International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Decentralized Time-Window Based Real-Time Anomaly Detection Mechanism (DTRAD) in Iot

S L Sanjith, E. George Dharma Prakash Raj

Abstract*: Detecting intrusions has become a mandatory service in IoT environments. This is due to the power and resource con-strained nature of the networks. This paper presents a Decentralized Time-Window based Anomaly Detection (DTRAD) model for cost and time effective intrusion detection in IoT environments. The proposed model is composed of time win-dow based training data selection module, which enables better detection and reduced bias. Training data are selected based on their temporal significance and the bag creation process is also temporally performed such that data with similar temporal signatures are grouped into same bags. The ensemble model is created and weighted voting is performed to ena-ble better results. The data reinforcement module enables new data to be appended to the training data, hence maintaining the recency of the data. Further, the entire process is decentralized, hence enabling data processing at appropriate nodes. This keeps the size of the training data low, hence reducing the computational complexity of the model to a large extent. Experiments were performed with benchmark data and comparisons were performed with recent models. Results indicate high performance of the proposed models.

***Index Terms*: Intrusion Detection, Temporal Data Selection, Ensemble, Bagging, Decentralization.**

I. INTRODUCTION

Improved automations in the technology domain has resulted in usage of interconnected smart devices in several areas [1]. These devices form the Internet of Things (IoT). IoT has been adopted in several areas due to the ease of use and the high level of advantages and flexibility they provide to the users [2, 3]. In an IoT network, many smart devices are connected to the internet to form a closed infrastructure. Communications in this network are performed using specialized protocols based on the IPv6 protocol. The IoT model usually uses the RPL (Routing Protocol for Low Power Lossy Network) and the 6LoWPAN (IPv6 over Low Power Wireless Private Area Network) protocols for internal communication [4, 5]. These protocols have been specifically designed for power constrained devices to communicate with each other. The communication usually occurs within a closed network and a connection to the internet is sometimes made available to the devices. IoT has several applications including healthcare monitoring, smart cities, smart homes etc [6,7].

Communication of IoT devices is similar to networked systems, however several specific issues arise when developing anomaly detection systems for an IoT network. This is mainly due to the power constrained nature of the system, due to the small size and low cost [8]. The major drawback of IoT networks is that they cannot afford resource intensive security algorithms. This makes IoT based networks an easy target for intrusions. Further, these networks are usually used by common people and not security experts, hence naïve usage is expected, which makes them highly vulnerable to attacks. The IoT devices are usually connected to the internet. Lack of security in these areas also raises the issues. Lots of attacks specifically targeting IoT devices exists in this domain [9]. Hence the major challenges existing in the domain includes developing an anomaly detection model that is light weight, with low computational requirements, low memory requirements and low power requirements.

This work presents a Decentralized Time-window based Anomaly Detection (DTRAD) model to provide effective predictions with low computational requirements. Training data for the proposed model is selected based on the temporal nature of the data. Only recent transmissions are selected, leading to low but effective training instances. Further, the proposed bagging model uses tree based base learners, providing highly efficient prediction process. The decentralized nature of the prediction process and the knowledge sharing capabilities available in the model enhances the prediction process to a large extent.

II. LITERATURE REVIEW

Anomaly detection in IoT is considered a special case of network anomaly detection, due to the specific constraints involved in developing such a system. This section discusses some of the recent works in the domain of anomaly detec-tion.

A Support Vector Machine (SVM) based model for intrusion detection was proposed by Bamakan et al. [10]. This method trains the SVM models by considering the problem as both classification and regression and performs predictions. The major focus of the model is to handle the highly imbalanced and skewed nature of the data to perform effec-tive training and eventually more effective results. A similar SVM and feature selection based model was proposed by Li et al. [11].

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
S L Sanjith, Indian Institute of Management Tiruchirappalli, Tiruchirappalli, India.
E. George Dharma Prakash Raj, Bharathidasan University, Tiruchirappalli, India.

