) International Journal of Advanced Computer Science and Applications*, Vol. 7, No. 1, pp. 62-71, January 2016.