www.ijarp.com ISSN 2456-9992 Page: 01-39

International Journal Advanced Research Publications

Research Article

Volume: 01
Issue: 03

A DEEP LEARNING BASED INTRUSION DETECTION MODEL IN INDUSTRIAL INTERNET OF THINGS NETWORK WITH RECURSIVE FEATURE ELIMINATION

***Fagbohunmi Grifin Siji**

Computer Engineering Department, Abia State University, Uturu Abia State Nigeriia.

Article Received: 20 November 2025, Article Revised: 10 December 2025, Published on: 30 December 2025

***Corresponding Author: Fagbohunmi Grifin Siji**

Computer Engineering Department, Abia State University, Uturu Abia State Nigeriia.

DOI : https://doi-doi.org/101555/ijarp.9572

ABSTRACT

Industrial Internet of things can be defined as a technology that enables communication between digital appliances and industrial system. It is a new research field which deals with the network of several IoT devices and industrial operations. IIoT devices are used to accumulate huge volumes of data from several sensors embedded in them. However, this technology has a major setback, namely cyberattacks which prevent the IIoT devices from providing reliable data for industrial operations. These risks have resulted in loss of finance and reputation issues to several organizations, some of which have resulted the shutdown of some organizations and theft of sensitive and classified information. In relation to this, many Network Intrusion Detection System (NIDS) have been designed to mitigate these shortcomings. The amount of information required to design a secure NIDS is cumbersome thereby making the detection of current and yet to be identified intrusion a very difficult task. The aim of this paper is to design a deep learning based intrusion detection system for IIoT systems with recursive feature elimination. The feature selection is aimed at training and checking the information obtained from TCP/IP packets from the internet. The model uses unsupervised learning technique and recursive feature elimination with deep feedback neural network. The design was tested using NSL-KDD and the UNSW-NBLS datasets as well as on a hardware testbed. The technique proposed in this paper outperforms other state of the art Network Intrusion detection Systems (NIDS) in terms of accuracy, scalability, rate of detection and IPR by 99.1%, 2.3 %, 99.3% and 1.5% respectively for NSL-KDD dataset and 99%, 1.9%, 99.9% and 1.7% respectively for UNSW-NB15 dataset. Also simulation experiment was performed which validates the appropriateness of the technique proposed in

this paper for both IIoT intrusion detection system and classification of intrusion network attack.

KEYWORDS: Industrial Internet of things, NSL-KDD, UNSW-NB15, Intrusion detection system,

1. INTRODUCTION

The acronym IIoT stands for Industrial Internet of Things, it comprises of embedded devices which communicate by means of their embedded sensors and industrial operations for the accomplishment of organizational and industrial goals. IIoT communications are used in data mining and in the management of real world applications (Ambika P 2021). The application of IIoTs in industrial processes allow information to be computed from several data samples for the development of Multiple Input and Multiple output (MIMO) systems. The use of these multiple IIoT devices allow for a more accurate decision making in industrial operations. The synergy between all these IIoT devices also allow for improved efficiency in industrial operations and other types of corporate organizations, thereby enabling improved quality of life. The ability to process huge volumes of data from the many embedded sensors in IIoT systems make prompt decision making possible in various stages of industrial processing due to its distributed nature. The applications of these IIoT systems include healthcare systems, retail, automobile and transportation systems. The application of IIoT can also enhance production efficiency, productivity and improved operational efficiency. It should be noted that IIoT system in its initial stage is used to develop various processes and facilities required for industrial production, while the end product is aimed at improved production efficiency. Innovations in IIoT will result in a more optimized production models for operational efficiency with improved quality of goods and services. The application of machine learning algorithms with special focus on deep learning on IIoT network will lead to an improvement in production reliability leading to better satisfaction for customers.

IIoT technology require various components before it can be integrated into an existing industrial operation required to produce intelligent integration of automated systems (Ashima et al 2022). Machine learning has been applied to smart workplace, automation of cognitive systems and exploration of smart data, just to mention but a few in industrial processing. IIoT is made up of devices in which microcontrollers are embedded for the attainment of certain goals such as monitoring and control in industrial processing. The design of IIoT networks require certain datasets which would be used to present data to the system for system testing

and maintenance. An important feature of any IIoT system is that it must be regularly updated in response to changes to the data received by its sensors. It must also be intelligent to apply machine learning techniques to take appropriate decisions at all times in response to environmental changes. It should be realized that IIoT technology has revolutionized how companies communicate online and also how the various stages in industrial operations are optimized for operational efficiency. IIoT devices are able to communicate with the cloud thereby enabling limitless operational variability. The interaction between the IIoT devices and the cloud is very important in the industrial and institutional economies nowadays (Sherasiya et al 2017). In view of these, IIoT are usually made up of different types of devices, control programs and sensors that enable comparison between the real and virtual world (Adeniyi et al 2022), As a result of the relation between information technology (IT) and organizational technology (OT), a system that is made up of a single embedded system is susceptible to attack as its characteristics can easily be cloned (Amit and Chinmay 2022), (Ayo et al 2021). . A connection between machine-to-machine and machine-to-person in a network is done with the aid of TCP/IP interface in IIoT network by applying different communication protocols (Abdulraheem et al 2022). The number of IIoT systems that can be compromised by some sophisticated attack techniques have increased in recent times. These attacks aim at getting vital information that can corrupt the network's system and lead to severe loss of funds for an establishment (Muna et al 2019). It is therefore important that a cybersecurity mechanism be put in place to mitigate existing and yet to be identified attacks to the IIoT system. The billions of naira lost to fraudsters recently in Nigeria through online bank information hacking has made the research into cybersecurity very critical in recent times. It should also be noted that the price of IIoT devices and support infrastructure is increasing astronomically, no thanks to the present economic realities in the country.

The most common threat to the IIoT network is malware that compromise on-the-spot (zero-day) vulnerabilities. In these type of attacks, the hackers corrupt vulnerable computers in order to monitor and change their operations by employing different methods such as Progressive Determined Risk (PDR), Denial of Service (DoS) and Decentralized DOS (DDOS). An example was the 2024 Stuxnet worm that attacked some Nigerian banks. This prompted most banks in Nigeria then to update their operations, leading to several down time of services resulting in frustrations to their customers. Hackers were able to intrude into the ICS of a dam in New York in 2013, while also the black energy passive arrack resulted in the shutdown of about 80,000 power stations in Ukraine (Dash et al 2023). The success of these

attacks show that the normal cyber-threat techniques such as security protocols, cryptography, access control, biometrics and the Interruption Discovery System (IDS) are not adequate for a secure IIoT network. It is therefore very imperative that a network intrusion detection system be developed to protect IIoT networks from all categories of threats. IIoT currently use its many embedded sensors to acquire large quantities of data from the environment which is then used to generate information knowledgebase in Industrial operations for network security (Kushner 2015).

In order to prevent a network workstation from multiple grid invasions, a network intrusion system (NIDS) is designed to detect abnormal traffic in a network. (Khan and Byun 2022). the term intrusion refers to a method wherein an attempt is made to compromise an IIoT network's security. There has been a drive to develop new IDS to mitigate the threats caused by invasive frameworks.

There are different types of intrusion detection system (IDS) that have been designed by researchers in the past. However these IDS systems are vulnerable to different categories of attacks. There are increased research on anomaly detection as a result of IDS not been able to detect and forecast compromising attributes of yet to be identified threats. Current machine learning methods of anomaly detection still have a high false alarm rate (Yan and Yu 2018). In recent times feature extraction technique is being used in the design of IDS system (Enache and Sgârciu 2017). In the most recent feature selection technique, a fitness value is used to classify input attributes to the system model, where fitness values that are approximately the same are aggregated and stored in a cluster so as to minimize error in the NIDS (Ayo et al 2021).

The vectors that make up each classifier features are large in size and they will not all apply to the groups being categorized, therefore certain techniques of feature selection must be applied. Approaches for feature selection are made up of three categories, these include, filter approach, wrapper approach and the embedded approach (Zhang et al 2018). Amongst the three categories, the most widely used feature selection technique usually converges on the best fitted functionality which is dependent on the measurement of datasets devoid of the performance of each classifier. The most superior feature selection technique is the wrapper method, this is because the quality of the feature subclass is evaluated by the classifier feedback. This makes the wrapper method of feature selection able to predict more accurately the secured nature of an IIoT device communication. The integrated process is similar to the

wrapper method because the classifier is based on an embedded process function that models the system used to enhance the search efficiency from the learning algorithm.

There are different categories of intrusion detection system based on the classifier algorithm used. These are the rule-based, misappropriation discovery and the diverse schemes. Also IDS can be classified as either real-time when they make use of persistent system tracking or sporadic when tracking rarely takes place, or at particular instant in time using data from dataset.