The model performs gradual feature selection using Ant Colony Optimization (ACO) algorithm and finally utilizes SVM for the actual prediction process. The RS-ISVM method proposed by Yi et al. [12] aims to use incremental SVM for the prediction process. The model uses an improved Gauss kernel called U-RBF to suppress noise. Other SVM based models include works by Kuang et al. [13], Ahmad et al. [14] and Chitrakar and Huang [15]. Other machine learn-ing models used on intrusion detection systems include, Artificial Neural Network based model [16], Decision Tree based IDS model [17], Extreme Learning based model [18] and K-Nearest Neighbor based model [19].

A specific wormhole based attack detection model for IoT was proposed by Bhosale and Sonavane [20]. The pro-posed model is composes of two major components; the distributed module and the centralized module. They distinguish between the transmission data effectively to provide enhanced predictions. a clustering based model for intrusion detection and prevention was proposed by Jarrah et al. [21]. This technique is a semi-supervised model, that operates based on multi-layered clustering. The major advantage of this model is that it has the ability to learn from partially labelled data. The performance of this model has been found to be in-par with most supervised machine learning mod-els. Further, it also exhibits detection accuracy levels comparable to even ensemble based techniques. An Artificial Neural Network (ANN) based intrusion detection system was proposed by Shenfield et al. [22]. This method applies deep packet inspection to discriminate between anomalous signatures and normal signatures. However, usage of neural net-works increases the computational complexity levels of the model to a large extent.

Energy efficiency is another major constraint when dealing with IoT based networks. The detection systems are usu-ally deployed in uniform and Gaussian deployment mode. However, they were not energy efficient and does not enable quick detection. This serves as the base for the physical intrusion detection model proposed by Halder et al. [23]. This method proposes a tailor-made Gaussian distribution based strategy for effective physical deployment of intrusion de-tection systems. The major advantage of this model is that it operates effectively even on heterogeneous networks. A time constrained system for anomaly detection was proposed by Yilmaz et al. [24]. This model operates on smart grids, enabling effective identification and prevention of attacks.

III. DECENTRALIZED TIME-WINDOW BASED REAL-TIME ANOMALY DETECTION IN IOTS

Anomaly detection in IoT is a complicated process. The challenges mostly arise due to the low power and the low processing power that are inherent to the IoT devices. This work presents a Decentralized Time-window based Anomaly Detection (DTRAD) system by considering the temporal state of the transmission data. The architecture of the proposed system is shown in figure 1.

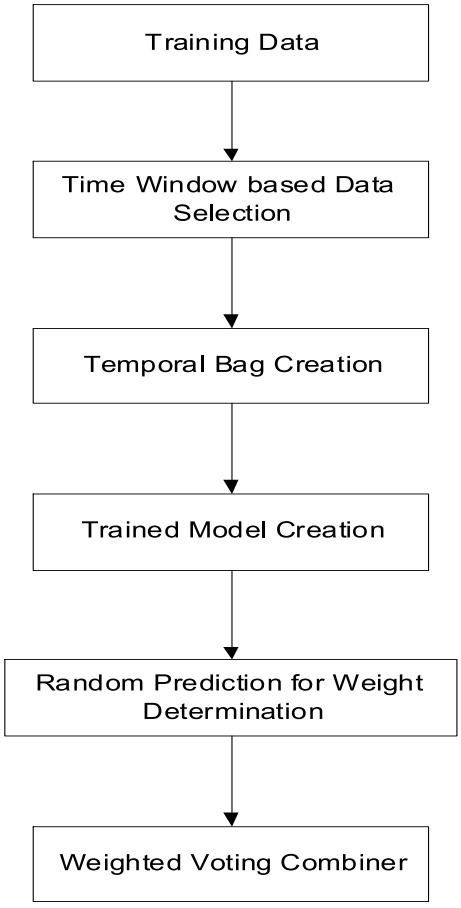

Figure (a): DTRAD Architecture

The proposed DTRAD architecture operates by selecting the training data based on its temporal significance. This eventually results in less training data for the model, hence lesser computational requirements. However, the recent nature of the data results in better predictions. Further, the architecture is decentralized, and distributed processing is performed. Processing is performed in each node and the anomalous signatures are shared. Every node processes its own data. The proposed model operates in six major phases; data preprocessing, temporal formulation of training data, bagged model creation, weight assignment, weighted voting combiner and data reinforcement.

