An Efficient and Enhanced Network Intrusion Detection and Prevention system

R Radhika*1, Sreenath Reddy B*2, U Mohan Dhath *3,

#1 Assistant Professor, Dept Of CSE, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram
#2 Scholar, Dept Of CSE, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram
#3 Scholar, Dept Of CSE, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram

Abstract

This paper presents a new approach for network intrusion detection based on concise specifications that characterize normal and abnormal network packet sequences. Our specification is to develop robust network intrusion detection by enforcing a strict type discipline via a combination of static and dynamic type checking. Unlike most previous approaches in network intrusion detection, our approach can easily support new network protocols as information relating to the protocols are not hard-coded into the system. Instead, we simply add suitable type definitions in the specifications and define intrusion patterns on these types. We compile these specifications into a high performance network intrusion detection system. Important components of our approach include efficient algorithms for pattern matching and information aggregation on sequences of network packets. In particular, our techniques ensure that the matching time is insensitive to the number of patterns characterizing different network intrusions, and that the aggregation operations typically take constant time per packet. However, the literature still lacks thorough analysis and evaluation on data fusion techniques in the field of intrusion detection. Therefore, it is necessary to conduct a comprehensive review on them. In this article, we focus on DF techniques for network intrusion detection and propose a specific definition to describe it. We review the recent advances of DF techniques and propose a series of criteria to compare their performance. Finally, based on the results of the literature review, a number of open issues and future research directions are proposed at the end of this work.

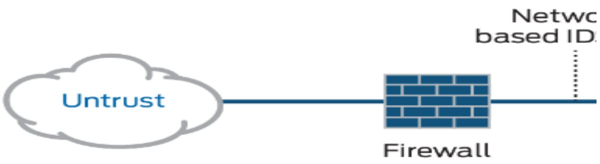
Keywords

Pattern Match Intrusion Detection System Pattern Group Pattern Index Destination Port

Introduction:

Intrusion detection is the process of monitoring the events occurring in your network and analyzing them for signs of possible incidents, violations, or imminent threats to your security policies. Intrusion prevention is the process of performing intrusion detection and then stopping the detected incidents[1]. A typical business network has several access points to other networks, both public and private. The challenge is maintaining the security of these networks while keeping them open to their customers. Currently, attacks are so sophisticated that they can thwart the best security systems, especially those that still operate under the assumption that networks can be secured by encryption or firewalls. Unfortunately, those technologies alone are not sufficient to counter today’s attacks.

Intrusion detection systems (IDS) and intrusion prevention systems (IPS) constantly watch your network, identifying possible incidents and logging information about them, stopping the incidents, and reporting them to security administrators. In addition, some networks use IDS/IPS for identifying problems with security policies and deterring individuals from violating security policies [2-5]. IDS/IPS have become a necessary addition to the security infrastructure of most organizations, precisely because they can stop attackers while they are gathering information about your network.



How Does IDS Work?

The three IDS detection methodologies are typically used to detect incidents.

Signature-Based Detection compares signatures against observed events to identify possible incidents. This is the simplest detection method because it compares only the current unit of activity (such as a packet or a log entry, to a list of signatures) using string comparison operations.

Anomaly-Based Detection compares definitions of what is considered normal activity with observed events in order to identify significant deviations. This detection method can be very effective at spotting previously unknown threats [6].

Stateful Protocol Analysis compares predetermined profiles of generally accepted definitions for benign protocol activity for each protocol state against observed events in order to identify deviations.

Network Intrusion Detection System (NIDS) is a new generation of network security equipment following the traditional security measures such as firewall and data encryption [1], which has been rapidly developed in recent years. It successfully resists many attacks and malicious actions and is called the second line of defense in the Internet[9]. However, in the current big data era, the large amount of traffic data makes NIDS face critical challenges. First, large amounts of high-dimensional data increase processing complexity and need huge computing and storage resources. Second, many redundant and unrelated data could adversely affect network security detection. Third, some new attacks are difficult to detect due to big data process and analytics. Besides, the inherent weakness of NIDSs, such as high false positives (FP) and high false negatives (FN), raises urgent requests on effective solutions. Data Fusion (DF), as a promising technology of big data, has been applied into the domain of network intrusion detection to overcome the above-mentioned challenges in recent years.