In recent times, new classifier techniques have been designed for Industrial Control System (ICS), having distinct features and criteria. According to the researchers in (Kabir et al 2019), they designed a classification system for IDS known as the ICS. These were divided into three types namely protocol review, traffic processing and the control process modelling. Based on the information gained from the attacks detected, countermeasures were performed by the IDS. If an attack is classified and predicted accurately, then the countermeasures designed for it will be appropriate and there will not be false alarm. This will lead to an efficient IDS system where the overhead of false alarm is eradicated. However if an attack is predicted incorrectly, then the overhead will make the IDS inefficient thereby making such model counter-productive. This makes the design of an accurate threat detecting IDS system that is capable of detecting both known and yet to be identified attack types imperative for efficient intrusion detection in IIoT system. The detection capability of an IDS system should be real time, this is because of their application in industrial operations where critical infrastructures are used and wrong outputs from such infrastructure would be detrimental to the company (Hu et al 2021).

An effective technique for the design and implementation of intrusion security system is the feature extraction. It is also important for improved performance of IDS (Cruz T et al 2019). The need for accurate threat detection where the probability of false alarm is close to 0% brought about the data pre-processing and identification known as the two mutual steps required for IDS prototype (Maglaras et al 2019). The pre-processing stage is responsible for separating the identification process from the data processing stage, here the data from the IIoT sensors are processed. The identification process reduces the attributes vector after duplicate features from the datasets are removed. The reduced dataset feature is then used to produce a more efficient and quicker attack classes.

The contributions of this paper are as follows: (i) design of an intrusion prevention in IIoT network, (ii) a technique based on the combination of deep learning and recursive based feature elimination was used for the analysis of the intrusion detection (iii) comparison of the model used in this paper with previous models in IIoT network was demonstrated. It was observed that the model presented in this paper was stable, has improved efficiency and uses lesser resource based on the results obtained from the experiments.

2. Related works

According to the work of the researchers in (Ayo et al 2021), their paper combined the current infrastructures in industrial setting for the design of an IIoT network. In their paper, the design was split into a three-step design, where each subdivision was tested and compared using the NSL-KDD and UNSW-NB15 datasets. A rule based model and generic search technique was employed for the re feature selection. The relationship between the class and each feature was calculated using the evaluation subset. The correlation between the highest value from the relationship of the class and attribute was selected. From the attributes that were selected, their advantages were evaluated and the subset of these are finally selected using function selection. The function selection makes use of genetic search method which selects attributes with the highest value. In an event where two attribute segments have equal fitness value, the recursive feature algorithm selects the attribute with lesser amount of components. At the end, the selected features are input to the artificial neural network for assault selection and template matching.

In the work of the researchers in (Soto and Nogueira 2020), the capability of the combined use of feature representation and neural network learning techniques for accurate output was described. The study was based on the assumption that combining classifier optimization with feature representation and executing with rule based algorithm will make the performance of IDS better. This paper demonstrates intrusion detection in IIoT network using deep neural network and recursive feature algorithm.

According to researchers in (Cecchinel et al 2017), an Anomaly Detection System (ADS) is a very important ingredient in security management system, its function is to detect and decide what constitutes a threat in IIoT network traffic. It is used to determine an unusual traffic and also defines normal and abnormal traffic in IIoT network. It is able to detect known and yet to be identified zero day threats, this it does by defining a pattern from both normal and abnormal traffic pattern. If there is any variation from the normal traffic pattern of the ADS,

it is regarded as an intrusion (Moustafa et al 2022). According to the works of the researchers in (Moustafa et al 2022) it concentrated on determining ADS with the aid of Particle swarm Optimization (PSO) techniques for the optimization of One-class support vector machine performance (OCSVM). OCSVM is a technique which includes harvesting Modbus/TCP message network systems for testing and validating the ADS system. According to the researchers in (Shang et al 2019), an ADS/IDS system was designed using offline data from SCADA for its learning.

According to the researchers in (Maglaras and Jiang 2017), an IDS system was designed based on Modbus/TCP protocol that made use of the K-NN classifier. The techniques used in the works presented earlier though had better performance in accuracy and other important metrics, however, they were designed for specific application scenario where the False Positive Rate (FPR) was high. In the work of the researchers in (Silva and Schukat 2017), an improved intrusion detection system was designed that matches the different structures of SCADA schemes with the different OCSVM frameworks. The two schemes were combined so as to get the correct schemes that will provide efficiency in detecting different types of threats. However this design consumed a lot of the computer's resources and had high computational requirements with a high false alarm rate.

The researchers in (Stewart et al 2020) used SCADA mechanisms with SVM algorithm to design an anomaly detection system to detect threats occasioned by the Modbus/TCP network protocol intrusion threats. However this method was not efficient in detecting other types of threats. The researchers in (Shang et al 2019) used a combination of both OCSVM method and the recurrent K-means clustering algorithm to design an anomaly detection system to improve on the method employing only the OCSVM method. The authors in (Maglaras and Jiang 2017) designed a very important infrastructure intrusion detection system based on the artificial neural network which trains a multi-perceptron ANN to detect anomaly in network traffic with the use of fault back-propagation and Levenberg-Marquard features. The researchers in (Linda et al 2012) employed the artificial neural network to track DoS/DDoS threats in IoTs, while the authors in (Hodo et al 2019), designed a decentralized intrusion detection system using artificial immunity in IoT devices. The researchers in (Chen et al 2015) [used the Possibility Risk Identification-cantered Intrusion Detection System (PRI-IDS) technique to detect the replay attacks by using the Modbus TCP/IP protocol traffic

pattern. The shortcomings of these methods presented earlier is that they have a high false alarm rates and were not able to detect yet to be identified threats sufficiently.

In the same vein, the authors in (Marsden et al 2020), designed a learning firewall which gets tagged samples with their sensors and configures itself in an automatic way by applying conservative preventive rules to isolate false alarms. In this work, a new classifier family focuses on correctness in anomaly detection, while the decision making criteria was used for zero false positive detection. In the first instance, the reason why the naïve modifications of recent classifiers like the SVM don't give the required results are presented and a generic iteration technique was used to arrive at this gaol. The classifier proposed in the paper is based on CART and it was used for the creation of firewall for a Power Grid Monitoring System. The model was tested using the KDD CUP'99 dataset to verify its validity. The results obtained from the experiments conducted in section 5 of this paper verified this claim

The researchers in (Haghighi et al 2023) designed an intrusion detection system to analyse subsurface networks in order to detect anomaly behaviour between the host and network based system. In the work of the researchers in (Li et al 2018), they described an anomaly detection system using artificial neural network with one or two hidden layers. The authors in (Yin et al 2020) further explained that deep learning was capable of a more complex computation, so it can be used when the datasets are very high and its operation is closer to the work of the human brain than the ANN.

A number of researchers in (Van Dijk et al 1995) asserted that the large volume of datasets generated from many IIoT sensors will be too cumbersome for the limited processing capabilities of the IDS architecture. The researchers in (Zafar et al 2020 derived a new rule based technique for the detection of DoS attacks based on expert knowledge. In order to identify the DoS threats, a rule based classifier algorithm was employed while the final classifier was obtained by using rules from the rule base which was validated by the domain expert. The researchers in (Rajendran et al 2019) designed a feature selection method which helps in converting databases from a higher dimension to a reduced vector size that is well suited for the problem space. Variables that are not correlated from the datasets can be isolated without diminishing the accuracy of the intrusion detection model. This dataset vector size reduction is the bedrock of feature selection techniques in threat detection in IDS (Leonard 2018). Although there are many classes of datasets with various types of attributes, only a few lend itself to feature selection techniques.

According to the researchers in (Snášel et al 2018), an attack taxonomy was presented which was derived from the various layers of the IoT stack some of which includes device infrastructure, device communication and other characteristics of the different layers that could be compromised by adversarial threats. Also the different types of threats, techniques of exploitation mechanisms for threat mitigation as well as the strategies of defence to the IoT are explained with the aid of nine real-life cybersecurity incidents which compromised IoT devices designed for the commercial, consumer and industrial sectors. All these in addition with other examples were used to demonstrate the security weaknesses of IoT systems, which also shows the consequences of attack on such network infrastructure. The taxonomy described above describes a system of approach for classifying attacks with respect to the various layers.

