International Journal of Computing and Mathematics
Vol. 2, Issue. 1, March 2025, pp. 1~10

A Performance of Computational Intelligence for Security in Wireless Networks

Mohammad Nuruzzaman[1], Azham Hussain[2]
[1]School of Business, University of New South Wales, Canberra, ACT, Australia.
[2]School of Computing, Universiti Utara Malaysia, 06010 UUM, Sintok, Malaysia.

Article Info

Article history:

Received Jan 9, 2025
Revised Feb 23, 2025
Accepted Mar 11, 2025

Keywords:

Computational Intelligence
Wireless Sensor Networks (WSN)
Cyber Attacks
Network Intrusion Detection
ADFA-LD and ADFA-WD Dataset

ABSTRACT

Wireless Sensor Networks (WSNs) have been a crucial IoT development and, while strong advantages, security problems remain. New cyberattacks are growing as more computers are linked to the internet, following well-known attacks that represent serious risks to the security, credibility, and efficiency of data in WSNs. For many software and scientific questions, the implementation of intelligent computing works effectively; but the defense systems focused on computational intelligence (CI) are not being adequately examined. In this article, it examined two WSN intrusion detection evolutionary computing strategies. A neural network with backpropagation was connected with a classifier of the support vector machine. The ADFA-LD and ADFA-WD datasets were used to determine the detection rates of cyberattacks attained by the two approaches. According to the study, both strategies provide good intrusion detection solutions with an elevated true positive rate and an extremely small false-positive percent. Additionally, by demonstrating its responsibility to preserve small data sets and, consequently, a reasonable FPR ratio below the appropriate limit, it demonstrates the suitability of neural network classification techniques for identifying abnormalitiese.

Corresponding Author:

Mohammad Nuruzzaman,
School of Business, University of New South Wales, Canberra, ACT, Australia.

1. INTRODUCTION

With an anticipated billion systems made publicly available, A new era of greater mobility is about to be resulted in by the Internet of Things (IoT) [1]. The main objective of the IoT is to make widely separated computers to the cloud, thereby creating smart devices that besides human interference support data processing, storage, and sharing. To develop a dynamic network known as the IoT, such internet-connected smartphones combine traditionally related conventional smart devices. With the Internet of Things' exponential growth and sensor technology advancements, Wireless Sensor Networks (WSNs) have become an essential IoT application [2]. Such networks are made up of self-organized sensor devices connected to the network via a wireless medium being used to acquire wireless transmission abilities. They have been used in a multitude of areas including surveillance, climate change tracking, environmental monitoring, and various healthcare applications because of their low-cost design and simple implementation.

Wireless sensor networks (WSNs) [3] were data collecting and occurrence surveillance systems that act as a bridge between a computer system and a physical device. They contribute significantly to collecting environmental data in which it is not so simple to create and communicate the basic wired link to a center where it can progress more. The sensor networks are usually installed in a setting for periodic monitoring and event detection [4]. However, a WSN's nodes that collect data have less battery life, fewer storage capacity, and less computational capability, which might result in hardware malfunctions. Apart from the limited capabilities of sensor nodes, a WSN faces numerous challenges, such as sensor node deployment and configuration, mobility and topology variations, physical distribution and configuration, clustering, collection of data, safety, and client service efficiency. The protection of any communications or network system is a very critical consideration. Due to the restricted battery and memory space, but it’s more difficult and

exhausting in the sense of WSNs.WSNs just had to resolve security risks[5] like node discovery and verification, key configuration, safe routing, node authentication, group key development, and secure file collection.

There are also problems concerning protection in WSNs, considering the numerous advantages. This is mainly attributable to the intrinsic nature of becoming distributed, transmitted in general, and with minimal resources[6] in difficult unsupervised conditions. Various researches into the use of encryption, varificaiton, key protection, and protected routing in WSNs was already taken out to resolve these issues.While the potential defense measures are being established to limit cyberattacks, they haven't yet entirely removed them[7]. As such, excellently-known and recent cyber attacks cause many problems for WSNs and are also the reason for such an analysis. While data networks are used in dynamic scenarios where an intelligence-based approach works more effectively than a policy-based solution, the policy-based approach in a network draws attention to the problem in a static environment. The data allows each host to create new conditions, events, and behavior in such a way that the best solution can be calculated. Dizzy structures, neural networks made up of computers, genetic programming, crowd intelligence, and artificial antibodies are some of the most widely accepted computational intelligence (CI) components. Although CI-based privacy protocols have not been particularly well-liked for WSNs, researchers are using the majority of CI-based systems to address the difficult problems in WSNs, such as node coverage and energy management[8], integration, and optimization[9]. Different approaches, such as genetic algorithms [11], swarm ability, and artificial neural network-based WSN security protocols, are in the process of innovation since ambiguous security protocols in WSNs have been specified in [10].