The concept of DF originated from the US Air Force project; the US Department of Defense first proposed a Joint Directors of Laboratories (JDL) DF model based on national defense monitoring needs in 1987 [2]. Subsequently, DF was gradually studied and applied in other fields, such as automatic control, image recognition, target detection, and cyber security, and many scholars have proposed definition of DF based on their own studies and researches [3]. In order to clearly show the role of DF technology in network intrusion detection, an expression of DF in the field of NIDS is presented in this article.

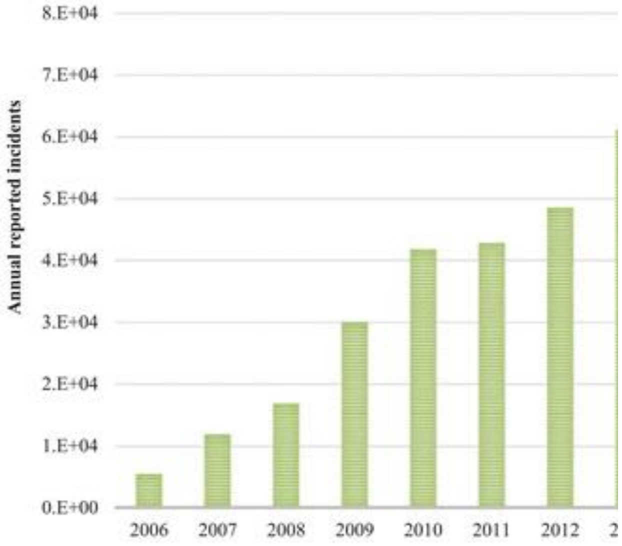
In general, DF can be applied into three layers according to where fusions are needed, namely, data layer, feature layer, and decision layer. The data layer is the lowest system layer, playing the role of processing and integrating raw network data; the feature layer is the middle layer, fusing and reducing features of the preprocessed data; the decision layer is the highest layer, fusing and combining the inferences or decisions of various processing units. In the field of NIDS, most researches of data fusion only focus on the feature layer and the decision layer. It is because the network data they need to fuse comes from public datasets that have already been fused at the data layer. The use of DF technology at the feature level can greatly reduce the size of data processing, thereby enhancing the efficiency of NIDSs. Besides, useful and refined data generated by feature fusion can support decision-making and further improve the robustness and accuracy of the system. As for using of DF technology at the decision level, the decision fusion center fuses the decisions of multiple local detectors to obtain more accurate and reliable identifications of network behaviors.

Currently, a lot of research work has been carried out on DF for intrusion detection in order to improve the performance of NIDS. However, we found that the open datasets, the number of experimental data samples, and the fusion techniques used in many literatures are diverse. It is difficult to understand and analyze the strengths and weaknesses of different fusion techniques. Thus, it becomes essential to specify uniform criteria to evaluate them in view of a large number of references and give performance statistics of the current literature. This work is meaningful because it can make it easier for researchers and practitioners to understand the characteristics of the current DF techniques and methods.

In this article, we provide a thorough review on DF techniques in NIDS. We first describe DF for NIDS by representing the process and role of fusion for motivating this research work. We review existing DF techniques used in intrusion detection and propose evaluation criteria to analyze and compare the characteristics and performance of different fusion techniques. Besides, we simply analyze different open network datasets that can be used for testing intrusion detection techniques.