According to the authors in (Xenofontos et al 2022), a rule based data reduction technique was processed. The dataset attributes vector size reduction was aimed at decreasing the complexity of the IDS model. The researchers in (Herrera-Semenets et al 2019) proposed a fuzzy –based semi-supervised learning technique for intrusion detection system which comprises of high number of unlabelled data being processed by lower vector size labelled data. The essence of the lower vector size labelled data is to reduce the latency of unusual network traffic detection. In the paper fuzzy measure was used by the authors to generate an independently trained hidden node feedforward neural network to produce a classification of a fuzzy set vector having small, medium and large classification of a high amount of unlabelled data. After using each vector of the classified data, the training set is used again until the start training dataset is exhausted. The researchers in (Kabir et al 2019), proposed a wrapper based network intrusion data system architecture that used the Bayesian networks. In the paper, feature selection was used to extract the relevant features from the dataset in order for the Bayesian network classifier to correctly predict the threat types.

The authors in (Ashfaq et al 2018) proposed a crossbreed technique made up of Support Vector Machine and ant colony. The purpose of combining these two machine learning techniques was in order to know the reason for the unsuitability of each method used separately and to derive a more appropriate grouping of the dataset features. In the work of the authors in (Kirnapure and Patil 2018), a wrapper technique for memory constrained devices using decision tree was designed. The proposed approach comprises of four operations, this includes (i) pre-processing, which was the isolation of redundant threat types,

(ii) feature selection based on genetic algorithm, (iii) post-processing required for generating a fixed sized vector output and (iv) traffic classification approach based on the neurotone method. In the same vein, the researchers in (Sivatha et al 2014) designed a wrapper appropriate approach based on a violation word for a large volume of attributes having good precision in its classification. The proposed wrapping fitness value is appropriate for feature extraction and at the same time has a high prediction accuracy. According to the researchers in (Chakraborty and Kawamura 2020), the authors proposed a decision tree classifier based network intrusion detection system function collection using genetic algorithm. In the paper, the authors employed genetic algorithm to generate the data to be imputed to the decision trees as a classifier algorithm for the improvement in the threat identification and also for the reduction of false alarms by the cybersecurity program.

In the work of the authors in (Leonard 2018), a smart rule based identification scheme for the detection, of the Denial of Service (DoS) attack in cloud server was proposed. The proposed method employed scoring and rating algorithm to formulate attack on the cloud and then identify and select the best functionality. In order to detect the threats, a rule based grouping procedure based on expertise in quality was employed for the selection of the features. The main advantage of the proposal is a reduction in the false alarm rate and an improved network security. However as a result of the complexity in nature of threats, the danger of threat confusion was not put into consideration. In the work of the researchers in (Stein et al 2008), a novel technique of feature selection with a better and efficient K-NN classifier was suggested for IDS. The proposed feature set reduced very greatly the presence of irrelevant features, thereby making the K-NN classifier to improve in the detection of the different types of intrusion.

From the different works highlighted in this review, it was observed that deep learning algorithms is very appropriate for an efficient intrusion detection system for IIoT networks. This is because, it is capable of a very high degree of threat prediction accuracy, resulting in low false alarm rate. This served as the basis for the design in this paper where a deep learning model was combined with recursive feature elimination technique for the extraction of important features in a dataset for an optimum intrusion detection system. The model has the capability of detecting anomaly in IIoT network traffic thereby isolating cyber threats. The deep feedback neural network (DFBNN) proposed in this paper with recursive feature elimination model has a rule based model which uses genetic search engine to extract the

important features from the dataset. The DAE-DFBNN algorithm was used to group the IIoT network by grouping the constraint data in the DAE. The model to a very large extent is able to convert a high vector dataset with redundancies to a lower vector dataset for optimum threat detection in IIoT.

3. Overview of Industrial Internet of Things

There are many types of industries, this includes: manufacturing, healthcare, transportation, electricity, education, just to mention but a few. All these industries are affected by the Industry 4.0 revolution, which requires that production and optimization in industries depend on operational technology. In the past, fog and edge computing technologies were required by IIoT for efficient production. Deep Learning (DL) algorithms enhance the capabilities of big data analytics, on the other hand, IIoT technologies, improve the efficiency of DL algorithms. DL algorithms help in identifying, categorizing, and making decision in each of these data types. The combination of DL algorithm and big data technologies produces important data for making policies. DL is very important in IIoT and data analytics for efficient attribute selection in datasets especially in data streaming real time processing in edge computing systems.

IIoT applications are used in many businesses including, grocery, education, healthcare, transportation and automobiles. IIoT can also improve performance and service delivery based on optimization of operational technologies (OT). It is important to first design the various stages of production with the aim of developing an optimized model for industrial operations. It should be realized that the combination of IIoT, machine learning and deep learning will make for enhanced production and satisfaction of customers by combining different machines, procedures and applications. IIoT technology requires many technologies which would be integrated properly.

Improvement in technology enables intelligent machines, engines and control systems to perform tasks that are repetitive in nature in order to accomplish certain goals without human interference. Also advancements in smart workplace, cognitive automation, intelligent data discovery are also affected by improvements in machine learning techniques. An Industrial Internet of things is made up of physical equipment, control systems and other integrated components. The advantage of IIoT is that it can receive multiple types of data with the help of its embedded sensors and process them with the help of its microcontrollers. IIoT networks are also scalable, as a result of this, it can perform many intelligent functions. Being

intelligent, the IIoT control software are usually updated regularly and it is able to learn using neural network learning paradigm in real time. This helps it to make informed decision on the go.

3.1 Components of Industrial Internet of Things

The IIoT architecture is made up of four main components, these are:

(i) Architecture Intelligent Edge Gateway (AIEG): This is a software embedded in the IIoT in addition to the sensor nodes that receive, aggregate and send aggregated data to the IIoT gateway. The IIoT gateway processes the aggregated data and transmit the filtered data to the cloud IIoT network. It should be realized here that the IIoT framework utilizes data processing, machine learning and AI techniques to process very large size of data. The various processing applied to the sensed data includes device control, stream analytics, management of event, rule engine, updates and alerts. Operations performed by the IIoT network includes analysis of big data, data authorization, virtualization, end-to-end encryption, APIs application and SDKs

(ii) Business Incorporation and Platform: This connects the various IIoT schemes with the backend framework in order to ascertain that the appropriate data are captured by the sensors and processed in accordance to the control software. Examples of business incorporation and platform are ERP, QMS, scheduling and planning. Data analysis from IIoT systems is of three types, they are descriptive, predictive and prescriptive. The IIoT architecture is shown in figure 1. The figure contains three layers namely intelligent gateway, IoT cloud and business application and integration.

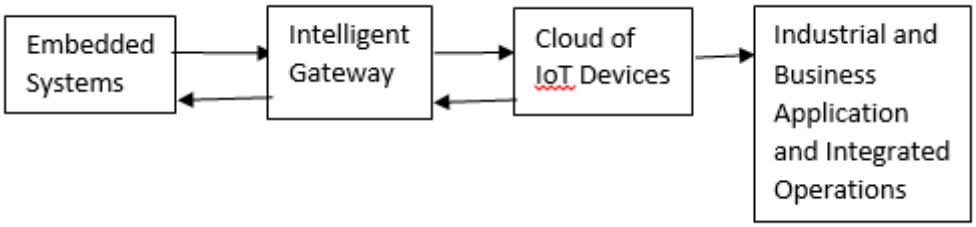

Figure 1 Architecture of Industrial Internet of Things

4.1 Materials and Methods

Intrusion Detection System for Industrial Internet of Things

In this section, the deep feedback neural network (DFBNN) was used to obtain an efficient anomaly detection system for IIoT positions. During the testing step, a dual feature extraction

uses genetic search system together with rule based algorithm. The testing stage computes the relationship between individual features together with the category it belongs. A class and attribute interaction with the most similar features was selected for filtering. This process is referred to as function assessment. The aim of the genetic search is to assess the importance of each feature using the function assessment to produce the attributes having the highest value of suitability. In the event where two attribute subsections obtains equal performance score, the rule based algorithm will be used to derive the feature vectors with the lowest amount of subsection attributes. In the end, the selected attributes are input to the artificial neural network, which creates the model and groups the various types of attacks. These are the components from which the DFBNN was created in order to detect current and yet to be identified threats. The use of the DFBNN was to detect anomalous dataset vectors when the data traffic was tested. When the reduced vector set is fed into the DFBNN model, different types of feature vectors will be developed from which the normal and abnormal vector sets will be grouped.

4.2 The Deep Feedback Neural Network (DFBNN)

The deep learning models used in this paper was the deep feedback networks also known as the recurrent network. It is different from the feedforward networks in that it has a feedback loop in other words, data can be transmitted in two directions both forward and backward. The aim of a feedback neural network is find an approximation to a function f(x). For instance, if b = f'(a) forwards an input a to get the output c. In general, the neural network will have many inputs which will be summed with a weighted inputs and the bias to give the sum of the input m. i.e. $m = w_1a_1 + w_2a_2 + w_3a_3 + w_4a_4 + -\ldots\ldots + w_ka_k + b$. Here the number of the inputs are k. The summation input m is then fed to the activation function to get c. The summation inputs can also be in various layers known as the hidden layers before the final output is fed to the activation function. The advantage of the feedback neural network is that the data is transmitted both forward and backward until the appropriate output c is obtained. Feedback is often necessary due to the fact that a pattern of previous inputs may be needed to obtain a given output i.e. the neural network is said to learn from its previous inputs. The neural network was trained by a stochastic slope descent back-propagation technique. A Feedback neural network can be made to model a more complex activity by adding hidden layers, however the number of hidden layers used in this paper was two.

