Performance Analysis of Anomaly Based Network Intrusion Detection Systems

Md. Zainal Abedin, Kazy Noor-e-Alam Siddiquee,
M. S. Bhuyan, Razuan Karim
Faculty of Science, Engineering and Technology
University of Science & Technology Chittagong
Chittagong-1079, Bangladesh
jakcse99@gmail.com, knas11@gmail.com,
m.s.bhuyan@gmail.com, karim7@usa.com

Mohammad Shahadat Hossain
Department of Computer Science and Engineering
University of Chittagong
Chittagong-4331, Bangladesh
hossain_ms@cu.ac.bd

Karl Andersson
Pervasive and Mobile Computing Laboratory
Luleå University of Technology
S-931 87 Skellefteå, Sweden
karl.andersson@ltu.se

***Abstract*—Because of the increased popularity and fast expansion of the Internet as well as Internet of things, networks are growing rapidly in every corner of the society. As a result, huge amount of data is travelling across the computer networks that lead to the vulnerability of data integrity, confidentiality and reliability. So, network security is a burning issue to keep the integrity of systems and data. The traditional security guards such as firewalls with access control lists are not anymore enough to secure systems. To address the drawbacks of traditional Intrusion Detection Systems (IDSs), artificial intelligence and machine learning based models open up new opportunity to classify abnormal traffic as anomaly with a self-learning capability. Many supervised learning models have been adopted to detect anomaly from networks traffic. In quest to select a good learning model in terms of precision, recall, area under receiver operating curve, accuracy, F-score and model built time, this paper illustrates the performance comparison between Naïve Bayes, Multilayer Perceptron, J48, Naïve Bayes Tree, and Random Forest classification models. These models are trained and tested on three subsets of features derived from the original benchmark network intrusion detection dataset, NSL-KDD. The three subsets are derived by applying different attributes evaluator's algorithms. The simulation is carried out by using the WEKA data mining tool.**

Keywords—Intrusion detection systems; machine learning; NSL-KDD; feature selection; classification model; performance analysis

I. Introduction

Network security appeals significant research attention in this the age of Internet. Because of using high-capacity computer networks, vulnerability of security breach has also increased to a great extent. The intruders are nowadays attacking in numerous ways, such as Denial of Service (DoS) attacks, illegal access of network resources, spoofing, botnets, and so on. To defend these types of intrusion, existing methods such as firewalls, password policy, and cryptography are no longer enough to provide complete security. Furthermore, these defensive ways fail to secure cloud computing networks and core networks for next generation pervasive computing environments, such as Internet of Things (IoT) due to failure in detecting internal intrusions. The detection of internal intrusion is a challenging job due to the distributed network infrastructure over wide geographic areas. Hence, these types of network are very vulnerable to security breach as their extensive use. It is therefore a great need to secure the communication medium against intrusions by applying reliable security systems, such as IDSs with intelligent security measures [1][2][3][4]. Besides, it is important to ensure the security by detecting intrusion in Wireless Sensor Networks (WSN) based systems, which are deployed to assess the risk of flooding, to support the irrigation and many other areas [5][6][7][8][9][10][11][12].

Based on deployment, IDSs can be categorized into two classes: host and network based IDSs. Network-based IDSs capture data packets travelling across the networks and detect any anomalies by using known patterns. Based on detection mechanisms, IDSs are classified into anomaly, signature, hybrid, and specification based methods. Any abnormality in the packet is treated as an intrusion or attack. The pattern of the packet can be classified as normal or attack by machine learning models. The key benefit of this approach is high detection rates, less false alarms, and fast computational times [13].

In this generation of computing, the field of artificial intelligence (AI) is one of the topics for developing intelligent systems, which has the capability to take real time decisions without human intervention. Machine learning is a rich area in AI that has many application areas, such as image understanding, regression analysis, smart decision support systems, anomaly detection from packet label data in networks, autonomous driving, pervasive computing, Internet of Things (smart agriculture, smart cities), etc. The arrival of big data has stirred comprehensive benefits in machine learning and cyber security by facilitating a bunch of rich algorithms to classify patterns and make the prediction more accurate than before [14][15]. The research community has made efforts analyzing prospects and challenges of machine learning models for developing smart systems based on AI [16][17][18]. In particular, the main goals of this research are to build intelligent systems focusing on learning from big data with high efficiency, real time computing cost and significant predictive accuracy.

