ASSERTIVE SEARCH OPTIMIZATION ROUTING BASED RECURRENT NEURAL NETWORK (RNN) FOR INTRUSION DETECTION IN MANET

[1]Dr. P. Revathi, [2]Dr. N. Karpagavalli, [3]Dr. K. Juliet Catherine Angel

[1]Assistant Professor of Computer Science,
Holy Cross College (Autonomous),
Tiruchirappalli – 620 002

[2]Assistant Professor of Computer Science,
Holy Cross College (Autonomous),
Tiruchirappalli – 620 002

[3]Assistant Professor of Computer Science,
Holy Cross College (Autonomous),
Tiruchirappalli – 620 002

Abstract

In Deep learning contains procedures that are initially trained in context to “learn” their functions and then used with invisible input for sorting dedications. The benefits of Mobile Adhoc Network (MANETS) and the growing demand have attracted a percentage of attention from the research community. However, it appears to be extra vulnerable to several attacks affecting presentation than other types of networks. The Intrusion Detection System (IDS)delivers a second line of protection against manipulation by monitoring network activity to investigate malicious attempts by attackers. Due to Manet underlying distributed architecture, traditional cryptographic systems cannot fully protect Manet from new threats and insecurities. Implementing in-depth technology for IDS can meet these challenges. In this paper, the simulation stage was developed with a NS2 simulation platform. The RNN classification algorithm was evaluated using several metrics to detect intruders. The efficiency of RNN as an approximate tool for detecting, isolating, and reconfiguring attacks was measured on datasets with different data traffic situations and dynamic patterns for manifold attacks. With a final search rate of 0.32 to 2. 35%, this feature not only provided a creative and less effective way to carry out man-in-the-middle attack (MITM) attacks on modelstages, but also made it an important factor in identifying and isolating such attacks. In addition to existing IDs, this work is intended for future generation, identification, isolation, and redesign of malicious software.

Keywords: -Mobile adhoc wireless networks (MANETs), Intrusion Detection Systems (IDSs), Recurrent Neural Network (RNN), classification algorithm and attacks.

Introduction

MANET is a gathering of independent nodes that decentralize dynamic and multi-hop radio networks and establish them cooperatively (Panos et al. 2011). There is no access point to connect to these nodes. That can then join and leave the network at any time. This brands

MANET vulnerable to many types of attacks. MANETs contains a wide diversity of mobile devices such as laptops, cell phones, PDAsthat affect computer memory, bandwidth, storage capacity, etc. The hotspot that connects the node as a central authority. It provides a lot of trust between nodes, so individual nodes depend on dynamically connecting to other authorized person. Their flexibility has led to increased use in military applications and emerging response situations. With its mission-critical information infrastructure embedded in distributed architectures, MANETs has become a prime target for many complex distributed threats, most of which target the network and interconnect layers of the protocol stack. For the reasons mentioned above, it is exact significant to implement IDS in MANET as defense. An IDS is a system that observers and detects events that occur on a PCs.

IDS includes technologies and advanced technologies for demonstrating and detecting abnormal behavior. They are trying to regulate if there is any malicious activity on the network. Typically, this is a process, device, or integration that monitors scheme and network action for malicious action. The main aim of IDS is to notice attempts before an attacker harms the network. The IDS is accountable for nursing network performance, checking network and arrangement configurations for vulnerabilities, and examining. IDS has three main functions: monitoring, detection and alarm generation. Firewalls guard the flow of information and prevent intrusion, while IDS determines whether the network is vulnerable to attack or whether a security breach has infiltrated the security of the firewall. Infiltration detection processes can be divided into two main types: abuse detection and malpractice detection. Databases containing known attack signatures are also known as abuse detection techniques and signature-based techniques. This means that the system uses examples or patterns of previously known threats and compares them to current behavior. The next most important method, which depends on the nature of the interaction differently from the typical user pattern data. Deviation detection systems detect deviations from previous operations. As a result, abnormal performance shows signs of different properties compared to normal use. Defective detection approaches in detecting new attacks never seen before in existing patterns outweigh abuse detection. However, their shortcomings are unrecognizable because high alarms occur at a high rate.