Computational intelligence offers relatively low-cost IDSs advanced technology when calculating the cost of minimal use of energy. Two important computational intelligence techniques were used in this study: support vector machines (SVM) for WSN detection of breaches and reliability monitoring, and multi-layer comprehension of feed-forward backpropagation. Particularly, our investigation evaluated the detection percentages from Denial of Service (DoS) incursions derived from the three methods. Until detection rates and precision are evaluated, a standard dataset is preprocessed, normalized, and used as an output to every other network. We observed that certain strategies worked well, providing strong, true positive detection values based on our empirical test, while the SVM gave the best false positive score. This indicates that both strategies may be accustomed to build an IDS in WSNs to mitigate cyber-attacks.

The rest of this paper is structured in the following way. A similar analysis in this area is discussed in Section 2. Well-known cyber threats suffered by WSNs found in the IoT are mentioned in section 3.The proposed methodology used to include a difference between the feed-forward neural network detection rates and the support vector classifier is detailed in Section 4. In Section 5, the findings obtained are discussed. Finally, Section 6 discusses the strategies for future opportunities with a certain session.

2. RELATED WORKS

Decentralized, resilient, flexible, or without a specified system feature are characteristics of wireless sensor networks (WSNs) [12]. Such characteristics prove to be dangerous to develop and build a WSN that is protected and responding to cyber-attacks. Priority has now moved to the timely identification of threats to mitigate their effect on the network. It has been proposed that Support Vector Machines (SVMs) and Artificial Neural Networks (ANNs) offer distinct approaches to intrusion detection in WSN. In [13], a support vector machine-based hybrid intrusion detection system is presented and set up to operate in cluster-based WSNs. Using a decentralized goal function, the SVM learns and generates excellent detection accuracy with few major false positives. In [14], the developers use the Dynamically Increasing Self-Organizing Tree clustering technique to optimize performance and obtain an SVM for intrusion detection. They show that the proposed system represents an important enhancement during the training phase, exceeds the methodology of Rocchio Bundling, and gives excellent detection rates.

Different data preprocessing approaches are suggested and evaluated in [15] using different algorithms for data mining, namely SVM. In their findings, when using datasets with local features, the findings indicated that the SVM classifier provides the largest efficiency with low computation time necessary. Additionally, research is being done on the application of artificial neural networks to intrusion detection. The authors of [10] propose an artificial neural network-based intrusion detection technique and test it using 22 different attack types from the KDD99 data set. For most attacks, their findings indicate a 75 percent success rate, while attacks with fewer samples produce significantly lower values. For the first time, the WSN intrusion detection paradigm makes use of the Genetic Algorithm-Levenberg-Marquardt Algorithm [16]. It provides multilayer collective detection, a self-leaning framework with mobile robots and fuzzy computing capabilities, high detection rates, and lower energy usage than conventional BP-based models. In[17], the authors use an improved neural network with a Fuzzy Adaptive theoretical framework to classify

risks in WSNs. By learning the network is trained in a time series and identify time-related shifts, the initial system presented in [18] was improved, due to higher precision rates relative to the initial.

In terms of the datasets, problems, and methods, previous research, including (Sadotra & Sharma, Buczak & Guven) have not thoroughly checked IDSs. In this article, to solve the issues and datasets, to include a systematic and modern, wide-ranging analysis of the intrusion detection system; and also identify methodology issues and then provides proposals. Existing research papers (e.g., such as (Buczak & Guven, 2016;)) concentrate on approaches of intrusion detection or data set problem or device attack sort and prevention of IDS. Intrusion prevention, dataset questions, avoidance methods, and various forms of attacks have not been extensively analyzed in the papers. Besides that, the evolution of intrusion detection technologies is such which, in the meanwhile, many new solutions were developed, and there is a requirement for an up-to-date framework.

3. CYBER ATTACKS IN WSN'S

Even though WSNs are the target of many cyberattacks, these attacks are often divided into active and passive categories. threats. Passive attacks don't really impact a network or modify data generally, but instead track failure targets or retrieve network traffic. By comparison, active attacks aim to modify data on or relatively close to a destination or use different types of denial of service (DoS) attacks to refuse network access. Established active cyber threats targeting WSNs, being used to modify data, or execute DoS attacks, would be described in this section. Because of their intrinsic limits in communication bandwidth, the WSNs have numerous security problems. A complete secure sensor network incorporates preventive measures including node authentication, secure routing, safe key management, and lightweight encryption schemes must be implemented, but these issues have been partially addressed by advancements in hardware and software [19, 20]. Some common WSN attacks are as follows:

- **Spoofed Attack:** An attacker can create routing information, issue a bogus error code, or modify the packet distribution channels in this kind of routing replay or attack alteration.
- **Selective Forwarding Attack:** This is a certain kind of hole attack in which an opponent receives the network communication authority and may fall the sequence number or keep refusing to send any specific packet to the next one.
- **Sinkhole Attack:** Here, the adversary identifies an artificially high standard path on which other nodes begin routing. All nodes begin routing via an intermediate node due to this negative output.
- **Sybil Attack:** An opponent introduces his identification in various ways in this attack such that it behaves and acts as multiple different nodes. The adversary node interrupts the conversations or could monitor the network traffic and is accessible over the internet.
- **Wormhole Attack:** An adversary first captures the packets in this type of attack, tunnels them to a different location, and then replays those frames into networks. Even without knowing the secret key, the hacker will send the authentication transfers in order to get access through wormhole assaults.
- **Hello Flood Attack:** To find network k nodes, a sensor nodes requires the Hello keyword. An attacker uses the same approach by trying to send the fake email of Hello to a node density. In this sort of attack, the attacker aims to destroy any neighboring server.
- **Denial of Service (DoS):** Attack In this form of attack, an opponent transfers several fake transmissions to a single network to generate a network message flood. The DoS assault causes a scarcity of all limitations for WSNs with restricted batteries and storage space and renders the device inactive or out-of-order.
- **Acknowledge Spoofing Attack:** In this type of attack, an adversary acts as a man-in-the-middle and provides the sender with the bogus identity response on top of a real receiver node.
- **Misdirection Attack:** To redirect internet traffic to its final destination, a corrupted node sends communications along inappropriate paths.
- **Desynchronization Attack:** Sequence number packets are interrupted, causing end nodes to assume that certain samples are being skipped and then order retransmission.
- **SYN Flood Attack:** T The SYN Flood Attack happens when a network is overloaded with malicious SYN request packets, resulting in a significant number of node-to-node partially-open state relationships. In the absence of required ACK reply packets, denial of service is carried out and the nodes' services are used up.
- **Collision Attack:** As other nodes are still communicating, a corrupted sensor forwards small noise packets, triggering a collision with a random network.
- **Exhaustion Attack:**A infected system receives an RTS message frequently, achieving a CTS response, which will ultimately drain both node's resouces is performed constantly.
- **Unfairness Attack:** Infected nodes effectively control network connectivity, minimizing actual data transfer window time. As a consequence, while access is not completely denied, access is greatly reduced.

- **Tampering Attack:** To keep sensor nodes alive and keep them from going into sleep mode, malicious detectors are regularly transmitted to them.
- **Battery Exhaustion Attack:** According to sleep mode protection, node resources are exhausted.

Build methods and resources to enable distributed signal processing, safe data processing, networking, data transformation and control, and program growth are the problems facing neural activity in WSNs. The resilient solutions for the attacks described below are the key management and stable routing protocols. The study about a full protected sensor network, nevertheless, is still in its stage of growth. The battery capacity, memory space, and the expense of the sensor network should be treated correctly by the researchers to develop a new protection system. Such dimensions are leading researchers to build a new framework that will be sufficiently safe and effective to deploysensor nodes.

4. PROPOSED WORK

Computational Intelligence (CI) is an analysis of various processes in dynamic and evolving conditions to allow or promote intelligent behavior. It is a subcategory of Artificial Intelligence (AI) that works based on certain frameworks of AI which demonstrate a capacity to recognize, adjust, generalize, abstract, and explore new situations.

A single defensive mechanism is hardly realistically nor possible toward cyber threats. Thus, the second line of protection like intrusion prevention should be complemented with first-line security measures like encryption, authentication, and authorization.Interference could be categorized below as any collection of actions that aim to reach a resource's credibility, secrecy, or accessibility. An intrusion detection system (IDS) addresses this by minimizing unwanted data, improving data integrity, making data inaccessible to unauthorized people or machines, ensuring that data is protected in terms of cost effectiveness, quality, and intended use, and assisting in making sure that unauthorized people grant network access and system access. Two different approaches may identify intrusion detection schemes, with a third hybrid solution still active, that would be mentioned below.