Related work

Intrusion detection is an active field of research for about more than three decades. The interest in network intrusion detection has increased among the researchers along with the needs of security. Using automated tools and exploit scripts for the attacks, experienced intruders have performed large numbers of attacks 1980s in order to affect sites on the Internet. However, anybody can intrude using different tools. The given figure illustrates the statistics of federal agencies in the United States, which shows that the number of cyber security incident reports increased dramatically from 2006 to 2015. However, due to some changes in the federal guidelines, it decreased by 60% in 20016.



An anomaly-based IDS approach is proposed to incorporate between a multivariate statistical process control (MSPC) which is called Hotelling's T^2 and radio frequency fingerprinting (RFF) in order to detect the attack. Depending on the generated signal, RFF is responsible for distinctively identifying a transceiver based on the transceiver print. We can achieve through wireless device MAC (media access control) address. However, still there is an issue because MAC address could be attacked, the transceiver prints would not match the profile with the claimed MAC address. Wormhole Geographic Distributed Detection (WGDD) algorithm is proposed for distributed wormhole detection. The main task of this algorithm is to find a disorder of network produced by a wormhole. The passive nature of this kind of attack, a hop counting method, is used in the algorithm for detecting wormhole attacks. The local maps are reconstructed in every node. The algorithm can detect the abnormal behavior produced by wormhole attacks using a feature named diameter. A key benefit of applying the algorithm is that it can detect the position of wormhole that can help in the future to secure against these attacks.

Payload-based anomaly (PAYL) detector builds a profile for the normal application payload of the network traffic in the training phase and uses that profile later for comparing detected intrusions. In the training phase, the profile of the application payload is built automatically in an unsupervised way. The profile consists of the centroids and the standard deviation of the byte frequency distribution of the network traffic payload for the flows based on the network hosts and ports. The byte frequency is computed by calculating the number of existences of every byte in the traffic payload and then dividing it by the total number of bytes. For each different payload length, a different byte frequency distribution model is calculated. To detect intrusions, the byte frequency distribution of the network traffic payload is calculated. After that, the distance between the byte frequency distribution of the network payload and the profile is calculated based on the centroids and the standard deviation. If the distance is larger than a specified threshold, then an alarm is activated. Moreover, incremental learning is supported by PAYL, where the profile can be updated using new data without the need to recreate the whole profile again. As a result of the dependency on the payload length to build the models of the profile, a huge number of models are required. Therefore, to satisfy this requirement, the clustering technique is used to reduce the number of required models.

Hierarchal Intrusion Detection (HIDE) developed as a distributed hierarchal system based on anomaly network intrusion detection system (NIDS). HIDE depends on statistical modeling, preprocessing, and classification of a neural network to detect network-based attacks. The network traffic information is observed to build the network statistical model. HIDE contains many intrusion detection agents, which are gathered in different hierarchal tiers. HIDE divides the network into zones. For each zone, a set of tier-1 agents is used to monitor the activities of the servers and the network bridges of that zone, to build the traffic statistical model, generate the monitoring reports periodically, and send the reports to an agent in tier 2. A tier-2 agent is used in each zone to receive the periodical reports of tier-1 agents of that zone, monitor and analyze the performance of the zone based on the received reports, and generate and send the report to an agent in tier 3. In addition, to receive the reports of tier-2 agents, tier-3 agents receive the reports of the tier-1 agents that are deployed in the network firewalls and routers. The network statistical model is built up by all agents participated in all different tiers to provide the neural network classifier. The neural network classifier's main objective is to decide whether the provided statistical model is normal or not.

HIDE has different components, a probe component monitors the network traffic to collect and extract a set of statistical variables based on the collected data for network traffic to reflect the network situation and generate periodical reports to the event preprocessor. Event preprocessor receives the reports generated from both the probe component and the reports of the agents in the lower tier, and construct the statistical model based on the received reports. The statistical processor compares the reports generated by the even preprocessor to the reference model and creates the stimulus vector which is provided to the neural network classifier. The neural network classifier receives the stimulus vector generated by the statistical processor, analyzes it, and classifies the network traffic whether it is normal or not. Postprocessor the neural network classifier to generate a report to the agents in the upper tier by the classifier. A neural network classifier needs time for training to learn before it can be used for detection. In the training phase, the neural network classifier is learned using learning data.