The mechanism for detecting inconsistencies and misuse was viewed from the perspective of machine learning: For example nervous system, rules of association, classification methods. Learning algorithms and artificial immune systems. Comprehensive learning tools to identify fake search engines have shown a significant increase in search rates. This is due to the field of intrusion detection, which works with special functions, which makes the machine learning approach more rigorous and rigorous.

Related works

Kurosawa et al. [1]. Dynamic learning, where training data is updated at fixed intervals, aids as the basic idea for noticing malicious activity on the network. A humble clustering procedure is used to identify malicious nodes.. Search speed and node dynamics are used to characterize performance and range from 70% to 84% for node dynamics from 0 to 20 m / s. Bose et al. [2] Bach's classification algorithms, Markov chain construction algorithms, and association rule mining algorithms were used to detect discrepancies among MAC, routing and application layers to effectively detect attackers. The global integration module received a confidence factor of 0.33% and a false positive rate (FPR) of 0.8%. Cabra [3], where a three-level hierarchical system for data modification, processing and transfer was described. A mismatch index is calculated at each level and the final decision is made in the highest hierarchy. The C4.5 decision tree was used to identify the CFA algorithm. Mitrokotsa et al. [4] The performance of five known observation classification algorithms (BMC displacement model, linear model, Gaussian mixing model, multilayer perceptron used as detection methods in detection machines for MANETS were analyzed . Their results indicated that the Naive Bias classifier performed the worst, while the perceptron multilayer classifier performed the best. Guilt, et al., [5] have suggest the five learning algorithms tested to separate common IoT packages from DOS attack packages. Algorithms: (1) nearest cadet algorithm; (2) support linear kernel vector machine (SVM); (3) Decision tree using guinea pigs; (4) Random forests using Guinea pollution levels; (5) The complete random neural network structure gave the best results in the classification tested for accuracy, responsiveness, False measures and accuracy tests.