4.1 Signature Intrusion Detection Systems (SIDS)

Often referred to as knowledge-based identification or abuse detection, signature intrusion detection systems (SIDS) rely on structure matching techniques to cl asify a recognized attack. SIDS uses corresponding techniques to find a previous incursion. To put it another way, when an intrusion signature in a signature database surpasses the signature of a previous intrusion, a warning alert is triggered. Host logs for SIDS are reviewed to identify sets of signals or behaviors that have already been historically recognized as threats. For generally accepted intrusions, SIDS typically has superior identification, precision. However, SIDS has trouble detecting cloud assaults since each matching signature stays in the database until the most recent attack signature is received and gathered. SIDS is utilized in certain statistical techniques, like Snort. Network packets are analyzed by standard SIDS techniques, which then try to fit them with a signature index. However, these methods are unable to identify attacks that span numerous packets. Given how sophisticated the present infection was, it might be fair to gather signature data from many packets. This involves the IDS remembering the properties of earlier packets. Specifically, there are several methods for creating a signature for SIDS, such as using software modules, appropriate language string patterns, or semantic criteria [21–22]. Since there was no prior signature for zero-day attacks, their increasing frequency has made SIDS techniques more potent. Polymorphic malware and the increasing number of targeted attacks could further undermine the efficacy of this basic architecture. A possible approach to this issue would be to provide AIDS methods, that, as stated in the next section, function by analyzing what is an acceptable activity instead of what is anomalous.

4.2 Anomaly Intrusion Detection Systems (AIDS)

With AIDS, a standard simulation of a computer system's behavior is created using machine learning, statistical, or knowledge-based techniques. Every significant difference between the model and the activity that was estimated is referred to as an exception, and this could be seen as interference. It is assumed that harmful activity differs from typical user behavior for this class of techniques. Intrusions are actions by abnormal users that differ from normal behavior. The planning phase and the research phase are the two phases of AIDS development. A new data set is employed in the testing phase to evaluate the device's capacity to validate previously unidentified incursions, while the normal traffic pattern is utilized in the training phase to create a framework of typical activity. Depending on the training method, AIDS could be classified into several categories, including mathematical, knowledge-based, and machine learning-based [23]. Since AIDS doesn't rely on a signature database to identify questionable user behavior, its primary advantage is its capacity to identify zero-day attacks [24]. AIDS triggers a risk signal when the conduct under

investigation deviates from typical behavior. AIDS, on the other hand, has unique advantages. They will then be capable of identifying hostile information systems. An alarm is set off when an attacker starts making purchases in a compromised system that are not recognized by the typical user behavior. Furthermore, as the framework is built from personalized accounts, it is very challenging for a cyber-criminal to know what is a common user activity without triggering an alarm. Only known intrusions can be detected by SIDS, while AIDS may recognize zero-day attacks. However, AIDS may lead to a high false-positive rate since abnormalities may be nothing more than new habitual habits rather than actual intrusions.

4.3 Computational Intelligence Techniques

Computational Intelligence (CI) is a subcategory of Artificial Intelligence (AI) that works based on certain frameworks of AI which demonstrate a capacity to recognize, adjust, generalize, abstract, and explore new situations. A popular machine learning tool for identification based on anomalies is the Artificial Neural Network (ANN) [25]. To approximate or forecast outcomes from supplied input patterns, interacting neurons share knowledge. They are often organized into layers, each of which presently consists of a specific number of linked neurons. As shown in Figure 1, data patterns are provided at the input layer, processed at the hidden layer, and then, using a weighted correlation method, a response is provided at the output layer.

The arrows show interactions between neurons and the direction of movement of information is often suggested. The signal between the two neurons is determined by a weight in each relation. If the network output may not exceed the target output, efficiency can be increased by sequentially improving the weights once an acceptable precision is reached or till now no learning changes would be made. In feed-forward networks, input-to-output communication is unidirectional. The performance of any layer doesn't influence the same layer, as feedback mechanisms do not occur. Feed-forward networks are extensively utilized in pattern formation, detection, and classification. There are already feedback pathways that allow knowledge to travel both ways, and feedback networks come in different forms. Each neuron's input can still be altered, suggesting that the network's state is always changing and evolving. For image captioning, voice recognition, and action identification, feedback networks are widely used.