Buczak et al. [19] and Li et al [20] did comprehensive surveys of data mining and machine learning algorithms which are used in IDSs. The most commonly used learning models include Naïve Bayes, Artificial Neural Networks (ANN), clustering, fuzzy association rules, decision tress, ensemble learning, Random Forest (RF), evolutionary computations (such as Genetic algorithm, Particle Swarm Optimization, Ant Colony Optimization, Artificial Immune System), hidden Markov model, Support Vector Machine (SVM) [21] and inductive learning. Malhotra et al. [22] developed an IDS for software define networks by means of K-means++ and adaptive boosting algorithms. The features section was done by ranking the features using an RF algorithm. The most common dataset used to evaluate learning model is KDD 1999 [23] that was created based on the DARPA 1998 TCP/IP data captured by packet sniper (pcap). This dataset has some demerits such as redundant and duplicate records. An innovative data set, NSL-KDD, that comprises of records of the original KDD data, was proposed to overcome the shortcomings of KDD. In this work, the NSL-KDD dataset is used to train and test the models.

To manage mobility in fifth generation (5G) mobile wireless systems handling large numbers of IoT devices and to provide a reliable and secure ubiquitous connection, development of a light-weight threat management is a in important research concern. In this regard, IDSs based on machine learning algorithms can open up new dimensions. The IoT infra-structure is usually a three tier architecture with the physical layer, the network layer, and the application layer. The physical layer percepts the physical world through sensors, actuators, RFID devices, and are often built using a Low Power and Lossy Network (LLN). The network layer is responsible for transport and networking capabilities for routing the data from the physical environment to the application layer for further processing. A gateway, called border router, is necessarily placed between the physical layer and the network to incorporate the LLN protocols with network layer protocols. Finally, the application layer consists of the edges to develop IoT applications. To secure IoT networks, IDSs can be deployed in distributed and centralized manners. Taking the distributed approach, the IDS is employed in each physical device of the LLN network, while in the centralized approach, the IDS is positioned in a key location, for example, in the border router or a dedicated node. Every frame that the LLN nodes generate and transmit to the Internet cross the border router as well as the requests that Internet clients send to the LLN nodes have to undergo inspection. Hence, the IDS placed in a border router can scrutinize all the traffic exchanged between the LLN and the Internet. There are also hybrid IDS schemes combining the features of the distributed and centralized approaches [24].

The objective of our work is to evaluate the performance of some learning models for network intrusion classification by considering basic metrics such as accuracy, area under curve, precision, recall, and f-score. To evaluate the classification accuracy, we create three subsets of the NSL-KDD original dataset with the help of attribute evaluators of the WEKA data mining tool for multiclass detection problem.

The organization of the rest of paper is the following; our methodology is discussed in Section 2, while our simulation set-up is described in Section 3. Our model is evaluated in Section 4, and finally, our conclusion and future research directions are presented in Section 5.

II. Methodology

The methodology depicted in Fig. 1 outlines the steps to execute how our machine learning model is trained and evaluated. This section also illustrates properties of the NSL KDD dataset, how data preprocessing is implemented, the brief idea of selected supervised models (Naïve Bayes, Multilayer Perceptron, J48, Naïve Base Tree, and Random Forest), and model evaluation metrics.

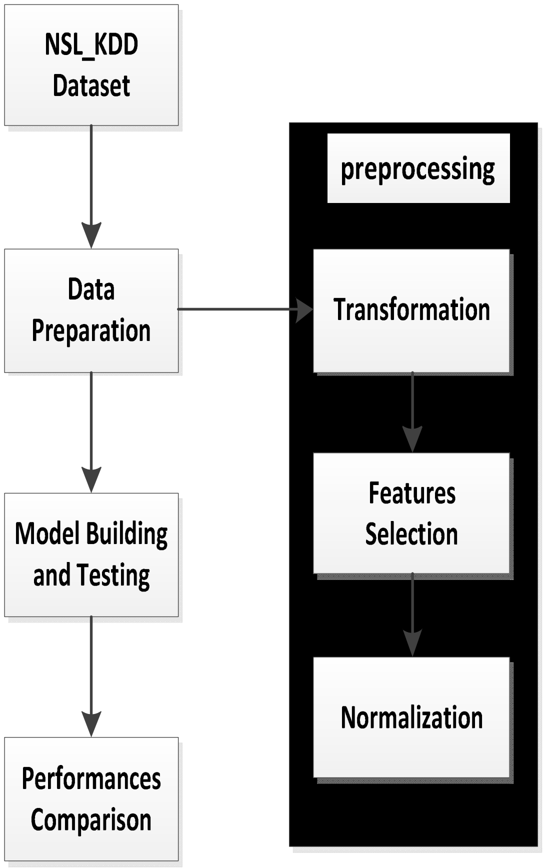

Fig. 1. Methodology.