The input data is connected to the input nodes after which it is being forwarded to the hidden nodes, there may be need for feedback to previous layers with a bias to get the appropriate summation value before this will be fed to a hyperbolic tangent transfer function in order to generate the output. In order to compute the output of the DFBNN mode, a backpropagation fault is computed which serves as the difference between the predicted and actual data, this causes the neural network to transmit backwords to previous hidden layers to manipulate the input weights and possibly the bias. The bias is computed by comparing the given reduced vector size datasets to the expected output. In order to better the convergence rate of the model, the neural model is trained with different weight and biases, which are learned for a quicker match of predicted and expected output in the future.

4.3 Variational Autoencoder (VAE)

An encoder is a neural network which maps input data into a latent space. In traditional auto-encoder e.g. deep auto-encoder, the output generated usually consists of fixed point in latent space, however in variational auto-encoder, the output generated is usually a probabilistic distribution of the input data. This enables variational auto-encoder (VAE) to model uncertainties in data as well as variation in datasets.

A decoder is used together with the encoder to get the original data from the latent space representation. When a sample is input as the representation of the latent space distribution, the decoder attempts to reconstruct a close representation of the original input from the latent space distribution. This enables VAE to derive new instances of data from a given distribution. This is the strength of the variational auto-encoder in being able to detect yet to be identified threats and anomaly in traffic pattern of the IDS.

The latent space is a reduced representation of the probability distribution of the input data. Therefore VAE produces a reduced and a good approximate of the complex probability distribution of the original dataset. VAE was used in this paper due to its ability to produce a good approximate to a complex probability distribution in which the intrusion detection datasets of both NSL-KDD and UNSW-NB15 represents. Its ability to derive new instances of data from a given distribution also makes it ideal for the problem space proposed in this paper. .The variational auto-encoder approximates the actual posterior distribution of the latent variables from the original large input dataset. The Bayesian inference is used for the estimation of the latent variable distribution. VAE is composed of the following steps:

a) Feeding the input data x to the encoder, which then produces the latent space parameter distribution b(y/x) with mean ψ and variance σ^2
b) Sampling of the latent variables y from the b(y/x) distribution using reparameterization techniques.
c) Passing the sampled y into the decoder to get the reconstruction of the original input x/ The reconstructed data will be similar to the original input data

A variational auto-encoder makes an assumption about the probability distribution of variables in a given dataset. A variational approach is used for learning the distribution of data in the dataset. This results in reduction of the size of the dataset using an estimate function for algorithm training, known as the stochastic Gradient Variational Bayes estimator. The data generated from the Bayesian estimator is modelled as a directed graph, while the output is the approximated distribution of data in the dataset and where ψ and φ define the encoder parameter

4.4 The Variational Autoencoder Model (VAE)

This represents a feedback neural network technique for fast unsupervised neural network training. It studies the approximation of a given task in which the output (y) is the sum of the reduced dataset of inputs (v_1……..v_n), in order to find a group of data such that (y v_1….v_n). This can be represented by the vectors ($x^{(i)}$), as the nodes in the input as well many hidden units with initialization attributes that are non-linear

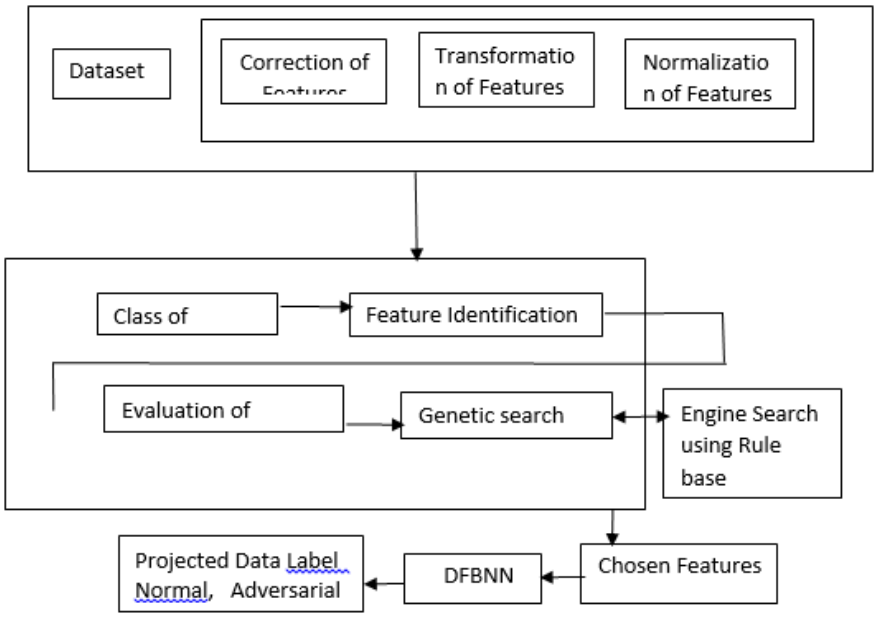

Fig 2 Projection Intrusion Detection System in Industrial Internet of Things.

In order for the model to learn about the possibility of reducing input data features, the features that were extracted use lower number of neurons than that of the nodes at the input using an estimate function for algorithm training, known as the stochastic Gradient Variational Bayes estimator. Due to this fact, the model uses only the important features in the dataset vector for its intrusion detection, this invariably reduces the vector size of the output dataset. At last the output layer given by ($x^{(-i)}$) is a near description of the input layer.

The simplest framework of a variational auto-encoder consist of three layers, these include feeding the input data to the encoder, sampling the outputs (reduced set) from the inputs and passing the sampled output into the decoder to get a reconstruction (reduced dataset) of the inputs . When the dataset used for training ($x^{(i)}$) contains n samples, with each ($x^{(-i)}$) with n samples having many dimensions together with a vector with spatial function of (f0) itsTanh starting function can be computed as shown in equation 1

$$X(x) = \frac{1 - e^{-2x}}{1 - e^{-2x}} \qquad (1)$$

The two parts of the varaitional auto-encoder algorithm are the encoder and decoder [85, 86]. The encoder technique mapping technique (fφ) is used for the transformation of the input vector ($x^{(i)}$) unto the representation of the hidden layer ($y^{(i)}$), while the dimension ($x^{(i)}$) will be decreased to generate the correct number of datasets as shown in equation 2.

$$f\varphi\ (x^{(i)}) = U(R_{x(i)} + b) \qquad (2)$$

here R is a weight matrix of size s^0 by s^n , s^n denotes the number of neurons in the hidden layer $s^0 < s^{n,}$, while b denotes the bias vector, u denotes the tanh activation function, while φ, r and b represents the parameters for mapping.

The graph of the hidden layer's product representation is plotted, while the translator method is computed using the Bayesian inference h(φ) to be an estimate for ($x^{(-i)}$) the function used to represent the input ($x^{(-i)}$) as an approximation for h(φ) is as shown in equation 3

$$h\varphi\ ((x^{(i)}) = U(R_{z(i)}' + b') \qquad (3)$$

here R' denotes an s^0 by s^h weight matrix, while b' represents the bias vector, φ' denotes the parameters for mapping (R', b').

The information contained in the reduced vector representation was used as the input for reconstructing the real information after its transformation to represent the hidden surface. The difference between the original dataset and its reduced vector representation that was used for training the dataset can be computed as shown in equation 4

$E(x,x') = \frac{1}{2}\sum_i^r \quad x^i - x^{-1} \quad ^2$, $\varphi = [R, b] =$ argmin E (x, x') (4)

The selection of the feature phase

Assumption 1: In order for a feature W to be considered relevant, there should be w_i and d where $p(W_i = w_i) \gtrdot 0$ is as shown in equation 5

$$p(D = d, W_i = w_i) \neq p(D - d) \quad (5)$$

Assumption 2: When the relation between the association that exists between a component and an outside parameter function is known, together with the inter-correlation over every set of parameters, the relation between a standard test which comprise of all the modules in the outer parameters can be computed as shown in equation 6

$$S_{yd} = \frac{l_{s_{yi}}}{\sqrt{ll+1(l-1)\ s_{jj}s}} \quad (6)$$

In equation 6, S_{yd} is the relation between the addition of the modules and the outer parameter, while l refers the element size, s_{yjj} denotes the mean of the outer variable correlations components with the outer component, while s_{ii} is the mean of the component to the outer variable inter correlation.