Proposed System

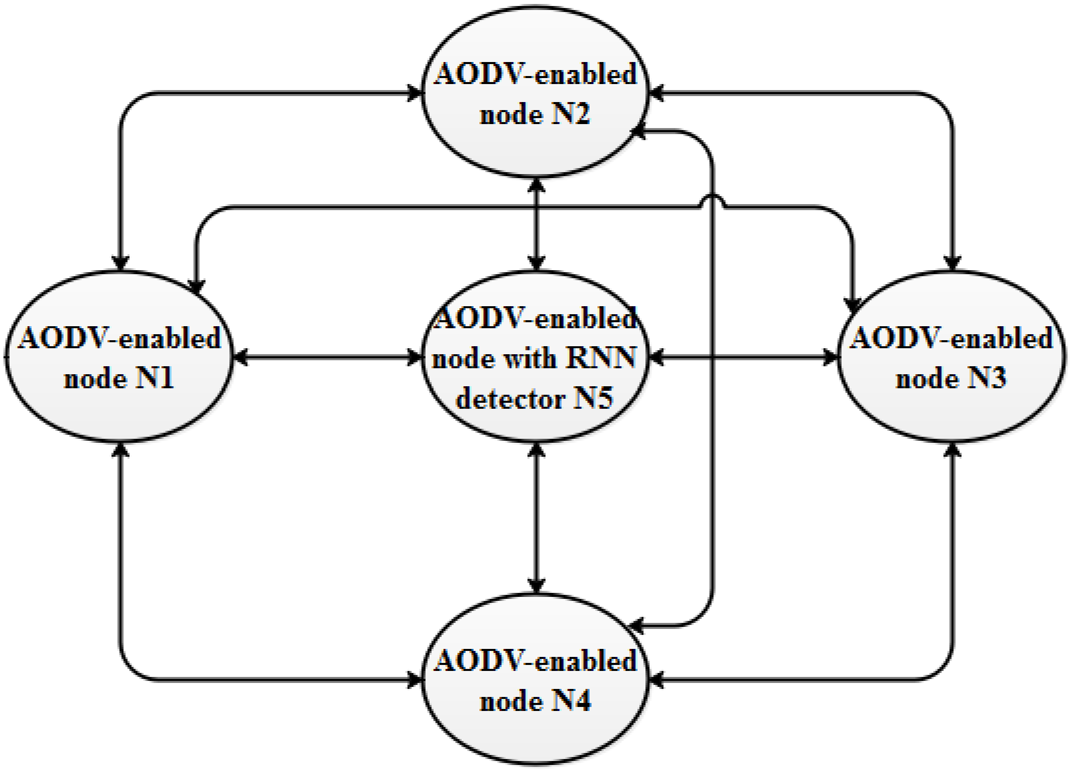


Figure 1: *General architecture of the novel working scheme.*

`A middle attack is a cyber-attack in which a third party disguises itself on one side in a two-way communiqué scene and provoke one into believing that the other is talking. In such cases, the attacker can block messages between two suspicious parties in order to gather information. Such attacks are likely in wire and wireless substructure, the latter being more vulnerable. This is due to the comparatively vague restrictions on wireless networks. Therefore, MITM attacks are a powerful way to access wireless systems that are part of MANET.

MANET uses two main routing protocols: (1) Destination Sequence Distance Vector (DSDV) and (2) Dynamic Source Routing (DSR). Many of the additional security measures discussed in the next section of this document have been implemented on MANET [6]. However, it is clear that there are no dynamic technologies available to prevent MITM attacks against MANET, especially with the AODV protocol. In the upcoming sections, various static methods are described. However, an appropriate approach is needed to counter the growing number of attacks from novel and emerging covert threats. The more diverse the method, the easier it is to manage unexpected attack vectors. Recurrent neural networks (RNNs) are well suited for solving such problems because of their universal learning capability and quality as well as their aptitude to identify information from opaque data and generate new information.