DTRAD Algorithm

1. *Input temporal threshold*
2. *Select training data satisfying the threshold*
3. *Input number of bags to be created*
4. *For each bag to be created*
    a. *Select data from the training data such that it overlaps with the data prom previous bag*
5. *Pass each bag to the ensemble model to create trained model*

6. *Select random data from the training data set to create temporary test data (TTD) for weight analysis*
7. *Predict TTD using each trained model*
8. *Identify the accuracy of prediction of each model*
9. *Assign highest weight for the bag with higher accuracy and reduce weight levels till the last bag*
10. *Predict test data using the models*
11. *Perform weighted voting for result aggregation*
12. *Save predicted transactions in a buffer*
13. *Identify the actual status of the transaction in the buffer*
14. *Pass the transactions to the training data set*

Figure (b): DTRAD Algorithm

A. Data Preprocessing

Data preprocessing is one of the major components of a machine learning model. Data for analysis is generated by IoT devices. Hence the data is prone to inconsistencies. Missing data is a common problem when dealing with such data. Machine learning algorithms cannot operate on such data. Hence they should be corrected and prepared appropriately for the machine learning models.

Input data, being generated by devices, contains several details that represent the source node location and other additional transmission information. Most such details are only for representational purposes and are not useful for the learning models. Such data are to be eliminated. Further, the data might also contain categorical data. Such representations are not directly interpreted by the learning models, hence they should be converted to numerical formats.

Preprocessing begins by deleting attributes that are provided only for identification purposes. The next step identifies categorical data. Categorical data re usually treated by applying appropriate encoding techniques. This results in all data converted to numerical formats. Numerical data, however, will be in varied ranges depending upon the property represented by the attribute. Passing such diverse data results in bias in machine learning models. Hence the data should be converted into similar ranges. This is performed using normalization. Normalization is the process of converting data from varied formats into consistent formats. This work uses Min-Max normalization for processing. This is given by

$$x' = \left(\frac{x - \min(A)}{\max(A) - \min(A)}\right) * (D - C) + C$$

where x' is the normalized value and x is the actual value of the attribute A. C and D are the predefined boundaries [C,D] between which the data is to be scaled. Completion of these process ensures that the data is ready for machine learning.

B. Time Window based Training Data Selection

Packets generated by the IoT devices are temporal in nature. i.e. data generated in completely dependent on the time of generation. Certain time periods exhibit increased values, while certain others exhibit reduced values. Variations could be observed based on the temporal nature of the transmission. Considering transmissions from IoT, it is sufficient to analyze the recent transmissions rather than analyzing all older transmissions. Further, older transaction will not contain the recent variation level that has been added to the data. Hence tends to add unnecessary bias to the prediction process. Since it is recommended to select training data for the learning model based on the temporal nature of the data, only the most recent transactions are selected. The number of transactions selected is based on the computational level of the processing node and the variation levels existing in the transactions. This is usually decided by the domain expert.

C. Bagged Model Creation using Training Data

The training data obtained from the previous phase is used for the bag creation process. Bags are created based on the temporal significance of the data and not by random selection. The initial bags are composed of old data, followed by bags with more recent data. Data overlaps between bags are maintained. A bag contains common data overlapping with its previous and next bag. This maintains a gradual temporal distribution change, rather than keeping it abrupt. The last bag is composed of most recent data.

Each bag is used by a different machine learning model for training. This work utilizes two machine learning models, Decision Tree and Random Forest for the training process. It was observed from the previous work of the authors, that tree based models perform effectively in identifying network intrusions. This has resulted in the selection of Decision Tree and Random Forest for this work.

Decision Tree is a tree based classifier model that creates graphs to perform decision making. Branching in Decision Trees are based on entropy. Entropy is the process of identifying the information gain that occurs due to the branching process. Hence each branch in a decision tree contains a condition and the path directs to a specific flow if the condition is met. Leaves in the decision tree corresponds to a final decision. The major advantage of using Decision Trees is that it is a simple but powerful model that can effectively perform fast learning on complex data.

Random Forest is also a tree based model, however, it is also an ensemble model that utilizes Decision Tree as its base operating component. Random Forest considers multiple decision trees as its base learners. Data is passed to each of these decision trees and learned model is obtained. The major advantage of Random Forest is that the model can effectively handle issues like data imbalance and noise to provide a robust model. Every created bag is passed to both the machine learning models and the trained ensemble model is obtained. Since a single data is trained on two models, the total number of trained independent base learners is equivalent to twice the number of bags.

