IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Intrusion Detection Network Using Feature Optimization Technique Based On Neural Network: A Review

Priyanka Sataw[1], Chetan Gupta[2], Dr. Ritu Shrivastava[3]
M. Tech Research Scholar, Dept. of CSE[1], Assistant Professor, Dept. of CSE[2], HOD Dept. of CSE[3]
SIRTS, Bhopal, India[1], SIRT, Bhopal, India[2, 3]

Abstract: Network security has become a vital issue in present scenario, as it is very important to maintain high-level security to ensure safety and trusted communication of information in a computer network. Presently data communication between network and host is leading to a different number of intrusion activities and misuses. Thus, to provide a solution for such malicious attacks, the Intrusion Detection System (IDS) came into existence and held the attention of the computer science field. There are various approaches to design an IDS. Though there is several existing literature on IDS issues, still they pose some disadvantages. It is important for IDS, to detect previously known attacks with high accuracy and unseen attacks to minimize the losses and their impact at an early stage. The process of selecting the best feature is a vital role to ensure the performance, speed, accuracy, and reliability of the detector. Here, in this research study we will try to improve detection accuracy and efficiency for hybrid IDS model based on feature optimization method using multilayer neural network (ML-NN).

Index Terms - Intrusion Detection System **(**IDS), Neural Network, Feature Optimization

I. INTRODUCTION

Millions of people use computers for a variety of tasks, including banking, insurance, shopping, filing taxes, protesting, serving in the military, and keeping track of students, among other things. The introduction of distributed systems and the utilization of networks and communication facilities for the transfer of data between terminal users and computers, as well as between peer to peer networks, have had the greatest impact on security. Data transmission must be protected by network security measures. To get to the security necessities of an association really assess and pick different security items and approaches, the individual liable for the security, need some orderly approach to characterizing the necessity for security and describing the ways to deal with fulfilling those prerequisite, such methodology is IDS. Security is the absence of danger and the protections taken to ward off attacks [1, 2].

Security is wide point and covers large number sins. Let's put it this way: let's say you want to send a message to someone and want to make sure that no one else can read it. However, it is possible that an additional individual will open the second one or hear the electronic communication. The majority of security issues result from malicious individuals attempting to gain or harm something.

A device or software program known as IDS searches a system or framework for malicious movement or arrangement infringement. The installation of the IDS and various soft computing methods like Genetic Algorithm, ANN, Support Vector Machine, Back Propagation, Particle Swarm Optimization, Extreme Learning Machine, TLBO, and others can be used to detect intrusions are utilized to make IDS. Above AI based half and half interruption location framework model is joined with certain successful computerized reasoning strategies are viewed as the most proficient methodology for interruption identification. The elements of IDS are to identify the interruptions, produce the spring up message to the client, and go to the important restorative lengths. The prevention-based approach to computer and network security has a few drawbacks. It is probably impossible to create a system that is completely secure. The avoidance based security reasoning contains the client's action and efficiency. Therefor, interruption discovery frameworks are planned in light of different location techniques [3].

Utilizing security information and event management (SIEM) framework, any identified movement or violation is typically detailed to a manager or gathered midway. A SIEM system unites yields from different sources and uses alert filtering techniques to perceive vindictive activity from misleading alarms [4]. A duplicate of live traffic is shipped off interruption location framework from network tap to perform complex examination and examinations and this traffic isn't steered back again to the confided in network. Because it does not work with live traffic, this is also known as passive monitoring. Internal attacks can be detected by some IDS as well. As a result, IDS can be set up wherever we need it to be in the network. This IDS system first collects all incoming traffic or behavior from the target computer or network, then learns and creates patterns and stores them in a database as an example. It then checks and monitors all incoming traffic or behavior with training patterns and generates an alarm to notify the administrator of dangers, as shown in fig. 1 below. Host-based IDS and network-based IDS are the two main types of intrusion detection systems. HIDS monitors each host individually. The HIDS usually works by accessing log files or monitoring the host's usage in real time to identify risky actions. While HIDS is installed on the client computer, NIDS is added to the network.

All incoming and outgoing traffic is checked and controlled by a network-based intrusion detection system at a single network component [5, 6]. By placing a single capture tool (sensor), such as a sniffer, on the chock point of the segment, this is accomplished. This sensor determines whether the traffic packets in this segment are attacks or normal activities by capturing all network traffic [7]. NIDS, on the other hand, looks at all network traffic.