A. *NSL KDD Dataset*

Two benchmark datasets, KDDCup 1999 and NSL-KDD, are widely used for simulation of IDSs. The NSL-KDD dataset is an improved version of the KDDCup 1999 dataset where redundant and duplicated data is removed from the training and testing datasets. Every instance of this dataset has 41 features and is labeled as either normal or attack. The attack types fall into the following four classes:

Denial of Service (DoS) attacks: In this type of attack, the intruder tries to keep the network busy by exploiting the bandwidth that results in denial of service for legitimate requests.

User to Root (U2R) attacks: By logging in as a normal user, the intruder tries to access the system with root privilege.

Remote to Local (R2L) attacks: The attacker tries to locate vulnerability to access the system remotely.

Probing: The intruder tries to gather information about a network in order to bypassing its security policy

B. *Data Preparation*

The performance of a learning model is dependent on the quality features. Data preparation is an important step when building a model. This phase consists of transformation, features selection, and normalization. Every instance of NSL-KDD has 42 attributes, where the first 41 are predictors or attributes, and the last one is the class name. The transformation step converts the nominal attributes into unique numeric values. For example “TCP” is transformed to 1, the flag “OTH” is transformed to 4, and so on. Table 1 illustrates the result of attributes transformation. Feature selection increases the accuracy of the learning model by eliminating unrelated and redundant attributes. Significant features of data must be inspected to reduce dimensionality. In the NSL dataset, every instance has 42 attributes to recognize a class.

TABLE I. TRANSFORMATION OF NOMINAL ATTRIBUTES INTO NUMERIC

Type	Features	Numeric or Normal Value	Type	Features	Numeric or Normal Value
Protocol Type	TCP	1	Flag	OTH	4
	UDP	2		REJ	5
	ICMP	3		RSTO	6
Multiple attack type	Normal	A		RSTOS0	7
	DOS	B		RSTR	8
	Probing	C		S0	9
	R2L	D		S1	10
	U2R	E		S2	11
Services	All services	15 - 86		S3	12
				SF	13
				SH	14

However, all these features are not equally significant in classifying the attack types. So, features selection is implemented in this simulation by using two evaluators: Consistency Subset Eval and Cfs Subset Eval with different search algorithms, such as rank search and best first search. The result is depicted in Table 2. As a result, the original NSL dataset is transformed to three new datasets, numbered 1 to 3, for multiclass intrusion classification. Best First Search searches the space of attribute subsets by greedy hill climbing augmentation with a backtracking algorithm. Rank Search evaluates the attributes based on rank statistics. Before training the model, the 3 datasets are scaled between -1 to 1 by using the max-min normalization method.

TABLE II. FEATURES SELECTION

Problem Type	Dataset No.	Attribute Evaluator	Search Method	Selected Features	Number of features
Multi Attacks Classification	1	Consistency Subset Evaluator	Best First Search	1, 3, 5, 6, 12, 23, 25, 32, 33, 35, 37, 39, 40	13
	2	Consistency Subset Evaluator	Rank Search	2–6, 8–12, 14, 22, 23, 25–27, 29–35, 37–39	26
	3	Cfs Subset Evaluator	Best First Search	3–6, 12, 14, 26, 29, 30, 37, 38	11

C. *Model Building and Testing*

We have considered the following machine learning models to build and test the three datasets derived from the original NSL-KDD benchmark dataset for multiclass network intrusion detection:

Naïve Bayes (NB) is a relatively simple and fast classification model based on Bayes’ theorem and assumes independence among the predictors. It predicts a class as a posterior probability by using Bayes’ formula that uses class prior probability, likelihood, and predictor prior probability. The main limitation of this model is the assumption of independence of the attributes, which is almost absent in real time classification problem.