D. Prediction based Weight Assignment

Since multiple trained models are created, passing the test data results in multiple predictions being generated. The pre-dictions are generally combined to form the final predictions. However, each model also exhibits a different significance level. Providing equal significance for all the models might result in biased predictions. The significance of each model should be identified prior to the prediction aggregation, which is performed in this phase.

Random instances are selected from the existing training data and a new test set for weight analysis is created. The new training set is passed to each trained model. Every model has been trained with a different set of data, hence each model predicts data in a different manner. Accuracy of prediction of each of the base learners is identified. Weights for base learners are assigned based on the prediction level. Higher weights are provided for base learners with higher predictions, while lower weights are provided for base learners with low prediction levels. These weights are used during the aggregation process for providing significance to the predictions.

E. Weighted Voting based Final Predictions

The streaming transmissions are passed to the created base learners for prediction. These transactions are initially pre-processed and normalized prior to the prediction phase. The transactions are entirely passed to each of the created models. Each model moves through the prediction process to provide the final predictions. Each model provides its own set of predictions for the test data. Hence multiple predictions are obtained for a single instance. These predictions are aggregated based on the weights obtained from the previous phase and the final single prediction set is obtained. The pre-diction process is given by

$$Prediction = \frac{w_1 v_1 + w_2 v_2 + \cdots w_n v_n}{n}$$

where w1,w2,…wn are the weights of each base learner and v1, v2,…vn are the predictions from base learners 1 to n where n is the number of base learners. The resultant predictions are passed to the user or administrator to perform the necessary counter actions if an anomaly has been detected.

F. Data Reinforcement for Prediction Enhancement

The predicted streaming transaction data is stored in the buffer. After a time threshold (waiting time) is reached, this data is added to the training data set. This corresponds to the most recent data in the training set. The threshold time is actual-ly the waiting time required to identify the actual state of the transmission. The actual state might or might not correspond to the predicted result. Hence utilizing the predicted label might lead to biased reinforcement. This leads to the necessity of utilizing a waiting threshold. The actual state is embedded with the transmission data and the transaction is appended to the training data. Addition of such data helps in maintaining the temporal nature of the data. Further, the anomalous signatures are communicated to all the other nodes in the network. This helps in maintaining the decentral-ized nature of the data. This makes the proposed model robust to variations in data and also enables faster and less computationally intensive prediction process.

IV. RESULT ANALYSIS

The DRTAD model has been implemented using Python and has been applied on the NSL-KDD dataset to measure its performance. The attribute details of NSL-KDD dataset is shown in table 1. Experiments were performed and the performance was analyzed based on standard classifier performance metrics.

Table (a) Attributes of NSL-KDD Dataset

No	Attribute
1	src_bytes
2	service
3	dst_bytes
4	flag
5	diff_srv_rate
6	same_srv_rate
7	dst_host_srv_count
8	dst_host_same_srv_rate
9	dst_host_serror_rate
10	dst_host_srv_serror_rate
11	dst_host_diff_srv_rate
12	serror_rate
13	logged_in
14	Attack

The Receiver Operating Characteristics (ROC) curve, representing the True Positive Rate (TPR) and False Positive Rate (FPR) is shown in figure 3. The curve is plotted by considering the FPR in the x-axis and TPR in the y-axis. Area of the curve is used to determine the efficiency of the prediction process. Higher area of the curve represents higher prediction level of the machine learning model. It could be observed from the figure that the proposed DTRAD model exhibits very high TPR levels (0.99 – as shown in figure label) and very low FPR levels (~0). This shows that the DTRAD model exhibits high efficiency in predicting the anomalous transmissions and also exhibits low false alarm levels, leading to a highly effective system.