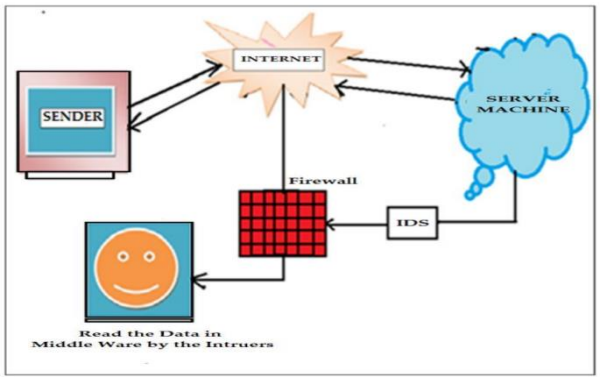

Fig. 1: Situated IDS into the network with the Firewall

II. LITERATURE REVIEW

S. K. B Sangeetha et al. [1], in imbalanced organization traffic, pernicious digital assaults can regularly stow away in a lot of typical information. It displays a serious level of covertness and jumbling in the internet, making it hard for Network IDS to guarantee the precision and practicality of discovery. This paper explores AI and profound learning for interruption recognition in imbalanced organization traffic. It proposes an original Difficult Set Sampling Technique (DSSTE) calculation to handle the class awkwardness issue. To start with, utilize the Edited Nearest Neighbor (ENN) calculation to separate the imbalanced preparing set into the troublesome set and the simple set. Then, utilize the K Means calculation to pack the larger part tests in the troublesome set to diminish the larger part. Zoom in and out the minority tests' persistent characteristics in the troublesome set integrate new examples to expand the minority number.

S. Sengan et al. [2], successful and proficient malware recognition is at the bleeding edge of examination into building secure computerized frameworks. Similarly as with numerous different fields, malware location research has seen a sensational expansion in the utilization of AI calculations. One AI strategy that has been utilized broadly in the field of example matching overall—and malware identification specifically—is covered up Markov models (HMMs). Gee preparing depends on a slope climb, and thus we can frequently work on a model via preparing on numerous occasions with various beginning qualities. In this exploration, we think about helped HMMs (utilizing AdaBoost) to HMMs prepared with different arbitrary restarts, with regards to malware identification. These procedures are applied to an assortment of testing malware datasets. We observe that irregular restarts perform shockingly well in contrast with helping. Just in the most troublesome "cold beginning" situations (where preparing information is seriously restricted) does helping seem to offer adequate improvement to legitimize its higher computational expense in the scoring stage.

K. Aravindhan et al. [3], in this paper, aimed at detection of internal intruders in HIDS. Commonly used login ids and passwords may be shared along with co-workers for professional purposes, which can be tampered or used by the attackers as a means of intrusion into the system details. The user was monitored and System Calls (SC) was extracted and the habitual SC pattern based on the habits of the user was taken into account and the profile of the user was stabilized. The forensic technique and other data mining techniques were applied at SC level host IDS to spot the internal attacks. Along with the user login credentials the forensic technique was applied to investigate the computer usage fashion against the collected user profile pattern and thereby check the identity of the user.

D. Kanthavel et al. [4], in this paper, With the decision rate threshold of 0.9, the system was able to perform with an accuracy rate of 94%. Nokia Research Center researchers modeled HIDS for mobile devices. The limitation include that each protocol state consume resources for tracing and testing, and its inability to guess the attacks resembling benign protocol. Access control fills in as the cutting edge of resistance against interruptions, bolstering both confidentiality and integrity parameters. Intrusion detection is the process of progressively observing the events occurring in a PC or network, examining them for indications of conceivable episodes and often interdicting the unapproved access. A state transition diagram can be constructed for the sequence of events, but not for the complex forms and hence the attacks having complex behavior which cannot be modeled as the state transition diagram will go unnoticed by the system.

S. Li et al. [5], in this paper, along with various protection mechanisms accompanied with mobiles they felt an urge for attack monitor methods as a second line of defense. The framework was designed, taking into consideration the privacy of the mobile user in creating the user profile. The framework had a major share with the host-based intrusion detection in line with the network-based detection system, as researchers felt that mobile requires the monitoring system at both ends. The framework included data collection and IDS modules, the former entrusted with responsibility of monitoring the operating system activities, calculating the system measurements and the data collection at the application level and the later feeding on the collected and pre-processed data performs the actual intrusion detection.