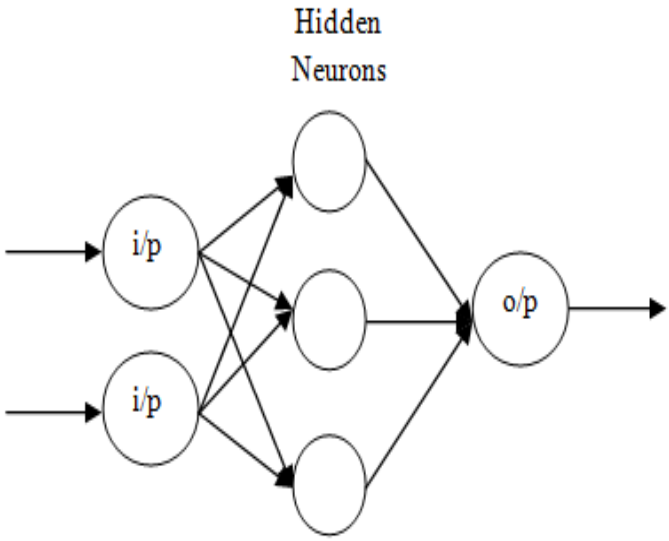

Figure 1. Basic Structure of Neural Networks

Both styles of NN frameworks transmit but doing it separately, a desire to understand. In feed-forward networks, supervised learning is used to teach the network to incorporate specific possible outputs by providing it with the necessary response to each input vector. As a result, the input network clusters the data according to the relationships among the data elements using unsupervised learning, in which the ideal result is simply not provided.

4.4 Support Vector Machine (SVM)

The SVM is another overseen machine learning method that can be used as an extra remedy for anomaly-based detection [26]. A SVM controlled machine learning approach is commonly used to study a variety of optimization issues; nonetheless, it supports a number of numerical and categorical data and aids both classification techniques. In this machine learning system, the method solves a classification model by

dividing cases from different class labels using a hyperplane in a high-dimensional vector. The next stage in the classification process is to determine which high-plane best separates between both groups. By identifying the hyper-plane, which properly separates the two classes, differentiation is then carried out. An element dividing a set of characteristics that have distinct classes in the class is a decision layer.SVM uses a series of protocols for supervised learning.By optimizing the boundary, the actual limitation on the expected error rate is minimized to produce the largest distance between the hyperplane dividing and the anomalies on each side of it [27]. The Support Vector Machine (SVM)'s basic setup is shown in Figure [1] [28].

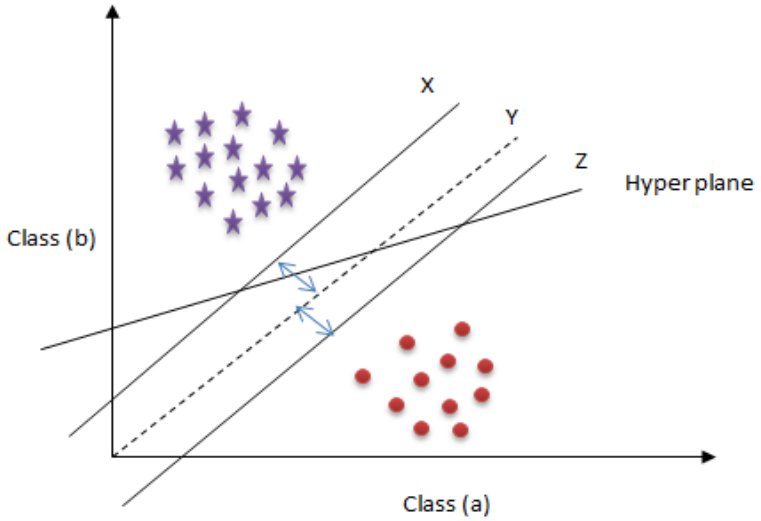

Figure 2.Structure of Support Vector Machine (SVM)

The category used to make boundary selection decisions is the basic SVM model. There should be 2 categories afterward, and high performance represents one classifier class and another classifier class is represented by 1. If for a two-class problem, the m-dimensional x interactions could be equally eliminated, the entire class is differentiated by the understanding of the given formulas.

$S(X) = b^n x + k$ (1)

From the Eq. (2), where x is a hyperplane value, v is a vector in the SVM function space and k is a bias. To maximize the support vectors between the two classification terms in Equation (3), each one will be minimized. The remaining inequalities will be provided to another training set for the effective analysis of every sample.

$\frac{1}{2} |b|^2$ min (2)

For both forms, it was possible to decrypt a simple hyperplane as $bx_n + k \geq 1$, Class $C_i = 1$, and $bx_n + k \leq 1$ for class $E_n = -1$.Some might integrate equations to provide the below equations.

$E_n(b^n x + k) \geq 1$, for training samples (3)

5. SIMULATION RESULTS

In this article, anomaly-based identification in wireless sensor networks was selected as a form of intrusion detection. To determine and examine detection levels, a Support Vector Machine (SVM) Classifier will be used to assess a Multilayer Feed Forward Neural Network (Multi-FNN).