J48 is an open source Java implementation of the widely used decision tree algorithm named C4.5. In the WEKA tool it appears as weka.classifier.tree.J48. In this model, the over fitting problem of a learning model is reduced by pruning the tree, where the largest tree is selected as a generalized model. This enhances performance significantly. To prune the tree, the confidence factor is set to 0.25.

Naïve Bayes Tree (NBtree) is a hybrid classifier used as a classifier model, which consists of a decision tree using the NB model. In this model, the tree is developed and pruned as the basic decision tree algorithm, although the leaves are replaced by the Bayes classifier.

Multilayer Perceptron (MLP) is a widely used model for classifier and regression analysis due to its capability of input output mapping, fault tolerance, and non-linearity. This model is designed based on a layered architecture consisting of an input layer for features with hidden and output layers of artificial neurons. The synaptic parameter is the only parameter which is trainable. For the classification problem, a back propagation algorithm based on error correction is used to train the model. This simulation sets the learning rate and momentum parameters to 0.3 and 0.2 respectively. The number of hidden layers is fixed by dividing the sum of number of attributes and classes by two. Initially the weights are distributed randomly.

Random Forest (RF) is an ensemble learning model which builds a forest consisting of a number of decision trees. The training is accomplished in this model in a 'bagging' manner. Every tree in the forest gives their individual decision and finally, all the decisions are merged to give one decision by using the max polling method. Hence, this model results in an accurate and stable classification. RF has almost all the hyper parameters of the decision tree and bagging model. This model brings more randomness when rising the trees. Rather than searching for the best predictors, it searches from a random subgroup of predictors. This results in greater diversity, which generally produces better performance. It can be used to build classification and regression models for large scale datasets. The vital constraint of RF is that a big number of trees increase the build time, which is a crucial factor for real time applications.

D. Performance comparison

To evaluate and make comparisons of the performance of the models, the following metrics are considered.

- Precision (P) = TP/(TP + FP), which is the proportion of intrusion classified that it truly occurred
- Recall® = TP/(TP + FN), which is the quantity of correctly classified intrusions versus the total intrusions that exist
- Area Under Curve (AUC), which is a measurement of the area under the Receiver Operating Curve (ROC) being a plot of true positive rate verses false positive rate
- F-score (F) = (2/(1/P) + (1/R)), which creates a tradeoff between the precision and recall to select a satisfactory quantity of classification accuracy
- Accuracy, which reflects correctly classified instances

where TP=True Positive, FP=False Positive, and FN=False Negative.

III. Simulation Results

The simulation is performed using the WEKA open source data mining tool, which provides an easy graphical user interface to train and test machine learning models with different parameters setting. This tool also makes available some good features evaluation algorithms. We have evaluated the NSL KDD-dataset by two algorithms with different search methods and best three subsets are picked based on the performances metrics. Dataset1 is created by using a consistency subset evaluator with best first search, while Dataset 2 is created by using a consistency subset evaluator with rank search. Dataset 3 is created by a Cfs subset evaluator with best first search. These three datasets are trained and tested by using NB, MLP, J48, NBtree, and RF models. The simulation statistics precision, recall, AUC, F-score, accuracy (correctly classified instances), and model build time are recorded for completely unseen test datasets and is presented in Table 3.

TABLE III. Simulation Statistics

Classifi-cation Model	Dataset	Precision	Recall	AUC	Model Build Time
Naïve Bayes	1	0.663	0.663	0.818	34.47 s
	2	0.759	0.711	0.879	90.33 s
	3	0.617	0.662	0.817	32.83 s
Multilayer Perceptron (MLP)	1	0.661	0.712	0.813	470.85 s
	2	0.612	0.674	0.86	797.42 s
	3	0.955	0.962	0.982	477.19 s
J48	1	0.997	0.997	1	155.55 s
	2	0.811	0.781	0.836	347.83s
	3	0.529	0.394	0.598	168.57s
NBtree	1	0.768	0.721	0.892	1607.99s
	2	0.803	0.767	0.91	4411.05 s
	3	0.647	0.66	0.863	1213.68 s
Random Forest	1	0.792	0.716	0.901	433.26 s
	2	0.785	0.717	0.877	598.42 s
	3	0.989	0.989	1	327.47 s

A simulation environment WEKA (Waikato Environment for Knowledge Analysis) was used in order to make the implementation easier. WEKA is a machine learning open source tool. It provides different tools for implementation of machine learning and contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