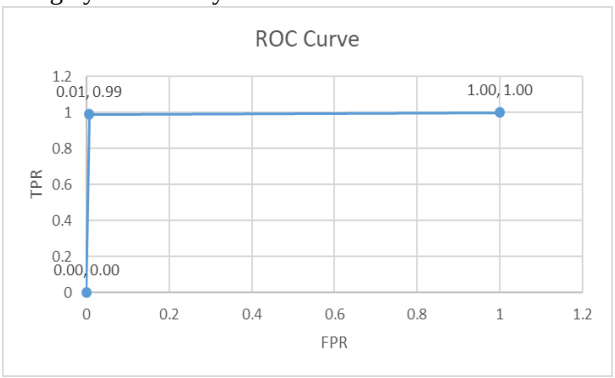

Figure (c): ROC Curve

Similar to the ROC curve, PR curve (Precision Recall Curve) is also used to represent the performance level of a classi-fier model. PR Curve of the proposed DTRAD model is shown in figure.

PR curve is constructed by plotting the recall in x-axis and precision in y-axis. Higher values for both precision and recall indicates high performance of the classifier model. It could be observed from the figure that the proposed model exhibits very high values of both precision and recall (0.98), hence exhibiting higher performance of the DTRAD model.

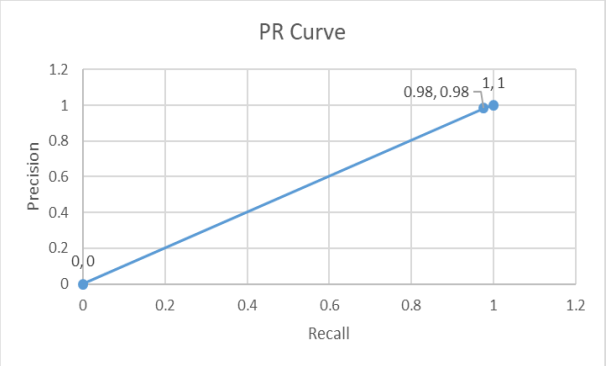

Figure (d): PR Curve

Performance of the proposed DTRAD model is shown in Table 2. The proposed model is observed to exhibit high TPR (0.98) and TNR (0.99) levels. It could also be observed that the proposed model exhibits very low false prediction levels. The aggregate measures; accuracy and F1-Score was also observed to exhibit high values (>0.98), exhibiting the enhanced prediction levels of the proposed model.

Table (b) Performance Analysis of the Proposed Model

	Performance Level
FPR	0.00672269
TPR	0.98934281
TNR	0.99327731
FNR	0.01065719
Recall	0.98934281
Precision	0.98584071
Accuracy	0.99201369
F1 Score	0.98758865

A comparison of the proposed DTRAD model was performed with recent anomaly detection models like RampLoss [11] and the DBSEM model [25] is shown in figure. Comparisons were performed based on accuracy, FPR, Precision and F1 Score. It could be observed that the proposed DTRAD model exhibits the highest performance compared to all the other models, exhibiting the high efficiency of the proposed model.

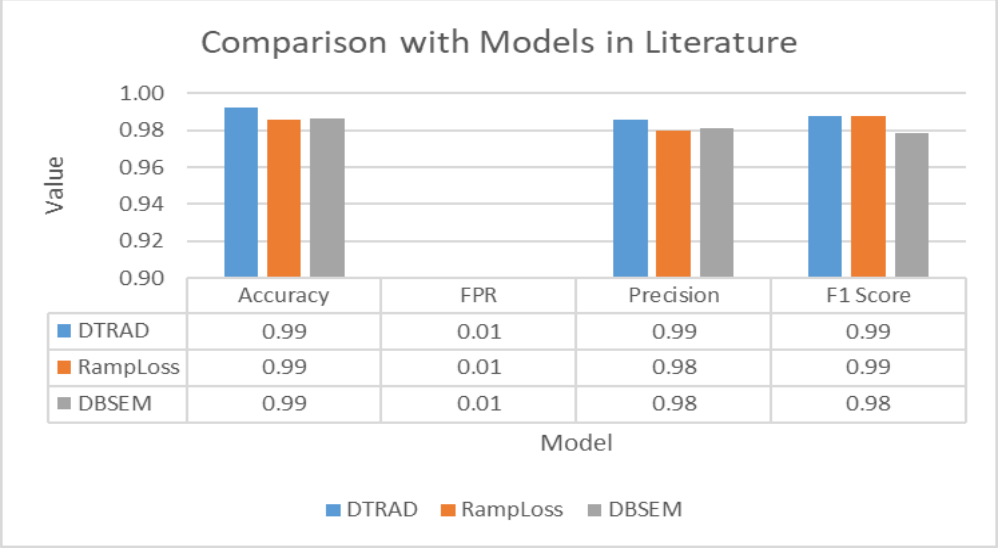

Figure (e): Comparative Analysis

A comparison of the performance has been tabulated and shown in table. The best performances are shown in bold. It could be observed that the DTRAD model exhibits highest performance based on all the reviewed metrics.