R. Kanthavel et al. [6], targeted Advanced Persistent Threat attacks, by analyzing the 30 behavioral pattern of the host user through a 83-dimensional vector, each attribute representing one manner of the user. In order to form the database, they collected 8.7 million features from 4000 malicious and normal programs through the Virtual Machine (VM) environment. The system was designed in such way that frequency of occurrence of each behavior is calculated for each process. C4.5 decision tree was used to build a classifier for the collected information, and each new instance was analyzed against the tree to be segregated as malicious or normal instance. The model had a false positive rate of 5.8% and a false negative rate of 2.0%.

L. Cai et al. [7], in this paper, represented a novel HIDS aimed at discovering unknown malware codes. The collection of previous malware codes was taken as repository and each new sequence of behavior was compared with the repository to identify new malware code. Applied rule-based IDS to tackle the DDoS attacks in which the resources are made unavailable for the user when they are required. The utmost capacity of each of the middle-ware layer was fixed and set of rules were formed to detect the DDoS attacks. The system produced an alert when the count of the requests to a particular resource exceeded a particular threshold and concepts from learning automata were employed to avoid further attacks.

L. Xie et al. [8], in this paper, applied Machine Learning techniques, namely Naive Bayes, a Bayesian Network and Artificial Neural Network, to perform supervised learning of the malicious code. They gathered 323 features for training the classifiers. The detection rate for a specific set of worms was over 98%. Though HIDS were able to perform better by centering the user profile collected across, it prompted the challenge and there was a lack of information of the user Centralized reporting wasn't feasible with HIDS. They consumed the host details and resources which may violate the privacy issues of the user and may dispute with the already existing security protocols.

M. K. Putchala et al. [9], in this paper, centered over detecting intrusion in Routing Protocol for Low Power and Lossy Networks (RPL) attacks. The operations of the RPL were converted into finite state machines through which the network was monitored and any malicious activity was detected. The research was further extended by wherein the simulation trace files were used to model the finite state machines to observe the RPL attacks. The model was further converted into a set of rules to monitor the data transferred between the network nodes. The drawback of the work was that the True Positive Rate was even able to reach 100% but the False Positive Rate was not that low ranging between 0 to 6.78%. Over that it also had an overhead of 6.3% in terms of energy when compared to normal RPL network.

R. Dhaya et al. [10], in this paper, designed IDS for cloud environment based on state transition methodology. A Hidden Markov Model which builds a model where the behavior of user observed over long time period is marked as states and relevant transitions between them was used in their research. They developed three profiles, namely, low, middle and high, based on the matching of the probability of the user to the baseline profile. They give input as VM SC and bring about a diagram of state transition, wherein each transition represents the probability of targeting the next state and the probability of creating next system call. This model is tested against DoS attacks with 100% detection rate but with a poor false positive rate of 5.66%.

Problem formulation:-

Intrusion detection is an extremely convoluted and tedious procedure. There are several works done in this area of the IDS field but not a single one is to develop the best IDS model with the best performance and less time is taken IDS model. So, research questions are the following: (i) why does the author(s) work on the intrusion detection system? (ii) How does the author(s) improve the performance of our model with the previously available model?

In our research work is trying attempt to build up a superior performing IDS model. The creator attempts to offer responses to the above inquiries are the following: (i) The main aim of this research study is to improve the detection rate of IDS with less of time and improve the accuracy of our proposed hybrid IDS model based on feature optimization techniques using Multilayered Neural Network.

(ii) Although the lot of research works is going on PSO, GA, ELM, and PCA with the BPN, still author(s) are trying to improve our algorithm for better results? Feature optimization is a technique for improving the performance of the intrusion detection system in a minimum period.

III. FEATURE OPTIMIZATION TECHNIQUE

Based on how they combine the selection algorithm and model building, feature selection methods are typically divided into three categories [8, 9].

3.1 Filter method

Regardless of the model, filter type methods select variables. They only use general characteristics like the correlation with the predictable variable. The least interesting variables are suppressed by filter methods [10]. Different factors will be important for a grouping or a relapse model used to characterize or to anticipate information. See fig. 2 to see how resistant and effective these methods are to over-fitting and reduce computation time. Channel strategies will quite often choose repetitive factors when they don't think about the connections between factors. However, more complex features, such as the FCBF algorithm, attempt to minimize this issue by removing highly correlated variables.