5.1 ADFA-LD and ADFA-WD Dataset Description

Two datasets (ADFA-LD and ADFA-WD) were created by researchers at the Australian Defense Force Academy as standard datasets that depict the design and execution of modern attacks (Creech, 2014). Both Linux and Windows OS documents are included in the data sets; they are generated from framework call-based HIDS research. To create ADFA-LD, a host machine running Linux was utilized (Creech & Hu, 2014b). Since the new zero-day code is used in some of the ADFA-LD attack scenarios, this database is ideal for illustrating the differences between SIDS and AIDS intrusion detection systems. It requires 3 different data categories, each data type containing the device's direct call records. Any other research dataset was acquired from the host for taking input, with user activities linked to web browsing to the paper planning for LATEX. Table 1 displays a couple of the ADFA-LD functions with the shape and the description of each

function.ADFA-LD also contains device request traces of multiple known attacks.For analyzing HIDS, the ADFA Windows Dataset (ADFA-WD) offers a modern Windows dataset.

Table 1.Classes of ADFA-LD Attack

Attack	Payload	Count
Adduser	Add new superuser	91
Hydra_FTP	Password brute force	162
Hydra_SSH	Password brute force	176
Java_Meterpreter	Java-based Meterpreter	124
Meterpreter	Linux Meterpreter Payload	75
Web_Shell	C100 Webshell	118

Table 2. ADFA-LD Dataset Functionality [29]

Name	Type	Description
srcip	nominal	Source IP address
sport	integer	Source port number
dstip	nominal	Destination IP address
dsport	integer	Destination port number
proto	nominal	Transaction protocol
state	nominal	Indicates to the state and its dependent protocol
dur	Float	Record total duration
sbytes	Integer	Source to destination transaction bytes
dbytes	Integer	Destination to source transaction bytes
sttl	Integer	Source to destination time to live value
dttl	Integer	Destination to source time to live value
sloss	Integer	Source packets retransmitted or dropped
dloss	Integer	Destination packets retransmitted or dropped
service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
Sload	Float	Source bits per second
Dload	Float	Destination bits per second
Spkts	integer	Source to destination packet count
Dpkts	integer	Destination to source packet count
swin	integer	Source TCP window advertisement value
dwin	integer	Destination TCP window advertisement value
stcpb	integer	Source TCP base sequence number
dtcpb	integer	Destination TCP base sequence number
smeansz	integer	Mean of the how packet size transmitted by the src
dmeansz	integer	Mean of the how packet size transmitted by the dst
trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
res_bdv_len	integer	Actual uncompressed content size of the data transferred from the server's http service.

To identify the two experimental approaches in this analysis, it gathered True Positive Rate(TPR) and False Positive Rate (FPR) statistical analysis of all attacks involved.

True Positive Rate (TPR)

The TPR may be determined using True Positive (TP), which indicates the amount of attacks that were detected, and False Negative (FN), which indicates the number of attempts that were not found.

$$\text{TPR} = \frac{TP}{TP+FN}$$

False Positive Rate (FPR)

The FPR could be determined using the number denoted by False Positive (FP). Attacks that were simply normal traffic were seen, however True Negative (TN) indicates when assaults fail to happen and are therefore never observed.

$$\text{FPR} = \frac{FP}{FP+TN}$$

False Negative Rate (FNR)

False-negative refers that an anomaly is not detected and labeled as normal by a detector.It is analytically necessary to describe the FNR as:

$$\text{FNR} = \frac{FN}{FN+TP}$$

Class examples in the training data set are also divided into two components at a ratio of 80 percent to 20 percent for classifier improvement.For training the classifier, the wider split is being used and the shorter split is used to verify its output. The FNN technique applied with n = 5 in Weka, which is also empirically assessed through an exploratory search, is trained as a classification model. The classifier's efficiency is measured for both multi and binary decision problems in the field of normal metrics such as

TPR, FPR, TNR, FNR, Accuracy, F-Measure, and ROC.For the binary category, two names, respectively "Normal" and "Attack" are used.The multi-class issues include the "Normal" category and six attack groups.The output metric functions and confusion matrix are provided in Tables 3 and 4 for the FNN classifier implemented to the binary class query, accordingly.The TPR for the regular class is 1.0 in Table 2, showing that all regular traces are observed.For the attack class, the TPR is 0.95, which means that 5625 out of 5750 attack traces are detected: only 75 traces are misclassified out of 14629 regular traces and attack traces.This model's average accuracy is 99.51 percent.