Table (c) Comparison of Metrics

	DTRAD (Proposed Model)	RampLoss by Bamakan et al.	DBSEM by Sanjith et al.
Accuracy	**0.99**	**0.99**	**0.99**
FPR	**0.01**	**0.01**	**0.01**
Precision	**0.99**	0.98	0.98
F1 Score	**0.99**	**0.99**	0.98

The major advantage of the proposed DTRAD model arises from the fact that the model operates on only a part of the data, while all the other models operate on the entire training data. This results in reduced time requirements, in-turn leading to faster results. This property serves as the base for IoT devices. This exhibits the capability of the proposed model in operating on devices with low compute resources and still provide effective performances.

I. CONCLUSION

Detecting anomalies serves as one of the major requirements in every IoT based environment. This is mainly due to the high vulnerability of their deployment environments. This work presents the Decentralized Time-window based Anomaly Detection (DTRAD) model, which is less compute intensive compared to existing algorithms. The major advantage of this model is that it filters the training data and utilizes only the recent data for the machine learning process. This results in faster model creation with low computational levels. Further, every node processes its own information and also shares the knowledge about the anomaly signatures. Hence every model is updated, leading to low error levels.
Limitations of the proposed model is that it exhibits a FNR level of 6%. This is due to the usage of time constrained data, leading to elimination of older anomalous signatures. Future enhancements of this model can be based on modify-ing the existing architecture to maintain the anomalous signatures irrespective of their temporal occurrence. This can effectively reduce the FNR levels.

ACKNOWLEDGMENT

We are indebted to IIM Tiruchirappalli, our colleagues and Bharathidasan University for the kind support provided to me during my study.