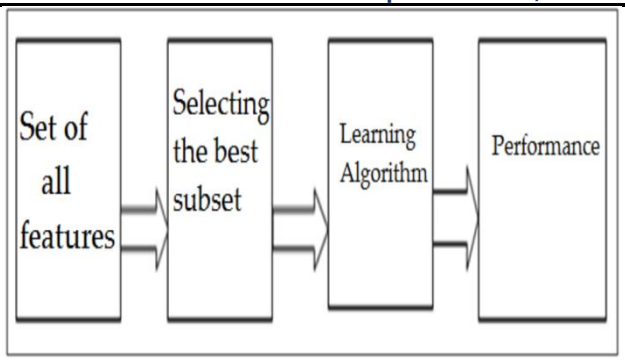

Fig. 2: Step-by-step process of the filter method

3.2 Wrapper method

In contrast to filter approaches, wrapper methods evaluate subsets of variables, making it possible to identify potential interactions between variables, as depicted in fig. The two principal hindrances of these techniques are:

- When the number of observations is insufficient, the risk of overfitting grows.
- The significant amount of time spent computing when there are a lot of variables.

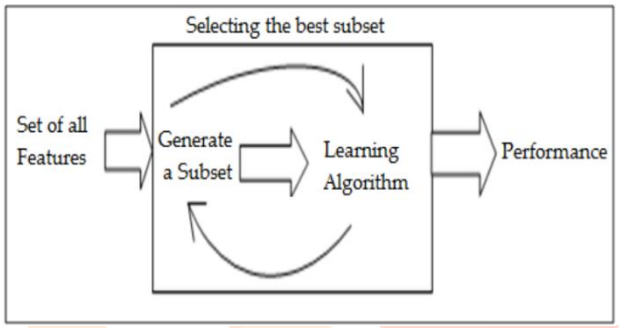

Fig. 3: Step-by-step process of a wrapper method

3.3 Embedded method

Newly proposed embedded methods attempt to combine the benefits of the two previous approaches. A learning algorithm uses its variable selection process to simultaneously select features and classify them.

3.4 Back-Propagation Neural Network

Back-Proliferation Calculation is a generally involved calculation in preparing feed-forward brain networks for administered learning. Speculations of back-proliferation exist for other counterfeit brain organizations (ANNs), and works by and large - a class of calculations is alluded to conventionally as "back-spread". In contrast to a naive direct computation of the gradient for each weight separately, back-propagation efficiently computes the gradient of the loss function regarding the weights of the network for a single input-output example. Because of this efficiency, gradient methods can be used to train multilayer networks, updating weights to reduce loss; The most common method is gradient descent, or variants like stochastic gradient descent. The back-propagation algorithm works by using the chain rule to calculate the gradient of the loss function for each weight. It does this by computing the gradient one layer at a time and iterating backward from the last layer in order to avoid making the same calculations for intermediate terms in the chain rule over and over again [11].

The term "back-propagation" only refers to the algorithm used to calculate the gradient and not its application; however, the term is frequently used in a loose sense to describe the entire learning algorithm, including the gradient's application, such as stochastic gradient descent. The Delta rule, which is the single-layer version of back-propagation generalized by automatic differentiation and in which back-propagation is a special case of reverse accumulation (or "reverse mode"), is how back-propagation makes gradient computation more general. Back Propagation Neural (BPN) network in fig. 4 is an example of the general use of the term "back-propagation" in neural networks.

The algorithm can be broken down into its four parts:

Feed-forward computation, backpropagation to the output layer, backpropagation to the hidden layer, and weight updates are all examples of weight updates. The algorithm is terminated once the error function's value reaches a small enough value.

The BP algorithm's most basic and rough formula is presented here. Other scientists have proposed a few different definitions, but Rojas' definition appears to be quite accurate and straightforward. Weight updates are occurring throughout the algorithm in the final step [12].

The objective of any directed learning calculation is to track down a capability that best guides a bunch of data sources to its right result. A classification task, for instance, would have an animal image as its input and the name of the animal as its correct output. Finding a method to train a multi-layered neural network so that it can learn the appropriate internal representations to enable it to learn any arbitrary mapping of input to output was the driving force behind the development of the back-propagation algorithm. The computation of a loss function's partial derivative, or gradient, for any network weight is the objective of back-propagation [13].