Flow-Based Statistical Aggregation Schemes (FSAS) produces 22 statistical features for every network flow. The neural network classifier receives those features extracted by FSAS. The network flow can be modeled to be classified into two modes, safe and unsafe flows. This modeling is basically built in the training phase as a set of probability density functions of the 22 features values. The model contains two profiles, normal and attack profiles. In addition, FSAS consists of two main processes, which are a feature generator and a flow-based detector. An event preprocessor collects the network traffic from hosts or networks. Flow management module decides if each received packet is a part of existing network flow, or if it is the first packet in a new network flow. Afterward, it updates the records of the corresponding flow based on the received packet. The probe receives the information from the network flow coming from the flow management module and then extracting a set of statistical components to introduce the network status. Neural Network Classifier classified every network flow based on its score vector to be a safe or malicious flow. Feature analyzer identifies the type of attack based on the network's major behavior changes.

KMNP (k-means clustering based intrusion detection protocol) detects intrusions efficiently using a clustering technique and a classification technique in two phases. In the first phase, KMNP uses the K-means clustering technique, the second phase uses the Naïve Bayes classifier. K-means technique is used to cluster and classify data into malicious and non-malicious groups in the first phase. In the second phase, Naïve Bayes classifier classifies data into its potential group. In addition, KMNP, K-means technique clusters data into three groups. The first group contains all the attack data such as a probe, R2L, and U2R. The second group contains the DoS attacks data. The third group contains normal network traffic data. K-means technique grouped data into K clusters/groups, where the centroid (mean value) of each cluster is considered as the seed point of that cluster. After that, based on the value of the squared distance between the data input and the centroids of the clusters, each data input is assigned to the nearest cluster. In the second phase, the Naïve Bayes technique is used which is considered as popular learning techniques. Naïve Bayes technique analyzes the relationship between the independent variable and the dependent variable to identify a conditional probability for that relationship. Therefore, the Naïve Bayes technique classifies the network data into five classes: normal, DoS, probe, R2L, and U2R.

Minnesota Intrusion Detection System (MINDS) is a data mining technique for intrusion detection. Each network connection is assigned with a score based on the probability of that connection to be an intrusion. MINDS detects the intrusions by using the packet's header information to construct the flow information. Flow information consists of IP addresses and ports of the source and destination, protocol, flags, number of bytes and number of packets of that flow. Based on time-window derived features, they are generated for the network flows with similar characteristics in the last “T” seconds. The local outlier factor (LOF) of the network flow is calculated based on the flow information and extracted features. LOF measures the degree of a network flow of being an outlier for its neighbors. To calculate the LOF, the density of the neighborhood is calculated. LOF is then computed as the average of the ratios of the density of the network flow and the density of its neighbors.

Proposed System

This section introduces the data fusion techniques, mainly focusing on feature fusion and decision fusion. We classify the fusion techniques shown in Figure 2 and describe the commonly used fusion techniques. As mentioned above, DF techniques in NIDS can be classified into the data layer fusion, the feature layer fusion, and the decision layer fusion. To the best of our knowledge, the majority of researches on NIDS are based on open datasets, which leads to the result that the data level fusion is omitted in the related literatures. Therefore, we mainly review the DF techniques at the feature layer and the decision layer.

There are two main categories for feature fusion in NIDS: filters and wrappers [14]. The filters are applied through statistical methods, information theory based methods, or searching techniques [15], such as Principal Component Analysis (PCA), Latent Dirichlet Allocation (LDA), Independent Component Correlation Algorithm (ICA), and Correlation-Based Feature Selection (CFS). The wrapper uses a machine learning algorithm to evaluate and fuse features to identify the best subset representing the original dataset. The wrapper is based on two parts: feature search and evaluation algorithms. The wrapper approach is generally considered to generate better feature subsets but costs more computing and storage resources than the filter [27]. The filter and the wrapper are two complementary modes, which can be combined. A hybrid method is usually composed of two stages. First, the filter method is used to eliminate most of the useless or unimportant features, leaving only few important ones, which can effectively reduce the size of data processing. In the second stage, the remaining few features representing the original data are used as input parameters to send into the wrapper to further optimize the selection of important features.