Assumption 3: (Genetic search) Genetic search is a method of exploring that is based on normal progression. A suitability task in which both the accuracy of detection and latency of detection are both considered crucial to the efficiency of the task at hand. This is defined by the fitness function shown in equation 7

$$\text{Fitness } (Y) = \frac{3}{5}B + \frac{2}{5} = \left(3 - \frac{U+G}{4}\right) \quad (7)$$

Here Y denotes a subset of a function, B denotes the mean of DFBNN's cross validation precision, U denotes the number of times the input dataset samples were trained while G denotes size of the subset features.

Assumption 4: When there exists many feature subsets (G) with similar values of fitness, the recursive feature elimination can reduce a vector to (W_i) which contains lesser features (fY_f), otherwise it will produce a feature subgroup having the highest appropriate value (G) with the first classifier as shown in equation

$$S = \begin{bmatrix} W_i \text{ i} & \text{if } W_i \in G > \cap Y_f \\ w_i & if\ G_{hi\cap\Phi} \end{bmatrix} \quad (8]$$

This paper provides an efficient model for intrusion detection. This is necessary to protect the IIoT system from abnormal situation. Figure 2 shows the model of the proposed architecture including the testing and training sessions. It also shows the intrusion detection model projected for the IIoT network. From the IIoT network, the model finds out important information in a large scale dataset. The first step in this model is the data pre-processing, this are function regularization and translation.

4.5.1 Feature Transformation

As the proposed framework has mathematical properties only, a mathematical formula is derived for the values of the attributes in the dataset, for example the NSL-KDD dataset has many expressive attributes, similar nature of procedures having metaphoric values such as ICMP, UDP and TCP, these are shown in figures 3, 4 and 5 respectively.

4.5.2 Normalization of Feature

Deep learning is based on different features which relies on masses. It is possible that data is skewed into different dataset attribute, this usually results in some data being updated faster than the other. [8, 11]. Due to this, it is crucial to solve the problem with statistical normalization, where the Z-score function of every feature value ($U^{(\varphi)}$) is computed as in equation 9

$$Y^{(i)} = \frac{u^{(i)} - \lambda}{\psi} \qquad (9)$$

Here ψ denotes the standard deviation and it is the average of the λ values in a function defined $u^{(i)}$ where I = 1,2,3…..m. In this work, the network is assumed to have high dimension, as a result of this it is necessary to reduce the size so as to enhance the network resources and design a model that is compatible to the resource constrained IIoT devices. Due to this, the VAE-DFBNN technique is designed to lower high feature attributes to fewer feature attributes.

In a more detailed way, the model proposed in this paper has a non-linear mechanism which perform encryption on many feature set to produce fewer feature set. The hidden stages of the artificial neural network was designed to reduce the feature set dimension. The aim of the recursive feature section is to reduce the attributes of the input dataset. The use of the VAE-DFBNN is to determine the crucial properties required for the intrusion detection from the full dataset attribute. The reduction of the dataset attribute size will result in an enhanced learning and reduced time lag

4.7 Dataset Details

In order to test, assess and examine the characteristics of the intrusion detection scheme, its results will not only be linked to the type of dataset used but also on future threats that are not yet identified. The dataset selected will also be crucial in order to obtain improved results. Selection of the appropriate dataset attributes will not only produce excellent results in offline scenario, but also in a real life setting, the dataset being used by many authors in the design of IDS is the NSL-KDD dataset, which represents an upgrade to the KDD CUP'99 dataset. The NSL-KDD dataset overcomes the shortcoming in the KDD CUP'99' by the deletion of redundant information, while selecting attributes based on their frequency in the dataset. The NSL-KDD dataset contains 148,517 attributes comprising of 77,054 standard and 71,460 anomalies. At the end of the pre-processing stage, 41 attributes in addition to a class are identified in the dataset. Iit should be noticed here that there are five types of classes in NSL-KDD dataset, these include Probing, DOS, remote to local (R2L), user to root (U2R) and the normal. However, the NSL-KDD dataset is now obsolete, as a result of this, the UNSW-NB15 dataset is being used .in this paper. The UNSW-NB15 dataset comprises of artificial attack attributes which is used together with current real life normal network traffic. It contains 257,673 records which consists of 61,000 normal and 164,673 threats, with each having 41 features and a classification indicator. There are ten different class indicators in the UNSW-NB15 dataset, these include Fussers, examination, DoS, backdoors, standard, vulnerabilities, reconnaissance, worm and shellcode making up I normal and nine threat classes.

4.6 Analysis of Model Performance

In order to compare the performance of the Deep learning model with the recursive feature elimination proposed in this paper alongside existing models, some performance metrics were used. The number of correct and wrong output in a classification problem was added together and the result compared with the expected output, which includes precision of accuracy, recall, F1 –score and specificity. The confusion matrix was computed using the following statistical indexes, correct positive (CP), correct negative (TN), incorrect positive (IP), and incorrect negative (IN). Equations 10 – 16 shows these.

$$\text{Accuracy} = \frac{CP + CN}{CP + IP + IN + CN} \quad (10)$$

$$\text{Precision} = \frac{CP}{CP + IP} \quad (11)$$

$$\text{Sensitivity} = \frac{CP}{CP + IN} \quad (12)$$

$$\text{Specificity} = \frac{CN}{CN + IP} \quad (13)$$

$$F1 - \text{score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} * \text{Recall}} \quad (14)$$

$$\text{CPR} = \frac{CP}{CP + IN} \quad (15)$$

$$\text{IPR} = \frac{IP}{IP + CN} \quad (16)$$

In equations 10 accuracy refers to the rate at which the prediction is right, on the other hand, precision in equation 11 refers to the rate at which the class is predicted right. Recall refers to the number of the positive class that was predicted right, while in equation 13, specificity describes how the negatives were correctly detected. The F1-score in equation 14 refers to the ratio of precision and recall. Specificity in equation 14 is defined as the ratio of the approximation of the number of true negative to the total number of incorrect predictions. In equation 15, the correct positive rate refers to the ratio of the correctly identified threats to the total number database classes. The correct positive rate can also be referred to as the rate of discovery. The incorrect alarm rate (IAR) in equation 16, is computed as the ratio of the incorrectly denied records and the total number of actual records. The evaluation metric of the incorrect positive rate is as defined in equation 16. Due to this fact, the bottom line in intrusion detection system is the attainment of accuracy rate close to 100% with almost no false alarm

Table 1 Projected Evaluation Method.

Dataset	CP Rate	IP Rate	Precision Value	Recall Value	F-measure	ROC	Class
UNSW-NB15	0.9985	0.09	0.972	0.9985	0.992	0.992	Threat
	0.9995	0.0009	0.9985	0.9965	0.971	0.999	Normal
NSL-KDD	0.9965	0.09	0.989	0.9995	0.998	0.999	Threat
	0.9995	0.0009	0.9985	0.995	0.973	0.998	Normal

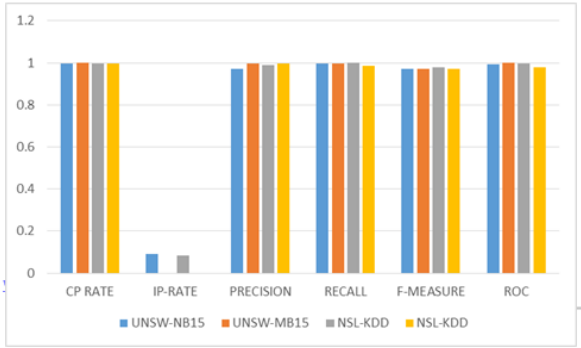

Figure 3: Evaluation of the performance of VAE-DFBNN on the two datasets.

5.1 RESULTS AND ANALYSIS

The model proposed in this paper was implemented with R programming, while the analysis was performed using the performance metrics stated in section 4. Datasets with the VAE-DFBNN and recursive feature elimination was incorporated into the design. The NSL-KDD database has 77,054 normal attributes and 71,460 threat attributes, together with some samples in the UNSW-NB15 datasets that has 93,000 normal attributes and 92,000 threat attributes.

The parameters and network topology used in the experiments produced the highest data rate with lowest incorrect positive rate. The VAE-DFBNN model projected in this paper produced the highest threat detection rate alongside with the least incorrect.

Table 2 Performance Evaluation of the selected datasets.