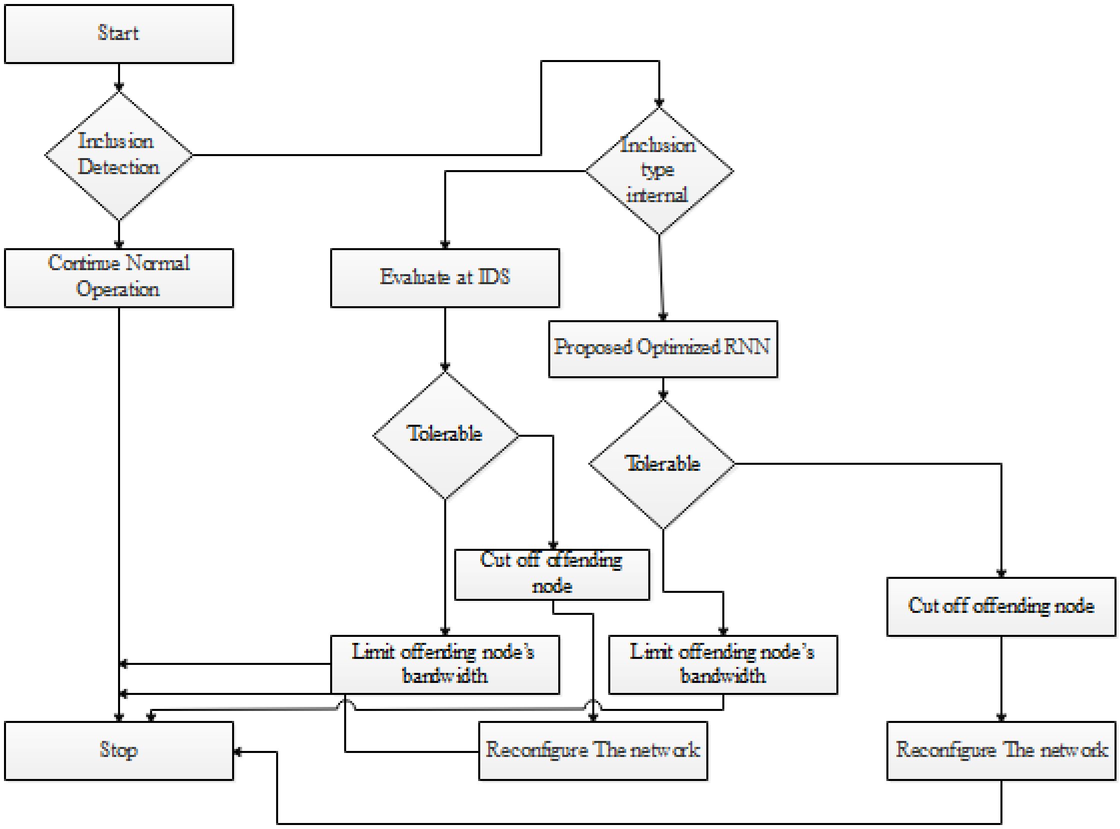


Figure 2: *Flowchart diagram of design.*

In figure 2, the blocks are properly connect to the IDS component system to create log files and detect intruders. The block diagram was obtained after realizing that installing IDS would outcome in additional resource usage (performance, processing time). In order to operate a system that can perform observable attacks to detect and retrieve malicious traffic, the process shown in Figure 4 is divided into three stages. L1 represented a phase of simulation of attack and regular interaction of network traffic. Attack detection starts from L2 and L3 provides the final step for attack recovery and reconstruction.

Implementation and testing of the research model 5.1. L1: This was done so that the system could send an attack packet already in the modeling phase. To modify the protocol as needed, we changed the forms of all AODV files in the NS-2.33535 directory to fit all codes for simulation as shown in the section. The TCL folder contains subdirectories such as Lib and Tests which contain most of the Otcal source code needed to run the simulation. All custom C ++ code and any additional changes in projects can be placed straight in the NS-2.3535 folder. From here an adapted version of the AODV protocol was installed and installed as part of the scheduler. It was used to simulate a medium range attack. The original log files have been renamed. The components of the developed package are shown in Figure 6. In addition to the Aodv_packet.h file, the line "mitm" has been added to each filename in this directory. Packet exchange between nodes was made possible using the proprietary AODV protocol and a new special version of Aodvmitm."

With the implementation of the new protocol, all classes and configurations have been renamed with the exception of the aodv_packet.h file. In addition to adding new routing protocols as needed, additional files have been added to the nspacket.tcl file in the NS-2.35535 / / TCL / Lib subdirectory. This file must be replicated each time for batch formatting. Additionally, each newly created package must be registered in this file. The file comprises a list of classes and functions that can create classes and functions directly in the coding platform CS + on NS2 OTCL / C ++, etc.

OTCL:

Criteria for Configuration STCL and OTCL were written for code simulation. "Node Replace" was configured using the "setdest" script, which contains the values specified in the first column of the initial parameters written on the files. Every simulation is performed, which contains the invading nodes in red and the original nodes in black. A simulation was run with a least of 6 nodes and aextreme of 20 nodes, and they underwent different changes during the stress test phase.

Attack Detection System.

In order to present a practicable approach to the detection of an attack in the second phase of the scheme operation, the data obtained in the attack model phase were examined with the "wrapper scheme".

The results of this phase of the analysis were then broken down into the Perl script of dataclinburter.pl, which was used to detect stage attacks and search classification functions.

Machine Learning and Feature Extraction (Wrapper Method).

Two common methods are used to select tasks: (1) a filter technique (better suitable for data mining) and (2) a wrapper technique (better suited for ML). Using the WEKA software tool, the wrapper method was found to be the best because the actual number of functions - only 26 was relatively small compared to the script obtained with the Perl script "dataklinerbutter.pl". The main problem at the time was the machine learning problem versus the machine digging problem. Thus, the Shell schemes assisted recognize features that could provide accuracy in classifying features. According to the concept, the Shell method generates all possible subsets from the feature vector and then uses a grouping algorithm to request a classification for each feature in each subset. This provides a number of features for which the classification algorithm (in this case, multi-layer perceptron) works best. However, prior to the use of the multilayer perceptron, functions were eliminated that did not add new information, which functions provided the most information before choosing the classification procedure, clustering was performed using a simple K-match-score algorithm. To simplify the essential clustering steps, manual program scripts were used with the program to detect attacks, deploy software solutions, perform required tasks, and preprocess files obtained from functional status information.