The decision fusion methods are divided into two classes: winner-take-all and weighted sum, by considering how to combine decisions from basic classifiers [23]. Majority vote, weighted majority vote, Naïve-Bayes, RF (Random Forest), Adaboost, and D-S evidence theories are classified as the type of winner-take-all because they all have measured values for each basic classifier and the final decision depends on the classifier with the highest measured value. In case of the weighted sum, the weight of each basic classifier depends on its own capabilities. The weights of basic classifiers are calculated, and then their outputs with the weights are added to give a final decision. The method of weighted sum mainly includes average and neural network.

Bayesian estimation is applied to DF for a long time. It is an excellent method if prior probability is known. In order to obtain the most accurate and comprehensive information, this method first analyzes the compatibility of various sensors, removes false information with low confidence, and makes the Bayesian estimate of useful information under the assumption that the corresponding prior probabilities are known. The advantages of Bayesian approach include explicit uncertainty characterization and fast and efficient computation. Moreover, Bayesian networks offer good generalization with limited training data and easy maintenance when adding new features or new training data [23]. The disadvantage of Bayesian estimation is that it cannot distinguish unaware and uncertain information, and it can only handle the related events. In particular, it is difficult to know the prior probabilities in practical applications. When the hypothetical prior probabilities are contradictory to reality, the results of the inference will be undesirable and will become quite complicated when dealing with multiple hypotheses and multiple conditions. In fact, the Bayesian inference methods are now rarely applied in DF because of these defects.

Neural Network

Neural Network (NN) is a supervised learning method that consists of input neurons, output neurons, and hidden neurons. In order to represent the relationship between the input neuron and the output neuron, the neural network needs a large amount of labelled data to train and obtain an accurate model. NN has the characteristics of self-learning, self-adaptation, self-organization, and fault-tolerant, which enable it to solve complex nonlinear problems. Furthermore, the advantage of NN is that it can automatically adjust the connection weights without any domain-specific knowledge, while other methods use preselected weights to combine outputs [3]. Therefore, its strong capabilities can be well adapted to the requirements of multisource DF in NIDS. In network intrusion detection, the classification results of multiple detectors are used as input neurons, and the output neurons are integrated classification results. The output of the neural network is used as feedback to adjust the training parameters. With the improved parameters, the detectors can be fused to produce an improved resultant output. The main drawback of NN is the lack of valid criteria for creating, selecting, and combining the results of the base classifiers. For example, one may use a Multilayer Perceptron (MLP) or a radial basis function to find fusion weights with different structure.

Evaluation Criteria of DF Techniques

The application of DF techniques in intrusion detection has received particular attention in the field of network security. Many studies on DF have been conducted to improve the performance of NIDS. However, DF in NIDS still faces many serious challenges, such as how to reduce the complexity of massive data, how to ensure data security, and how to overcome the complexity and improve the efficiency of the fusion. Therefore, in order to facilitate the analysis and comparison of different fusion techniques, we propose a number of criteria for evaluating the performance of fusion techniques in NIDS based on the traditional criteria of IDS performance. Herein, we introduce specific evaluation criteria. Since most of the experiments for NIDS performance testing are based on a few public datasets, we firstly introduce the commonly used datasets for intrusion detection.