Dataset	Accuracy	Rate of detection	IPR
NSL-KDD	99.3%	99.2%	0.9%
UNSW-NB15	99.1%	99.95%	1.0%

Positive rate using the network topology and parameters in the experiments conducted. In performing the experiments, the topology comprises of 40 nodes in the input layer with 3, 4 and 3 nodes in the three hidden layers and 40 nodes in the output layer. The DFBNN model was incorporated with the recursive feature elimination together with the variational auto-encoder feature. The deep feedback neural network comprises of 40 nodes in the input layer with 3, 4 and 3 nodes as the three hidden layers, with 40 nodes in the output layer. The learning rate for the NSL-KDD dataset was .0025 with 0.15 momenta start. In the case of the UNSW-NB15 dataset, the momentum start was 0.15, the momentum stable was 0.35, while the L1 and L2 regularization were 15 and 10^{-7} respectively, the ramp momentum was 10^{6}, the rate of annealing was 2.5 X $10^{-6.}$ The simulations involving both UNSW-NB15 and NSL-KDD dataset was conducted using 100 cycle of simulations with 9.9925 learning rate, the tanh activation function was used for the deep neural network model.

The data obtained from the experiments conducted for both NSL-KDD and UNSW-NB15 datasets are presented in table 1 while the graph is shown in figure 3. From the results, it can be seen that the model proposed in this paper outperforms other competing models in many of the performance metrics used.

The accuracy, incorrect positive rate and the detection rate of the VAE-DFBNN model proposed in this paper on the two dataset used is shown in table 2. From the table, it can be seen that the VAE-DFBNN model performs better with the NSL-KDD dataset by 1.5%, on the other hand the UNSW-NB15 dataset performs better in the rate of threat detection and incorrect positive rate by 2% and 12 % respectively The rate of threat discovery using the classes in both NSL-KDD and UNSW-Nb15 datasets for the VAE-DFBNN model is shown in table 3. From the table it can be seen that the UNSW-NB15 datasets was able to detect six (6) more types of threats than the NSL-KDD.

The cumulative performance measures for the VAE-DFBNN is shown in figure 3. It uses the reduced UNSW-NB15 dataset derived from recursive feature elimination. From the figure, it can be seen that the incorrect positive rate (IPR) and precision of the VAE-DFBNN model provides improvement to the compared models. The VAE-DFBNN intrusion detection model achieves an Intrusion detection accuracy of 99.25%, which means that it outperforms the closest model DFFNN by 0.15%. In addition to this, in the VAE-DFBNN comparison with other classification techniques, it has the least Incorrect positive rate (IPR) of 1.0%. From the experiments conducted in this paper, the VAE-DFBNN model outperformed all the compared techniques using the various performance metrics used in this paper, when the required subset of the UNSW-NB15 dataset was used.

Table 3 Rate of detection using the NSL-KDD and UNSW-NB15 dataset classes.

Dataset	**Normal**	**Backdoor**	**DoS**	**Probe**	**Shellcode**	**U2R**	**Exploits**
NSL-KDD	99.8%	-	99.5	-	-	76.1%	-
UNSW-NB15	99.7%	95.7%	97.5%	-	91.1%	-	98.4%
Dataset	**Analysis**	**Salicode**	**Generic**	**R^2**	**Reconnaissance**	**Fuzzer**	
NSL-KDD	-	-	-	95.2%	-	-	
UNSW-NB15	92.8%	91.1%	99.9%	-	92.6%	67.4%	

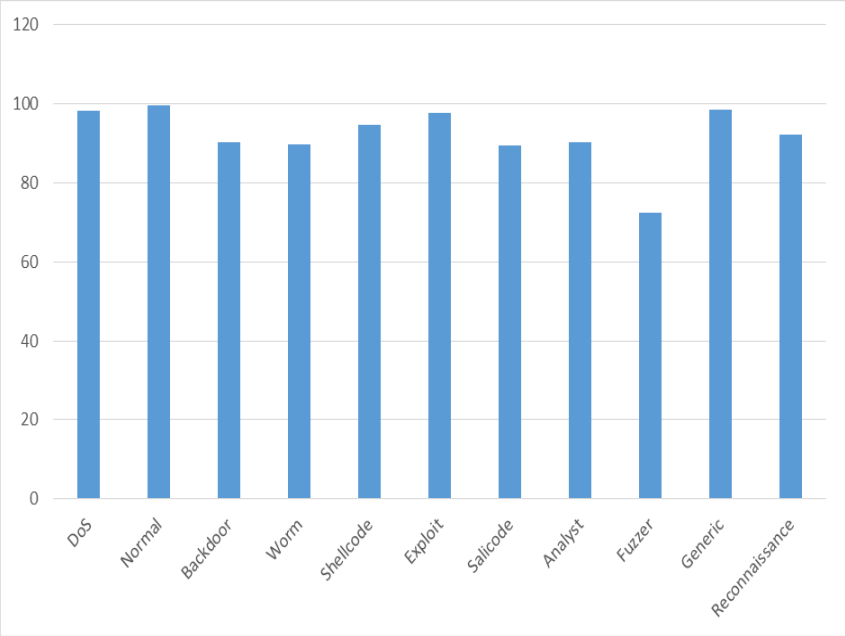

Figure 4: Rates of detection for UNSW-NB15 dataset classes.

The improvement witnessed in this model is due to the recursive feature elimination which used only a sunset of the UNSW-NB15 dataset thereby enabling quicker threat detection. Also the recursive feature based fitness helped to select the most appropriate dataset attributes for enhanced performance In terms of accuracy of threat prediction UNSW-NB15 dataset outperforms NSL-KDD by 8%. In terms of recall performance metric, both datasets perform very well with NSL-KDD edging UNSW-NB15 by 3%. In terms of F-measure performance metric, UNSW-NB15 dataset outperforms NSL-KDD dataset by 9%. With the receiver operating characteristics (ROC) the NSL-KDD dataset outperforms UNSW-NB15dataset by 8%.

Figure 4 shows the detection rates of the UNSW-NB15 datasets on the standard threat types, from the figure it can be seen that the dataset has a very high detection rate for the normal, exploits and genetic types. Among these, the normal represents a good traffic pattern devoid of attack while the other two are threat types. The UNSW-NB15 dataset however has a relatively low detection rate for worm and fuzzer threats 77% and 62 % respectively. Since the NSL-KDD dataset can detect only five (5) types of threats, a bar chart showing its performance on threat detection rate is shown in figure 5. From figure 5 it can be seen that the NSL-KDD dataset has a high detection rate on four of the threat types, namely DoS, probe, normal and R^2, with detection rates of 100%, 100%, 100% and 94% respectively while it has a relatively low detection rate on U2R threat i.e. 76%.

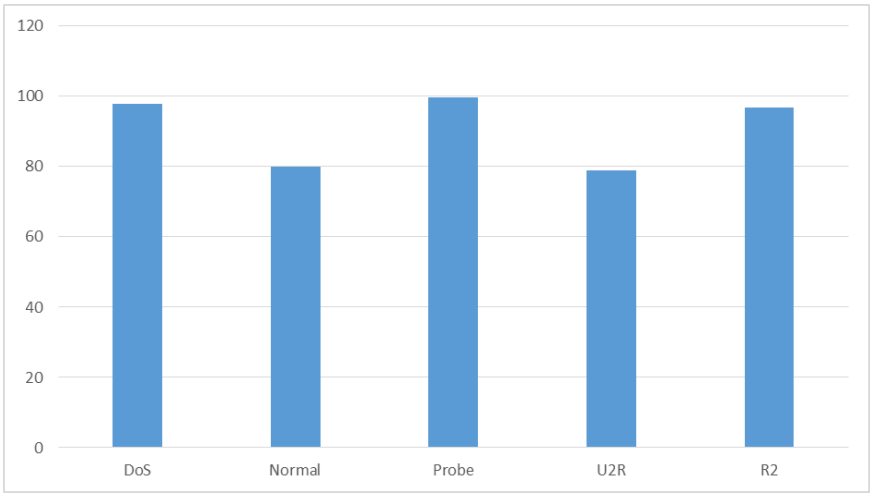

Figure 5: Rates of detection for NSL-KDD dataset classes.

5.2 The Performance of VAE-DFBNN with Current Methods

Table 4 shows the effect of the recursive feature elimination method on the detection efficiency of the classification algorithm. The table gives comparison between the recursive feature selection and other existing selection methods making use of the reduced UNSW-NB15 dataset. The incorrect positive rate of the recursive feature elimination technique is better than other compared methods by between 2% - 12%. In terms of precision the recursive feature elimination outperforms the other compared selection methods by between 7% - 20%.