Two phases make up the back-propagation learning algorithm:
1) Propagation 2) Weight revision

Phase 1: Propagation:- The following steps are involved in every back-propagation: The neural network's output value(s) are generated by the forward propagation of a training pattern's input through the network.
Phase 2: Weight update: The following steps must be taken for each weight: The weight's gradient is calculated by multiplying the weight's input activation and output delta. The weight is reduced by a ratio (percentage) of the gradient of the weight [14].

This proportion (rate) impacts the speed and nature of learning; the term for it is the learning rate.
The neuron trains more quickly if the ratio is higher, but the training is more accurate if it is lower. The indication of the angle of weight demonstrates whether the blunder fluctuates straightforwardly with, or conversely to, the weight. As a result, in order to "decline" the gradient, the weight needs to be updated in the opposite direction. The first and second phases are repeated until the network performs satisfactorily [15].

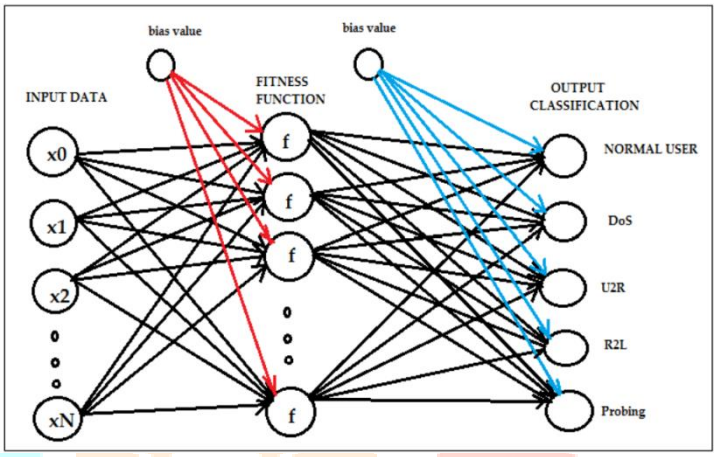

Fig. 4: Basic Diagram of Back-Propagation Neural Network

IV. PROPOSED METHODOLOGY

The proposed technique is based on multilayer neural network. In this paper, explain the multilayer neural network and further data analysis using ML-NN & flow chart will explain result paper.

Multilayer Neural Network (ML-NN)

In this fig 5, ML-NN is a feed-forward neural network used for feature optimization. It is a single hidden layer network; the hidden layer will choose on a random basis and optimize the features in minimum numbers of features. By using ELM reduces the complexity in training by using the random values in wait and bias values. This value is fixed. The property of ML-NN is feature minimization by increases the classification accuracy and improves the learning speed. In this ELM fixed random weights are given into the hidden layer. Here, I worked on an extreme learning machine for feature optimization. This is a single-layered feed-forward neural network (SLFFNS), but the hidden layer or called feature mapping) in ELM need not be tuned [14]. In ML-NN, learning parameters of hidden nodes, input weights, and biases are randomly assigned and need not be tuned. Compared with the traditional algorithm and ML-NN, ML-NN is performing better. In this ML-NN algorithm is SLFFN i.e. single layer feed-forward neural networks are using in classification, regression, clustering, sparse approximation, compression and feature learning with a single layer or multiple layers of hidden nodes, where the parameters of hidden nodes (not just the weights connecting inputs to hidden nodes) need not be tuned. These hidden nodes can be randomly assigned and never updated (i.e. they are random projection but with nonlinear transforms) or can be inherited from their ancestors without being changed. In most cases, the output weights of hidden nodes are usually learned in a single step, which essentially amounts to learning a linear model.

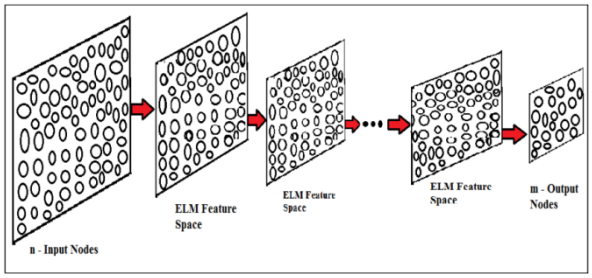

Fig. 5: Feature optimization in the ML-NN working model

V. CONCLUSION

Even though by increasing alertness in network-security problems, the presently ongoing solutions are not suitable for carefully shielding computer organize applications and undertaking security frameworks against the risk from consistently ever-propelling PC organize attack systems. So for this issue, adaptable security methods have been developed to turn out further severe than existing methods.
In our research work will trying attempt to build up a superior performing IDS model based on feature optimization technique. The main aim of this research work will to improve the detection rate of IDS with less of time and improve the accuracy of our proposed hybrid IDS model based on feature optimization techniques using ML-NN.