Conclusion

In this article, we categorically presented a detailed review on the feature fusion techniques and the decision fusion techniques used in NIDSs. A specific description of DF in the field of intrusion detection was presented in order to motivate this work. Based on the literature stud, we proposed the evaluation criteria of data fusion techniques in terms of NIDS. The performance of different data fusion techniques is measured using the proposed criteria. We found that, in the feature fusion, in addition to some excellent fusion techniques, such as SVM and MIFS, the improved types of fusion techniques and hybrid fusion techniques are generally efficient and valid. For the decision fusion techniques, D-S Evidence Theory, NN, RF, and Adaboost can combine multiple decisions more precisely than other methods regarding the studies based on KDD dataset series. In addition, we found many effective classification algorithms in NIDS, namely, RF, C4.5, NN, and SVM, as well as their variants. Unfortunately, the current fusion techniques normally did not consider the security and the scalability of DF.

DF has been regarded as one of the most important technologies in improving the performance of the NIDSs. The use of DF can well alleviate the defects of network intrusion detection and improve the comprehensive performance of NIDSs. However, there are still many deficiencies in current DF techniques. Based on our review, we pointed out the main challenges and promising future research directions in this field of research. In summary, this article provides a good reference for researchers and practitioners in the field of network intrusion detection.

References

1. D. Anderson, T. Lunt, H. Javitz, A. Tamaru, and A. Valdes, Next-generation Intrusion Detection Expert System (NIDES): A S

2. P. Porras and P. Neumann, EMERALD: Event Monitoring Enabled Responses to Anomalous Live Disturbances, National Information Systems Security Conference, 1997.

3. M. Ranum et al, Implementing A Generalized Tool For Network Monitoring, LISA, 1997.

4. R. Sekar, T. Bowen and M. Segal, On Preventing Intrusions by Process Behavior Monitoring, USENIX Intrusion Detection Workshop, 1999.

5. R. Sekar and P. Uppuluri, Synthesizing fast intrusion

detection/prevention systems from high-level specifications, USENIX Security Symposium, 1999.

6. R. Sekar and P. Uppuluri, Synthesizing Fast Intrusion Prevention/Detection Systems from High-Level Specifications, Technical Report 99-03, Department of Computer Science, Iowa State University, Ames, IA 50014

7. J. Tian, W. Zhao, R. Du, and Z. Zhang, “A New Data Fusion Model of Intrusion Detection-IDSFP,” in *Parallel and Distributed Processing and Applications*, vol. 3758 of *Lecture Notes in Computer Science,* pp. 371–382, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

8. F. E. White, “Data Fusion Lexicon,” *Defense Technical Information Center,* 1991. H. Boström, S. F. Andler, M. Brohede et al., “On the definition of information fusion as a field of research,” *Neoplasia*, vol. 13, pp. 98–107, 2007, IN1.

9. B. R. Raghunath and S. N. Mahadeo, “Network Intrusion Detection System (NIDS),” in *Proceedings of the 2008 First International Conference on Emerging Trends in Engineering and Technology*, pp. 1272–1277, Nagpur, Maharashtra, India, July 2008.

10. Y. Fu, Z. Yan, J. Cao, O. Koné, and X. Cao, “An Automata Based Intrusion Detection Method for Internet of Things,” *Mobile Information Systems*, vol. 2017, Article ID 1750637, 13 pages, 2017.

11. L. Wang and H. Xiao, “An integrated decision system for intrusion detection,” in *Proceedings of the 1st International Conference on Multimedia Information Networking and Security, MINES 2009*, pp. 417–421, chn, November 2009.

12. M. A. Aydin, A. H. Zaim, and K. G. Ceylan, “A hybrid intrusion detection system design for computer network security,” *Computers and Electrical Engineering,* vol. 35, no. 3, pp. 517–526, 2009.

13. H. Wang, X. Liu, J. Lai, and Y. Liang, “Network security situation awareness based on heterogeneous multi-sensor data fusion and neural network,” in *Proceedings of the International Multi-Symposiums on Computer and Computational Sciences*, pp. 352–359, 2007.

14. S. Mukkamala, G. Janoski, and A. Sung, “Audit data reduction for intrusion detection,” Training, 2008.

15. A. H. Sung and S. Mukkamala, “Identifying important features for intrusion detection using support vector machines and neural networks,” in *Proceedings of the International Symposium on Applications and the Internet*, pp.