IV. Performance Analysis

To make a comparison, the simulation statistics are visualized by a Python data visualization tool. Fig. 2 presents the comparison of AUC, Precision and Recall for three datasets and the above-mentioned learning models. Fig. 2 shows that the J48 decision tree algorithm performs better in terms of visualized parameters. The same visualization is done for dataset 2 and 3 in Figs. 3 and 4. Again, in dataset 2, J48 performs better. In dataset 3, the ensemble learning model – RF outperforms. The evaluation in terms of correctly classified instances is visualized in Fig. 5. From this graphical presentation, dataset 1 and 2 give better performances than dataset 3. In dataset 1 the accuracy of J48 is close to the upper bound, while in dataset 3, the accuracy of MLP and RF are noticeable and significant. The f-score is analyzed and plotted in Fig. 6. From Figs. 2 to 4, it is observed that the performance of these models depends on the features selection. So, optimal feature selection is an important parameter to develop an efficient IDS. Also, the dimension of the features vector is important for real time applications.

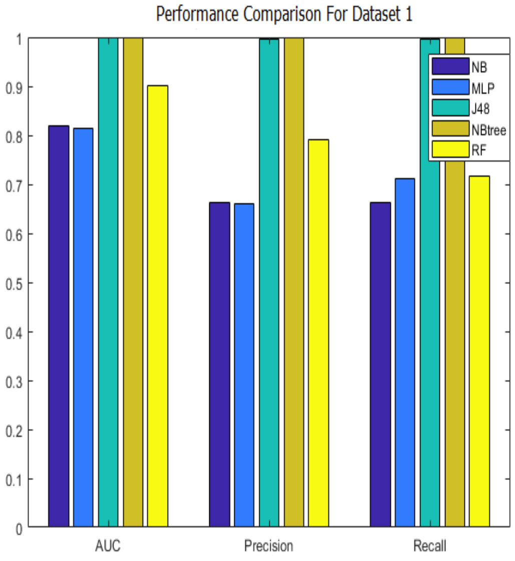

Fig. 2. Model Evaluation for Dataset 1.

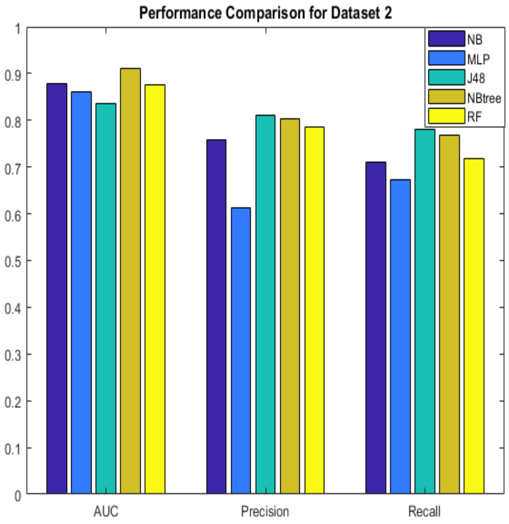

Fig. 3. Model Evaluation for Dataset 2.

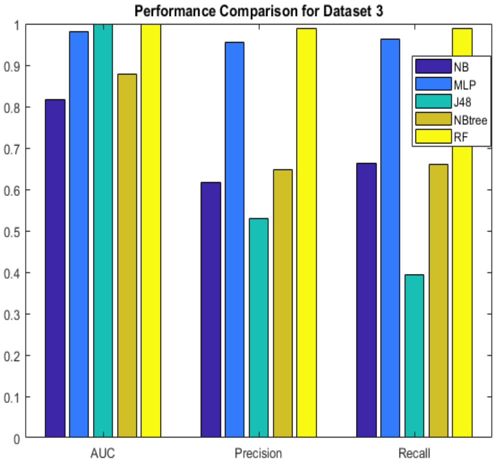

Fig. 4. Model Evaluation for Dataset 3.

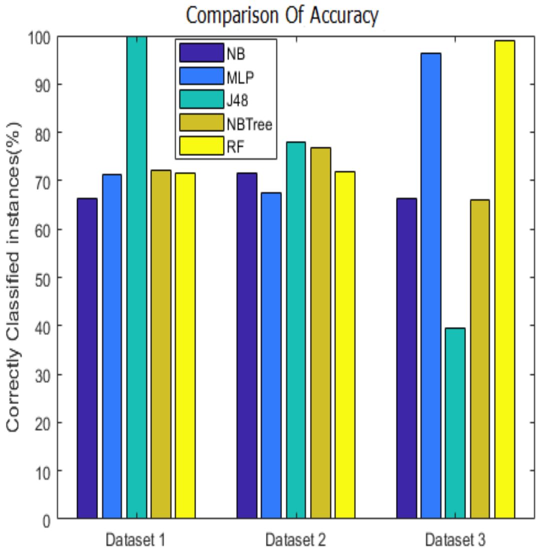

Fig. 5. Comparison of Accuracy.