On the F-score performance metric the recursive feature elimination outperforms others by between 12% - 25%. On the recall performance metric, the recursive feature elimination outperforms the compared selection method by between12% - 28%. On the precision performance metric, the recursive feature elimination method outperforms the other selection methods by between 5% - 21%. On the receiver operating characteristics (ROC) performance metric, the recursive feature elimination outperforms the other compared selection method s by between 4% - 15%. The reason for the improved performance of the recursive feature elimination technique is due to its combined use of machine learning algorithm at the core of its design which is used to select the appropriate features from the dataset. The machine learning has embedded in it a filter based feature selection which selects features based on its score on an importance ranking. This therefore gives it an edge over the selection methods used in comparison which uses only filter based selection.

Table 4 Performance analysis evaluation of the feature selection techniques on UNSW-NB15 dataset.

Metrics used for Performance Evaluation						
Model	**Accuracy (%)**	**IPR (%)**	**F-score (%)**	**Recall (%)**	**Precision (%)**	**ROC curve (%)**
Wrapper + neurotree	98.45	1.57	0.989	0.983	0.991	0.998
SVM+EML+K-means	95.81	1.83	0.949	0.998	0.899	0.989
Genetic Algorithm +SVM	97.6	0.015	0.969	0.998	0.942	0.985
CNN+LSTM	94.24	-	0.961	0.991	0.931	0.988
Modified KNN	98.9	1.2	0.994	0.997	0.991	0.998
DAE-DFFNN	99.1	1.1	0.991	0.998	0.972	0.991
Recursive Feature Elimination	99.463	0.9	0.994	0.999	0.986	0.998

In table 5, the VAE-DFBNN's performance was compared with other ten intrusion detection techniques, these are DL based ADS system, support vector machine using filter (F-SVM) .[58], Computer Vision Method (CVT) [64], the triangular area nearest neighbours, (TANN) [65], Dirichlet Mixture Method (DMM) [61], DBN [62], RNN [64], DNN [63], Ensemble-DNN `[66] and DFFNN [60]. The performance of all these intrusion detection techniques in terms of correct positive rate and incorrect positive rate when the NSL-KDD DATASET was used is shown in the table.

Table 5: Performance comparison of the NSL-KDD dataset, with other ten classifiers.

Technique	**Rate of Detection (%)**	**IPR (%)**
F-SVM	92.3	8.6
CVT	95.4	5.5
DMM	97.3	2.3
TANN	91.3	9.3
DBN	95.2	4.4
RNN	75.V	3.5
RNN	75.V	3.5
DNN	76.4	14.1
Ensemble-DNN	98.4	14.3
ADS-DL	99.1	1.7
DAE-DFFNN	99	1.1
VAE-DFBNN	99.2	1.0

From Table 5, it can be seen that the VAE-DFBNN has outstanding performance culminating in 99.2% detection rate and 1.0% incorrect positive rate (IPR). From the table, four models had the ability to detect threat events after the Recursive feature elimination was used. The F-SVM employed information sharing to compute for both linearity and non-linearity of data,

the solution was then combined with the SVM to detect threat activities .It should be noted that the use of the recursive feature elimination is important for the selection of appropriate subset of the used dataset due to its combined use of machine learning algorithm at the core of its design which is used to select the appropriate features from the dataset for enhanced and fast threat detection.

The detection rate (DR) for F-SVM is 93.1% while the Incorrect Positive Rate (IPR) is 6.4%. The detection rate of CVT is 96.2%, while its incorrect positive rate is 5.0%. The detection rate (DR) for DMM is 97.8% while its incorrect positive rate is 2.2%. The detection rate of TANN is 91.7%, while its incorrect positive rate is 8.2%. The detection rate of DBN is 95.7%, while its incorrect positive rate is 4.2%. The detection rate of RNN is 75.7%, while its incorrect positive rate is 3.2%. The detection rate of DNN is 76.7%, while its incorrect positive rate is 14.2%. The detection rate of ensemble DNN is 98.7%, while its incorrect positive rate is 14.2%. The detection rate of ADS is 98.7%, while its incorrect positive rate is 1.2%. The detection rate of VAE-DFBNN is 99.7%, while its incorrect positive rate is 1.0%. The performance of the VAE-DFBNN model proposed in this paper outweighs the other techniques because it uses the variational auto-encoder on the feedback neural network, together with the recursive feature elimination which uses only a trained subset of the used datasets. The dataset used was computed using the binary sigmodal activation function on layers of inputs to the neural network. The inputs are the datasets of the corresponding NSL-KDD and UNSW-NB15 while the output is obtained by applying the tanh activation function on the subset of the input neurons from the datasets. The output provides the condition of the network n being either normal or adversarial. This results in fast and accurate threat detection suitable for industrial setup having

5.3 Real World Deployment of IoT System

In order to further validate the model proposed in this paper, a real world deployment of the network was designed. The ioT system proposed in shown in Figure 6. It comprises of a Bouder Router (BR) which acts as the DODAG root for the 6LoWPAN network. The Bouder Router connects the IoT devices to the Internet using the SLIP interface. The Bouder Router acts as the edge router as it has higher memory and computation power than the 6LoWPAN network In the hardware network testbed, the IDS nodes/routers acts as the first tier of nodes and their task are to forward data packets towards the edge router and back. The IDS nodes are equipped with an intrusion detection protocol so as to track any malicious traffic on the

network which ae sent to the edge router which is responsible to sending periodic alerts of anomaly in traffic patterns.

The IDS nodes are at least a single hop from the edge router so as to be able to detect adversarial packet transmitted to the edge router. In the network topology, malicious nodes are assumed to be in the second tier of the network, this is necessary so that they can only be part of the network through any of the IDS nodes from where the data packets will be forwarded to the edge router. With this arrangement, it will be possible to simulate various patterns of malicious traffic. In Figure 2, a network of five IDS nodes with three adversarial nodes is shown. In the testbed, the experimental setup consists of Cooja which uses Tmote Sky motes, while the Tmote Sky uses a CC2420 IEEE802.15.4 transceiver having 250kbs and with 48kb of flash and 10kb of RAM.

In the testbed, the topology consists of one Ede router, five IDS nodes and three malicious nodes. The positions of the adversarial nodes are made random to the consistent with real life scenario. This scenario is shown in Figure 6.

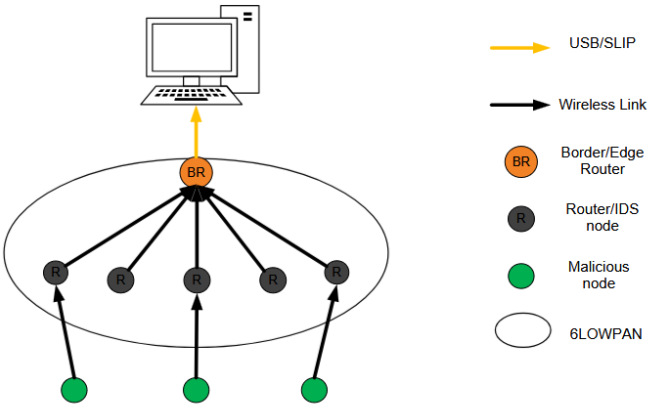

Figure 6 The Experimental Setup.

For the experimental setup, the network environment was formed without any adversarial nodes while the IDS nodes served as the RPL routers which transmitted only control messages with themselves with the edge router.

One functionality of the IoT is their Radio Duty cycle (RDC), As IoTs are usually embedded devices, they have low data transmission rate compared to normal network nodes As a result of this, it is illogical to leave the IoT radio on at all times especially when daa transmission does not occur often. In view of this, it becomes impertinent to devise ways save the power

consumption of IoT devices during data transmission, these resulted in the design of various RDC protocols which monitors the rate at which nodes are being turned on and off during data transmission. For example in the ContikiMAC RDC protocol, the sleep patterns of various network nodes was used for switching the IoT radio on or off for either transmitting or listening to the network. There is another version of the ContikiMAC RDC protocol known as the NullRDC where the IoT radio is switched on permanently. The shortcoming of the NullRDC is that the IoTs radio power dissipation is higher than all other network functionality including the computation overhead. The ContikiMAC RDC protocol was used in this paper.

In the result analysis, the power dissipated by the intrusion detection system was evaluated as well as the memory overhead resulting from the additional RDS features to both the IoT nodes and the Edge router. These two performance measure are discussed more elaborately in the next section.

5.4 Analysis of Power Consumption

It is well known that the IoT nodes are limited in both computation and memory capabilities, therefore the addition of any feature or protocol must take cognisance of these resource constraints. In order to measure the power consumption of the IDS protocol, the power trace tool embedded in the Contiki OS was used. The tool measures the time spent by each mote in any of the four states which are transmit (Ts), reception (Rx) low power mode (LPM) and CPU processing.

In order to compute the energy consumption, the formula shown in equation 17 is used.

$$E\ (mWs) = V * (Tx * 19.5 + Rx * 21.8 + LPM * 0.0545 + CPU * 1.8) \quad (17)$$

The current consumed is computed by multiplying time used up in each state and adding for the four different states. The energy consumption is computed by the product of the total consumed current and the nominal voltage.