Table 3.Performance evaluation for Feedforward Neural Network Classifier

Class	TPR	FPR	TNR	FNR	Precision	F-Measure	ROC
Normal	0.99	0.005	0.98	0.001	0.97	0.99	0.99
Adduser	0.96	0.001	0.99	0.000	0.98	0.97	0.94
Hydra_FTP	0.91	0.025	0.97	0.005	0.99	0.95	0.92
Hydra_SSH	0.97	0.000	0.98	0.007	0.99	0.99	0.99
Java_Meterpreter	0.90	0.001	0.98	0.001	0.97	1.00	0.97
Meterpreter	0.95	0.001	0.98	0.001	0.99	0.95	0.96
Web_Shell	0.93	0.001	0.99	0.001	0.96	0.97	0.98
Average	0.96	0.045	0.98	0.005	0.96	0.97	0.97

Table 4.Confusion Matrix

Normal	Adduser	Hydra_FTP	Hydra_SSH	Java_Meterpreter	Meterpreter	Web_Shell
9005	1	0	0	0	3	0
8	750	1	0	0	0	0
3	0	899	1	0	0	0
1	0	0	1034	1	1	0
0	0	1	3	895	0	1
4	0	0	0	0	597	0
5	1	0	0	5	1	1056

Tables 5 and 6 represent the performance assessment effects on the SVM Classifier for multi-class models, respectively.Table 5 describes the results of the successful assessment of those columns, and table 5 displays the confusion matrix for the same model. In terms of their class memberships, knowledge in those tables provides highly reliable accuracy in detecting traces. 50 signatures are mis-categorized for the multi-class mode. All of the 190325 common traces, 1210 meterpreter attacks, and 1983 web-shell attacks are listed accurately in this method, based on table 6.In comparison, 30 attack traces are misclassified as "Natural," where 20 attack traces are identified as referring to some other type of attack. The precision for both models combined is 98.9 percent or better. This means that, in comparison with SVM, the Eigentraces are such an effective classification technique for the ADFA-LD framework for misuse identification in file system call traces.

Table 5. SVM Classifier Performance

Class	TPR	FPR	TNR	FNR	Precision	F-Measure	ROC
Normal	0.99	0.004	0.97	0.002	0.98	0.98	0.99
Adduser	0.98	0.001	0.92	0.001	0.99	0.95	0.98
Hydra_FTP	0.93	0.000	0.99	0.000	0.96	0.97	0.91
Hydra_SSH	0.92	0.001	0.96	0.000	0.99	0.98	0.90
Java_Meterpreter	0.90	0.003	0.97	0.006	0.99	0.99	0.98
Meterpreter	0.99	0.001	0.99	0.002	0.95	0.92	0.99
Web_Shell	0.99	0.001	0.99	0.000	0.98	0.99	0.99
Average	0.98	0.005	0.98	0.001	0.99	0.98	0.98

Table 6.Confusion Matrix

Normal	Adduser	Hydra_FTP	Hydra_SSH	Java_Meterpreter	Meterpreter	Web_Shell
190325	1	0	0	0	3	0
8	1550	1	0	0	0	0
3	0	1943	1	0	0	0

1	0	0	2692	1	1	0
0	0	1	3	2837	0	1
4	0	0	0	0	1210	0
5	1	0	0	5	1	1983

6. CONCLUSION

The computer intelligence methods for intrusion detection in WSNs were suggested in this article. The proposed framework includes the Eigensystem for object detection and pattern formation, and the ADFA-LD information gathering class label for FNN and SVM. The ADFA-LD dataset uses seven different forms, one for regular attacks and six for specific assaults. The data is a list of device calls that were gathered from the Windows system. Adduser, Hydra FTP, Hydra SSH, Java Meterpreter, Meterpreter, and Web Shell are used in the attacks. For two separate approaches, the IDS performance was measured. For every scenario, two classes have been created: one for the conventional tracks and another for all six attacks combined. All assaults and even the typical class features were employed in both the training and testing datasets of the original ADFA-LD datasets, which were divided into two categories: one for training and one for testing. Via a simulation analysis, IDS efficiency was assessed. Our investigations showed that both computational intelligence approaches had the ability to actively detect intrusions. The help vector machine also proved its viability for recognizing anomalies by employing a restricted representative sample, which enabled it to keep an FPR level marginally below the 1 percent threshold, even though both approaches yielded excellent and comparable FPR values. This could suggest that the help vector machine was the better anomaly detection method for the dataset used (ADFA-LD).

REFERENCES

[1] McDermott, C. D., & Petrovski, A. (2017). Investigation of computational intelligence techniques for intrusion detection in wireless sensor networks. International journal of computer networks and communications, 9(4).

[2] Yinbiao, S., & Lee, K. (2014). Internet of Things: Wireless Sensor Networks Executive summary.

[3] Ranjan, P., & Om, H. (2017). Computational Intelligence Based Security in Wireless Sensor Networks: Technologies and Design Challenges. In Computational Intelligence in Wireless Sensor Networks (pp. 131-151). Springer, Cham.

[4] Kulkarni, R. V., Förster, A., & Venayagamoorthy, G. K. (2010). Computational intelligence in wireless sensor networks: A survey. IEEE communications surveys & tutorials, 13(1), 68-96.