REFERENCES

[1] S. K. B Sangeetha, Prasanna Mani, V. Maheshwari, Prabhu Jayagopal, M. Sandeep Kumar and Shaikh Muhammad Allayear, "Design and Analysis of Multilayered Neural Network-Based Intrusion Detection System in the Internet of Things Network", Hindawi, 2022.

[2] S. Sengan, O. I. Khalaf, D. K. Sharma, and D. K. Sharma, "Secured and privacy-based IDS for healthcare systems on E-medical data using machine learning approach," International Journal of Reliable and Quality E-Healthcare, vol. 11, no. 3, pp. 1–11, 2022.

[3] K. Aravindhan, S. K. B. Sangeetha, K. Periyakaruppan, E. Manoj, R. Sivani, and S. Ajithkumar, "Smart charging navigation for VANET based electric vehicles," in Proceedings of the 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1588–1591, IEEE, Coimbatore, India, March 2021.

[4] D. Kanthavel, S. K. B. Sangeetha, and K. P. Keerthana, "An empirical study of vehicle to infrastructure communications an intense learning of smart infrastructure for safety and mobility," International Journal of Intelligent Networks, vol. 2, pp. 77–82, 2021.

[5] S. Li, S. Lai, Y. Jiang, W. Wang, and Y. Yi, "Graph Regularized Deep Sparse Representation for Unsupervised Anomaly Detection," Computational Intelligence and Neuroscience, vol. 2021, Article ID 4026132, 19 pages, 2021.

[6] R. Kanthavel, S. K. B. Sangeetha, and K. P. Keerthana, "Design of smart public transport assist system for metropolitan city Chennai," International Journal of Intelligent Networks, vol. 2, pp. 57–63, 2021.

[7] L. Cai, Z. Chen, C. Luo et al., "Structural temporal graph neural networks for anomaly detection in dynamic graphs," in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3747–3756, Queensland, Australia, October 2021.

[8] L. Xie, D. Pi, X. Zhang, J. Chen, Y. Luo, and W. Yu, "Graph neural network approach for anomaly detection," Measurement, vol. 180, Article ID 109546, 2021.

[9] M. K. Putchala, Deep Learning Approach for Intrusion Detection System (Ids) in the Internet of 9ings (Iot) Network Using Gated Recurrent Neural Networks (Gru) Master 9esis, Wright State University, Dayton, OH, USA, 2017.

[10] R. Dhaya, S. K. B. Sangeetha, and A. Sharma, "Improved performance of two server architecture in multiple client environment," in Proceedings of the 2017 4th International Conference on Advanced Computing And Communication Systems (ICACCS), pp. 1–4, IEEE, Coimbatore, India, January 2017.

[11] G. Fortino and P. Trunfio, Eds., Internet of 9ings Based on Smart Objects: Technology, Middleware and Applications, Springer Science & Business Media, Berlin, Germany, 2014.

[12] Y. H. Kung and H. C. Hsiao, "GroupIt: lightweight group key management for dynamic IoTenvironments," IEEE Internet of 9ings Journal, vol. 5, no. 6, pp. 5155–5165, 2018.

[13] P. Porambage, A. Braeken, C. Schmitt, A. Gurtov, M. Ylianttila, and B. Stiller, "Group key establishment for enabling secure multicast communication in wireless sensor networks deployed for IoT applications," IEEE Access, vol. 3, pp. 1503–1511, 2015.

[14] N. Andreadou, M. O. Guardiola, and G. Fulli, "Telecommunication technologies for smart grid projects with focus on smart metering applications," Energies, vol. 9, no. 5, p. 375, 2016.

[15] S. Omar, A. Ngadi, and H. H Jebur, "Machine learning techniques for anomaly detection: an overview," International Journal of Computer Application, vol. 79, no. 2, pp. 33–41, 2013.