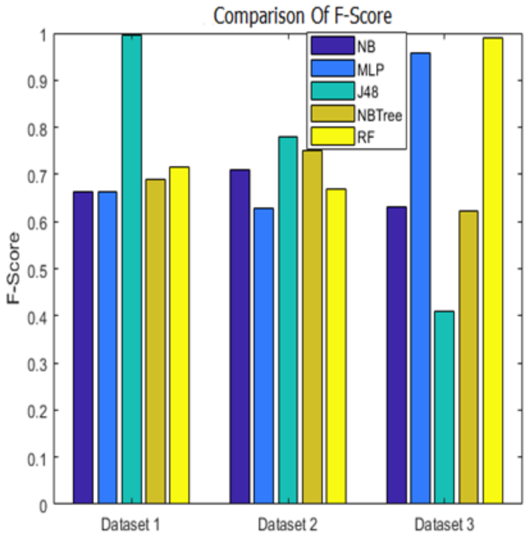

Fig. 6. Comparison of F-Scores.

The model complexity in terms of time is a very important parameter for applicability in an IoT context, because of constraints in the network infrastructure environment. Hence, a lightweight intrusion detection algorithm is necessary to keep the network alive as long as possible. Fig. 7 depicts the build time (training time) of the algorithms for three datasets. It is clear that the best case built time was obtained from NB and the worst case was recorded from NBtree model.

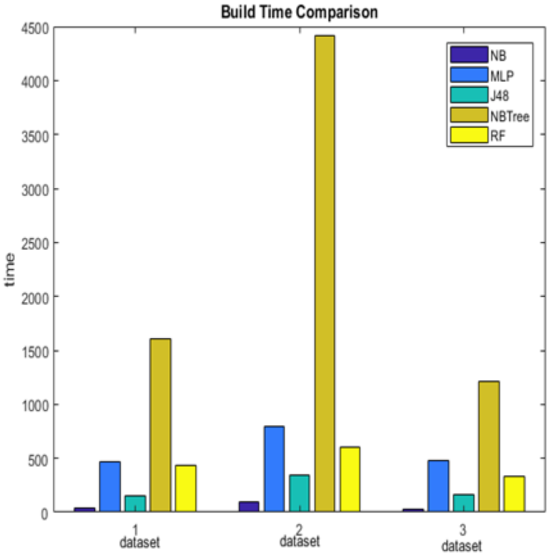

Fig. 7. Comparison of Model Complexity.

V. Conclusion and Future Work

To ensure the integrity of data, cyber security plays a very significant role. Due to widespread geographic area of the networks and big data, the cyber security policy has to be designed dynamically by taking the current challenges of the Internet technology into account. The revolution of AI opens up new dimensions to develop security policies to stop the intruders and keep the data integrity for better decision making. Machine learning models are now gaining more popularity to develop network IDSs due to their great dynamic and classification accuracy in anomaly detection by snipping the packet data. In this regard, this paper illustrates a comparative analysis of some popular learning models, which are being used to develop IDSs. As the accuracy of machine learning models depend on the features, the selection of significant features is very crucial. Also, the dimensionality of features vectors is also important for real time applications. We have created three features vectors from the original NSL-KDD by using features selection algorithm. From the simulation results, the findings are that the J48 algorithm performs well in case of dataset 1 and 2 and RF outperforms in case of dataset 3.

One of the challenges of cyber security is that a machine learning model has to be retrained. A good dimension of research can be to explore fast incremental learning algorithms for dynamic anomaly detection.

Acknowledgment

We would like to extend our gratitude to the Swedish Research Council (under grant 2014-4251) for the funding and the provision of resources to conduct our study.

References

[1] M. O'Neill, ”The Internet of Things: do more devices mean more risks?,” Computer Fraud & Security 2014(1):16–17, January 2014.