Attack Recovery System.

After identifying the functionality needed to detect a potential intrusion, the info was used to progressa Java software tool called WEKA tool. It was an RNN tool that used well-known functions to view network simulation log files, blacklist attack nodes based on information obtained from RNN classification, and reorganize the network when reporting failed nodes. It can be canceled. The most common design of a multilayer neural perspective network consists of three fully connected layers. Input layer nodes having basic information about multiple connections in a layer hidden by their individual inputs. In fact, the hidden layer and the output layer actively increase the signal flow to obtain the appropriate output. The effect of this neural network is resolute by the weight applied to the hidden and output nodes. The original RN model generated during the simulation with a learning rate of 0.3 and a pulse of 0.2 and is run at different times. The most important programming languages to implement were Perl, C / C ++, and Java, which group all the different scripts and code executable via shell scripts. A potential blacklist was generated from this log file to automatically detect attacks and configure the system.

Result and Discussion

The 5-node architecture network was modeled for a network intrusion detection system. A dedicated N5 node was used as administrator in the network, monitoring MANET for malicious nodes, removing them and reconfiguring the network. The NS2 simulator was used together with the NAM animator to develop the model. The languages such as Perl, C / C ++ and Java were used in a grouping of scripts and executable code in the Linux operating system Ubuntu 10 version. The WEKA software package for machine learning and its APIs provide the necessary resources for ANN and other ML algorithms.

The following machine learning algorithm performance measures were used to evaluate the system's performance, namely,

$$recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F - Measure = 2\frac{precision\ X\ recall}{precision + \ recall}$$

Where,

TP is the true positive: the number of positive examples as positively predicted.

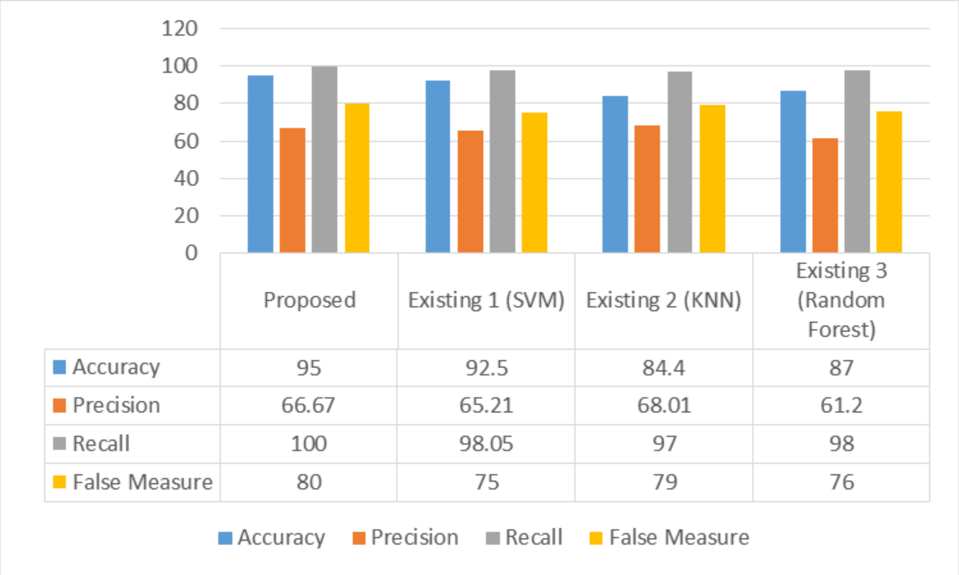
FP is the false positives: the sum of positively predicted positives, actually negative ones.

TN is the true negatives: the sum of expected negatives that are actually negative.

FN is the false negatives: the sum of predicted negative results that are actually positive.