209–216, IEEE, Orlando, Fla, USA, January 2003.

16. I. S. Thaseen and C. A. Kumar, “Intrusion detection model using fusion of PCA and optimized SVM,” in *Proceedings of the 2014 International Conference on Contemporary Computing and Informatics, IC3I 2014*, pp. 879–884, ind, November 2014.

17. A. Ammar, “Comparison of Feature Reduction Techniques for the Binominal Classification of Network Traffic,” *Journal of Data Analysis Information Processing*, vol. 03, pp. 11–19, 2015.

18. N. A. Biswas, F. M. Shah, W. M. Tammi, and S. Chakraborty, “FP-ANK: An improvised intrusion detection system with hybridization of neural network and K-means clustering over feature selection by PCA,” in *Proceedings of the 18th International Conference on Computer and Information Technology, ICCIT 2015*, pp. 317–322, bgd, December 2015.

19. J. Zhou, J. Wang, and Z. Qun, *The Research on Fisher-RBF Data Fusion Model of Network Security Detection*, Springer, Berlin, Heidelberg, Germany, 2012.

20. J. Zhang, M. Zulkernine, and A. Haque, “Random-forests-based network intrusion detection systems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 5, pp. 649–659, 2008.

21. C. Khammassi and S. Krichen, “A GA-LR wrapper approach for feature selection in network intrusion detection,” *Computers & Security*, vol. 70, pp. 255–277, 2017.

22. M. A. Ambusaidi, X. He, Z. Tan, P. Nanda, L. F. Lu, and U. T. Nagar, “A novel feature selection approach for intrusion detection data classification,” in *Proceedings of the IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 82–89, 2015.

23. Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, and K. Dai, “An efficient intrusion detection system based on support vector machines and gradually feature removal method,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 424–430, 2012.

24. H. T. Nguyen, S. Petrović, and K. Franke, “A comparison of feature-selection methods for intrusion detection,” in *Lecture Notes in Computer Science*, I. Kotenko and V. Skormin, Eds., vol. 6258, pp. 242–255, 2010.

25. S.-W. Lin, K.-C. Ying, C.-Y. Lee, and Z.-J. Lee, “An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection,” *Applied Soft*

Computing, vol. 12, no. 10, pp. 3285–3290, 2012.

26. I. Sumaiya Thaseen and C. Aswani Kumar, “Intrusion detection model using fusion of chi-square feature selection and multi class SVM,” *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 4, pp. 462–472, 2017.

27. Y. Xu and W.-B. Zhang, “A novel IDS model based on a Bayesian fusion approach,” in *Proceedings of the 1st International Conference on Multimedia Information Networking and Security, MINES 2009*, pp. 546–549, chn, November 2009.

28. S. Chebrolu, A. Abraham, and J. P. Thomas, “Feature deduction and ensemble design of intrusion detection systems,” *Computers & Security*, vol. 24, no. 4, pp. 295–307, 2005.

29. M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, “Building an intrusion detection system using a filter-based feature selection algorithm,” *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

30. K. S. Desale and R. Ade, “Genetic algorithm based feature selection approach for effective intrusion detection system,” in *Proceedings of the International Conference on Computer Communication and Informatics*, pp. 1–6, 2015.

31. A.-C. Enache, V. Sgarciu, and A. Petrescu--Niţă, “Intelligent feature selection method rooted in Binary Bat Algorithm for intrusion detection,” in *Proceedings of the Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics, SACI 2015*, pp. 517–521, 2015.

Authors Profile

R Radhika, working as an Assistant Professor in the department of Computer Science & Engineering in Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram.

Sreenath Reddy B, pursuing BE in the department of Computer Science & Engineering in Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram.

U Mohan Dhath, pursuing BE in the department of Computer Science & Engineering in Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Kanchipuram.