REFERENCES

1. Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches
2. M.-O. Pahl, F.-X. Aubet, (2018) All eyes on you: Distributed Multi-Dimensional IoT microservice anomalydetection, in: 2018 14th International Conference on Network and Service Management (CNSM)(CNSM 2018), Rome, Italy.
3. M.-O. Pahl, F.-X. Aubet, S. Liebald, Graph-based iot microservice security, in: NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, IEEE, 2018, pp. 1–3.
4. R. C. Deo, (2015) Machine learning in medicine, Circulation 132 (20) 1920–1930.
5. G. D'Angelo, M. Laracca, S. Rampone, (2016) Automated eddy current non-destructive testing through low definition lissajous figures, in: 2016 IEEE Metrology for Aerospace (MetroAeroSpace), IEEE, pp. 280–285.
6. X. Liu, Y. Liu, A. Liu, L. T. Yang, (2018) Defending on–off attacks using light probing messages in smart sensors for industrial communication systems, IEEE Transactions on Industrial Informatics 14 (9) 3801–3811.
7. H. H. Pajouh, R. Javidan, R. Khayami, D. Ali, K.-K. R. Choo, A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in iot backbone networks, IEEE Transactions on Emerging Topics in Computing.
8. I. Poyner, R. Sherratt, Privacy and security of consumer iot devices for the pervasive monitoring of vulnerable people.
9. Internet of Things: A Survey on Machine Learning-based Intrusion Detection Approaches
10. Bamakan, S. M. H., Wang, H., & Shi, Y. Ramp loss K ,(2017),support vector classification-regression; a robust and sparse multi-class approach to the intrusion detection problem. Knowledge-Based Systems, 126, pp.113-126.
11. Y. Li, J. Xia, S. Zhang, J. Yan, X. Ai, K. Dai, (2012) An efficient intrusion detection sys- tem based on support vector machines and gradually feature removal method, Expert Syst. Appl. 39 (1) 424–430. http://dx.doi.org/10.1016/j.eswa.2011. 07.032 .
12. Y. Yi , J. Wu , W. Xu , (2011), Incremental SVM based on reserved set for network intru- sion detection, Expert Syst. Appl. 38 (6) 7698–7707.
13. F. Kuang , W. Xu , S. Zhang , (2014) , A novel hybrid KPCA and SVM with GA model for intrusion detection, Appl. Soft Comput. 18, 178–184 .
14. I. Ahmad , M. Hussain , A. Alghamdi , A. Alelaiwi , (2014) , Enhancing SVM performance in intrusion detection using optimal feature subset selection based on genetic principal components, Neural Comput. Appl. 24 (7-8) 1671–1682 .
15. R. Chitrakar , C. Huang , (2014) , Selection of candidate support vectors in incremental SVM for network intrusion detection, Comput. Securit. 45 231–241 .
16. G. Wang , J. Hao , J. Ma , L. Huang , (2010), A new approach to intrusion detection using artificial neural networks and fuzzy clustering, Expert Syst. Appl. 37 (9) ,6225–6232 .
17. G. Kim , S. Lee , S. Kim , (2014) , A novel hybrid intrusion detection method integrat- ing anomaly detection with misuse detection, Expert Syst. Appl. 41 (4) 1690–1700 .
18. R. Singh , H. Kumar , R. Singla , (2015) , An intrusion detection system using network traffic profiling and online sequential extreme learning machine, Expert Syst. Appl. 42 (22) ,8609–8624 .
19. C.-F. Tsai, C.-Y. Lin, (2010) , A triangle area based nearest neighbors approach to intru- sion detection, Pattern Recognition 43 (1) ,222–229. http://dx.doi.org/10. 1016/j.patcog.2009.05.017 .
20. A Real-Time Intrusion Detection System for Wormhole Attack in the RPL based Internet of Things
21. Semi-supervised multi-layered clustering model for intrusion detection
22. Intelligent intrusion detection systems using artificial neural networks
23. Efficient Physical Intrusion Detection in Internet of Things: A Node Deployment Approach
24. Timely detection and mitigation of IoT-based cyberattacks in the smart grid
25. S L Sanjith , Dr. E George Dharma Prakash Raj, (2019) , A Comprehensive Analysis of Machine Learning Models for Real Time Anomaly Detection in Internet of Things, International Journal of Computer Sciences and Engineering, 6 (11) ,932–937.
26. S L Sanjith, Dr. E George Dharma Prakash Raj, (2019) , Decentralized Bagged Stacking Ensemble Mechanism (DBSEM) for Anomaly detection, Lecture notes in network and systems (Forthcoming)
27. S L Sanjith , Dr. E George Dharma Prakash Raj, (2019) Reinforcement-based heterogeneous ensemble for anomaly detection in streaming environment, International Journal of Intelligent Enterprise, DOI: 10.1504/IJIE.2019.10022335

AUTHORS PROFILE

Dr. E. George Dharma Prakash Raj completed his Master's Degree in Computer Science and Master of Philosophy in Computer Science in the years 1990 and 1998. He has also completed his Doctorate in Computer Science in the year 2008. He has around twenty-seven years of Academic experience and nineteen years of Research experience in the field of Computer Science. Currently he is working as a Faculty in the School of Computer Science, Engineering and Applications at Bharathidasan University, Trichy, India. He has published several papers in International Journals and Conferences related to Computer Science and has been an Editorial Board Member, Reviewer and International Programme Committee Member in many International Journals and Conferences. He has convened many National and International Conferences related to Computer Science..

Mr. Sanjith S L has completed his Master's degree from Manonmanium Sundaranar University, Tirunelveli in Computer and Information Technology in the year 2012 and Bachelor's degree in Electronics Engineering from Cochin University of Science and Technology in the year 1999. He has more than 19 years of experience in the Planning, Designing, Implementation and Management of ICT Infrastructure of reputed organizations out of which 14 + years in academic organizations. Currently he is working as Systems Administrator at Indian Institute of Management Tiruchirappalli (IIM Trichy). Before joining IIM Trichy, he was working as Senior System Manager at PAACET, Trivandrum. He also worked as part-time consultant in few organizations for the implementation of ERP. His experience includes Managed LAN (OFC & Ethernet), Servers with VMs, ERP implementation, Controller based WiFi network, IP Telephony system, IP Surveillance system and Automated Audio visual solutions..