[2] A. Sultana and M. Jabbar, “Intelligent network intrusion detection system using data mining techniques,” Proceedings of 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), July 2016.

[3] L. Oneto, F. Bisio, E. Cambria, and D. Anguita, “Statistical learning theory and ELM for big social data analysis,” IEEE Computational Intelligence Magazine, 11(3):45–55, August 2016.

[4] A. Chaudhary, V. Tiwari, and A. Kumar, “A novel intrusion detection system for ad hoc flooding attack using fuzzy logic in mobile ad hoc networks,” Proceedings of International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), May 2014.

[5] K. Andersson and M. S. Hossain, "Heterogeneous wireless sensor networks for flood prediction decision support systems," Proceedings of 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April–May 2015.

[6] S. Thombre, R. Ul Islam, K. Andersson, and M. S. Hossain, "Performance analysis of an IP based protocol stack for WSNs," Proceedings of 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), April 2016.

[7] K. N. E. A. Siddiquee, F. F. Khan, K. Andersson, and M. S. Hossain, "Optimal dynamic routing protocols for agro-sensor communication in MANETs," Proceedings of 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), January 2017.

[8] K. Andersson and M. S. Hossain, “Smart risk assessment systems using belief-rule-based DSS and WSN technologies,” Proceedings of 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), May 2014.

[9] R. Ul Islam, K. Andersson, and M. S. Hossain, “Heterogeneous wireless sensor networks using CoAP and SMS to predict natural disasters,” Proceedings of 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), May 2017.

[10] Z. Abedin, A. S. Chowdhury, M. S. Hossain, K. Andersson, and R. Karim, “An interoperable IP based WSN for smart irrigation system,” Proceedings of 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), January 2017.

[11] Z. Abedin, S. Paul, S. Akhter, K. N. E. A. Siddiquee, M. S. Hossain, and K. Andersson, “Selection of Energy Efficient Routing Protocol for Irrigation Enabled by Wireless Sensor Network,” Proceedings of 2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), October 2017.

[12] S. Thombre, R. Ul Islam, K. Andersson, and M. S. Hossain, “IP based Wireless Sensor Networks: Performance Analysis using Simulations and Experiments,” Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 7(3):53–76, September 2016.

[13] Z. Inayat, A. Gani, N. B. Anuar, M. K. Khan, and S. Anwar, ”Intrusion response systems: Foundations, design, and challenges,” Journal of Network and Computer Applications 62(2016):53–74, February 2016.

[14] L. Zhou, S. Pan, J. Wang, and A. V. Vasilakos, “Machine learning on big data: Opportunities and challenges,” Neurocomputing, 237(2017):350– 361, May 2017.

[15] S. Yu, “Big Privacy: Challenges and Opportunities of Privacy Study in the Age of Big Data,” IEEE Access, 4:2751–2763, June 2016.

[16] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, “Efficient Machine Learning for Big Data: A Review,” Big Data Research, 2(3):87–93, September 2015.

[17] X.-W. Chen and X. Lin, “Big Data Deep Learning: Challenges and Perspectives,” IEEE Access, 2:514–525, May 2014.

[18] A. L’Heureux, K. Grolinger, H. F. Elyamany, and M. A. M. Capretz, “Machine Learning With Big Data: Challenges and Approaches,” IEEE Access, 5:7776–7797, April 2017.

[19] A. L. Buczak and E. Guven, “A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,”. IEEE Communications Surveys & Tutorials, 18(2):1153–1176, October 2015.

[20] J. Li, Z. Zhao, and R. Li, ”Machine learning-based IDS for software defined 5G network,” IET Networks, 7(2):53–60, March 2018.

[21] M. S. Pervez and D. M. Farid, “Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs,”

Proceedings of the 8th International Conference on Software, Knowledge, Information Management, and Applications (SKIMA), December 2014.

[22] S. Malhotra, V. Bali, and K. Paliwal, “Genetic programming and K-nearest neighbour classifier based intrusion detection model,” Proceedings of 2017 7th International Conference on Cloud Computing, Data Science & Engineering – Confluence, January 2017.

[23] S. J. Stolfo, KDD Cup 1999 Data Set, University of California Irvine, KDD repository [Online]. Available: http://kdd.ics.uci.edu

[24] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. Carlisto de Alvarenga, ”A survey of intrusion detection in Internet of Things,” Journal of Network and Computer Applications 84:25–37, April 2017.