[5] Kifayat, K., Merabti, M., Shi, Q., & Llewellyn-Jones, D. (2010). Security in wireless sensor networks. In Handbook of information and communication security (pp. 513-552). Springer, Berlin, Heidelberg.

[6] Chen, X., Makki, K., Yen, K., & Pissinou, N. (2009). Sensor network security: a survey. IEEE Communications Surveys & Tutorials, 11(2), 52-73.

[7] Ping, Y., Xinghao, J., Yue, W., & Ning, L. (2008). Distributed intrusion detection for mobile ad hoc networks. Journal of systems engineering and electronics, 19(4), 851-859.

[8] Jia, J., Chen, J., Chang, G., & Tan, Z. (2009). Energy-efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. Computers & Mathematics with Applications, 57(11-12), 1756-1766.

[9] Guo, H., Low, K. S., & Nguyen, H. A. (2009). Optimizing the localization of a wireless sensor network in real-time based on a low-cost microcontroller. IEEE transactions on industrial electronics, 58(3), 741-749.

[10] Xia, F., Zhao, W., Sun, Y., & Tian, Y. C. (2007). Fuzzy logic control based QoS management in wireless sensor/actuator networks. Sensors, 7(12), 3179-3191.

[11] Biswas, K., Muthukkumarasamy, V., & Singh, K. (2014). An encryption scheme using a chaotic map and genetic operations for wireless sensor networks. IEEE Sensors Journal, 15(5), 2801-2809.

[12] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications Magazine, 40(8), 102-114.

[13] Sedjelmaci, H., & Feham, M. (2011). Novel hybrid intrusion detection system for clustered wireless sensor networks. arXiv preprint arXiv:1108.2656.

[14] Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB Journal, 16(4), 507-521.

[15] Sahu, S. K., Sarangi, S., & Jena, S. K. (2014, February). Detail analysis of intrusion detection datasets. In 2014 ieee international advanced computing conference (IACC) (pp. 1348-1353). IEEE.

[16] Lu, F., & Wang, L. (2014). The intrusion detection system is based on the integration of neural networks for wireless sensor networks. J. Softw. Eng, 8, 225-238.

[17] Li, Y., & Parker, L. E. (2008, April). Intruder detection using a wireless sensor network with an intelligent mobile robot response. In IEEE SoutheastCon 2008 (pp. 37-42). IEEE.

[18] Kulakov, A., & Davcev, D. (2005, April). Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms. In International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II (Vol. 2, pp. 534-539). IEEE.

[19] Boukerche, A., Nakamura, E. F., & Loureiro, A. A. (2008). Algorithms for Wireless Sensor Networks: Present and Future.

[20] Martins, D., & Guyennet, H. (2010, September). Wireless sensor network attacks and security mechanisms: A short survey. In 2010 13th International Conference on Network-Based Information Systems (pp. 313-320). IEEE.

[21] Meiners, C. R., Patel, J., Norige, E., Torng, E., & Liu, A. X. (2010, August). Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems. In Proceedings of the 19th USENIX conference on Security (pp. 8-8).

[22] Lin, P. C., Lin, Y. D., & Lai, Y. C. (2010). A hybrid algorithm of backward hashing and automaton tracking for virus scanning. IEEE transactions on computers, 60(4), 594-601.

[23] Butun, I., Morgera, S. D., & Sankar, R. (2013). A survey of intrusion detection systems in wireless sensor networks. IEEE communications surveys & tutorials, 16(1), 266-282.

[24] Alazab, A., Hobbs, M., Abawajy, J., & Alazab, M. (2012, October). Using feature selection for the intrusion detection system. In 2012 international symposium on communications and information technologies (ISCIT) (pp. 296-301). IEEE.

[25] Petrovski, A., Rattadilok, P., & Petrovski, S. (2015, September). Designing a context-aware cyber-physical system for detecting security threats in motor vehicles. In Proceedings of the 8th International Conference on Security of Information and Networks (pp. 267-270).

[26] Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB Journal, 16(4), 507-521.

[27] Chen, Y., Zhao, J., & Li, F. (2018, December). An SVM-Based Recognition Method for Safety Monitoring Signals of Oil and Gas Pipeline. In IOP Conference Series: Materials Science and Engineering (Vol. 452, No. 3, p. 032008). IOP Publishing.

[28] S.V Manikanthan, T Padmapriya, Azham Hussain, E Thamizharasi. (2020). "Artificial Intelligence Techniques for Enhancing Smartphone Application Development on Mobile Computing", International Journal of Interactive Mobile Technologies (iJIM), vol. 14, no. 7.

[29] Creech, G. (2014). Developing a high-accuracy cross-platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks (Doctoral dissertation, University of New South Wales, Canberra, Australia).