Table 1. *Performance analysis of proposed model.*

Performance	**Proposed**	**Existing 1 (SVM)**	**Existing 2 (KNN)**	**Existing 3 (Random Forest)**
Accuracy	95	92.5	84.40	87
Precision	66.67	65.21	68.01	61.20
Recall	100	98.05	97	98
False Measure	80	75	79	76



	Proposed	Existing 1 (SVM)	Existing 2 (KNN)	Existing 3 (Random Forest)
■ Accuracy	95	92.5	84.4	87
■ Precision	66.67	65.21	68.01	61.2
■ Recall	100	98.05	97	98
■ False Measure	80	75	79	76

Figure 3. *Comparisons of performance measure of proposed with existing model*

In above table 1 and figure 3 displayed that the performance evaluation of proposed model with different classifiers. In proposed model achieved the accuracy of 95%, precision of 66.67%, recall value of 100% and false measure of 80%. However, existing methods of Support

Vector Machine (SVM) attained the accuracy of 92.5%, recall of 98.05% and K-NN classifier attained the accuracy of 84.80%, precision of 68.01% and random forest classifier attained the detection accuracy rate of 87% and false measure value of 76%. By this comparisons, we conclude that proposed model provides better accuracy than existing methods.

Conclusion

MANET is a preferred target for a variety of sophisticated circulated threats that primarily affect the network and data link levels of the protocol stack. For this reason, it is significant to use the MANET intrusion detection mechanism as a second line of protection. Although verification and encryption methods can provide security in some cases, e.g. For example, to reduce tampering, these encryption methods may not work effectively against new or stolen attacks. In this case, the proposed approach will help detect unknown intrusions. In this article, Recurrent Neural Network (RNN) classification methods for intrusion detection were developed for MANET and used to identify, blacklist, and reconfigure attacking nodes using the NS2 simulation platform. The RNN classification algorithm was evaluated using several metrics to detect intrusions. The functionality of RNN is a predictive tool for measuring, isolating, and reconfiguring attacks against datasets with different traffic situations and dynamics patterns for multiple attacks. With the amount of finding the endpoint of 2.35%, this function not only provide a more creative and costly way to carry out MITM attacks on model platforms, but also recognized time as an important factor in identifying such attacks. These attack the nodes and configurations of the network. This work should be a future invention to create, identify, separate, and reconstruct malware temporarily in addition to existing intrusion detection systems. (IDSs).

Reference

[1]. Kurosawa, S.; Nakayama, H.; Kato, N.; Jamalipour, A.; Yoshiaki, N. Detecting Blackhole Attack on Aodv-Based Mobile Ad Hoc Networks by Dynamic Learning Method. Int. J. Netw. Secur. 2007, 5, 338–346.

[2]. Bose, S.; Bharathimurugan, S.; Kannan, A. Multi-Layer Integrated Anomaly Intrusion Detection System for Mobile Adhoc Networks. In Proceedings of the IEEE International Conference on Signal Processing, Communications and Networking, Chennai, India, 22–24 February 2007; pp. 360–365.

[3]. Cabrera, J.; Gutiérrez, C.; Mehra, R. Ensemble Methods for Anomaly Detection and Distributed Intrusion Detection in Mobile Ad-Hoc Networks. Inf. Fusion 2008, 9, 96–119.

[4]. Mitrokotsa, A.; Dimitrakakis, C. Intrusion Detection in Manet Using Classification Algorithms: The Effects of Cost and Model Selection. Ad Hoc Netw. 2013, 11, 226–237.

[5]. Doshi, R.; Apthorpe, N.; Feamster, N. Machine Learning DDoS Detection for Consumer Internet of Things Devices. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 24 May 2018; pp. 29–35.

[6]. D.Anandha Kumar, S.Nyamathulla M. Kirankumar, K.Vinoth Kumar T. Jayasankar "A Hybrid Secure Aware Routing Protocol for Authentication in MANET", International Journal of Advanced Science and Technology, Vol.29, No.3, (2020), pp.8786–8794, ISSN: 2005-4238.