Table 6 shows the consumed current in each transmission state, these values are obtained from the Tmote Sky data sheet shown in Table 6

Table 6: Tmote-Sky nodes base measurement.

Operating Condition	Minimum	Nominal	Maximum
Voltage in the Supply	2.0V		3.5V
Voltage in the supply when flash memory	2.6V		3.5V

is programmed			
Consumed current of receiving radio on microcontroller		21/9mA	31.1mA
Consumed current of transmitting radio on microcontroller		19.6mA	21.1Ma
Consumed current when radio is off on microcontroller		185µA	245µA
Consumed current when radio is idle on microcontroller		52.5µA	178Ma
Consumed current when radio is listening on microcontroller		5.2µA	21.2Ma

5.5 Consumption of RAM and ROM

One of the constraints of the IoT devices are the node's memory. Normally IoT nodes are usually embedded and therefore have small size with affordable price, therefore their memory is much smaller when compared to the personal computer's memory. The size of the Tmote sky mote used in this paper is 48kb flash and 10kb of RAM, it is therefore expedient to determine the percentage of the mote's memory that would be occupied by the IDS code. The results for power expended and consumed energy by the Tmote are shown in the next section

5.6 Hardware Testbed Analysis

The experimental testbed used for the analysis has the following specification, one edge router node, five intrusion detection nodes and three adversarial nodes. The motes were simulated as follows, adversarial nodes repeatedly send harmful data packets to the edge router. The transmission speed of the motes were varied i.e. 1, 10, 100 and 1000 packets per second. In addition to this, the adversarial nodes send one or two threat data patterns to the edge router. The intrusion detecting nodes checks the payload of all the data packets being transmitted across them for any threat patterns, now if a threat pattern is observed, the intrusion detecting nodes gathers information about these threat patterns, i.e. the origin and destination IP addresses and the number of the port from which the threat pattern emanated together with the signature of the attack. This information will e aggregated by the intrusion detecting system nodes to form an alert packet, which are invariably transmitted to the edge router to be processed further for full threat detection. The simulation was done in two variations, (i) intrusion detecting system nodes forward alert control packet to the edge router whenever adversarial data packets having a particular traffic pattern was detected. For the purpose of this experimental setup, IDS node 4 is assumed to forward the adversarial data packet. (ii) the second instance is when IDS node 9 sends the adversarial data packet. In each case, the consumed power and memory consumption by the Tmote is measured.

From the experiments carried out, the action of the edge router whenever threat pattern are transmitted to it was analysed, these actions include, the amount of alerts, the origin of alert and time it occurred. The value of this information will enable the edge router from a correlation between the source of alert and appropriate actions to be taken. For example, the confidence value and the correlation coefficient will determine if such adversarial nodes will be eliminated from the network. The expended power and memory consumption (overhead) was measured with respect to the added time the CPU was active during data packet transmission.

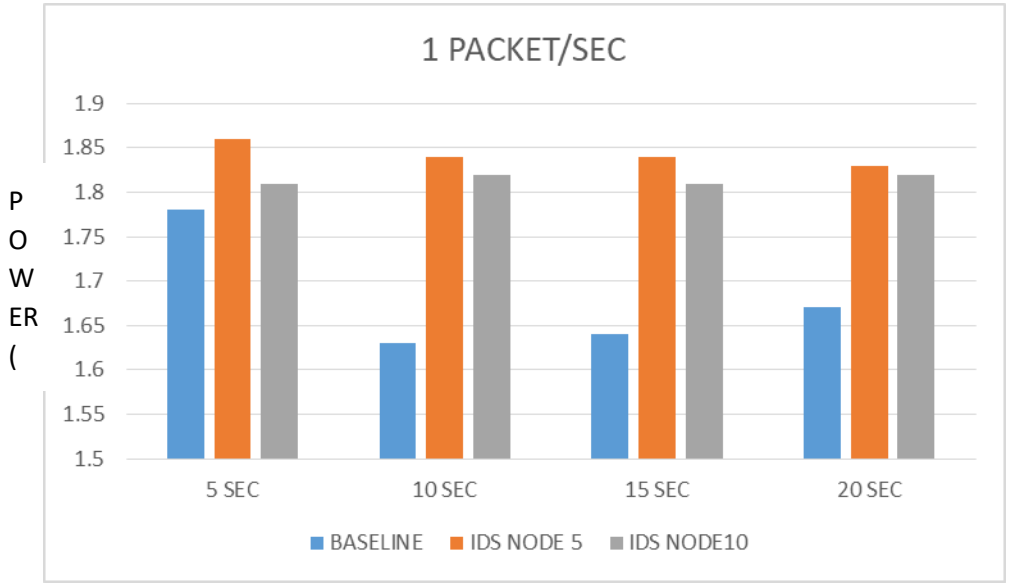

Figure 7a: IDS node Power Consumption with respect to time

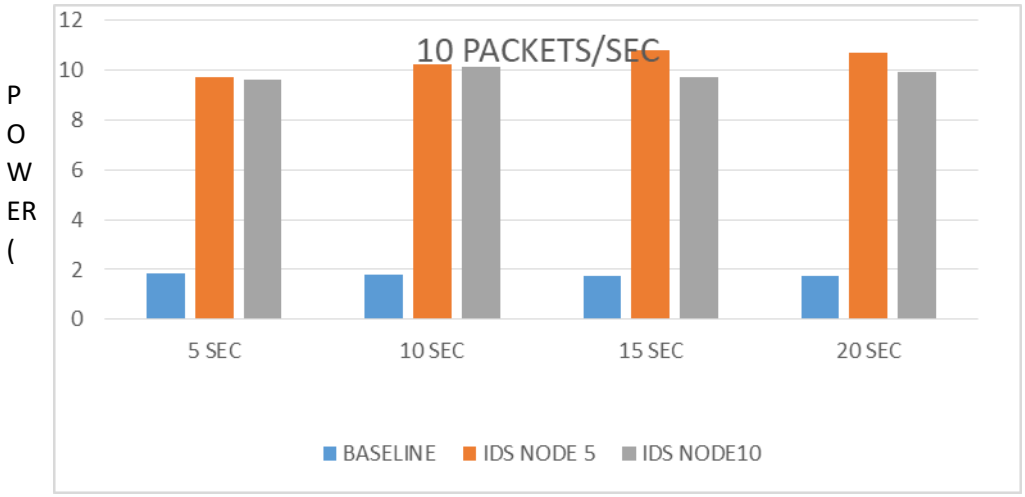

Figure 7b: IDS node Power Consumption with respect to time

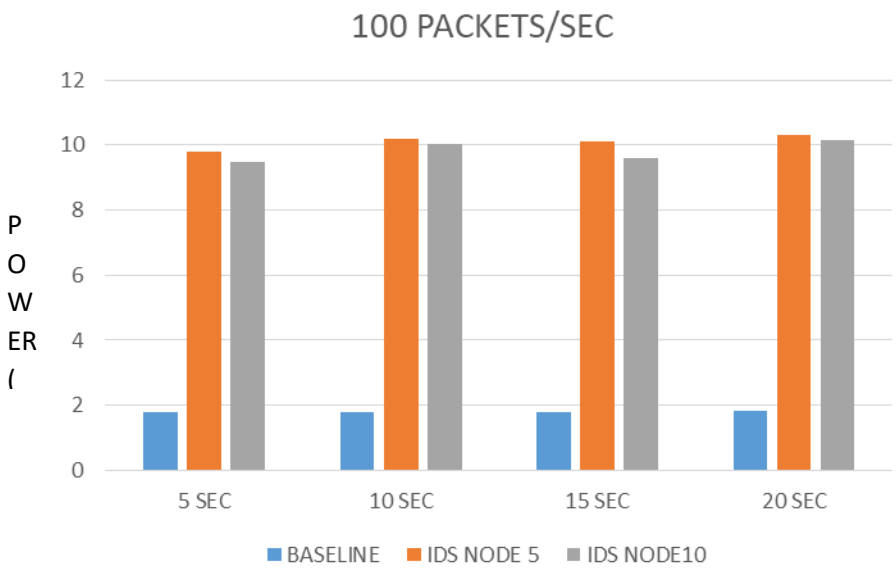

Figure 7c: IDS node Power Consumption with respect to time

The power consumption of the edge router's CPU in relation with varying rate of transmission of the adversarial nodes was shown in Figure 7. It was observed the edge router's CPU consumed power does not change considerably among any two IDS nodes irrespective of their rate of transmission. The consumed power of the edge router's CPU in relation with the number of alerts received was measured. It was observed from Figure 7 that the consumed energy of the IDS nodes does not change appreciably with increased number of alerts

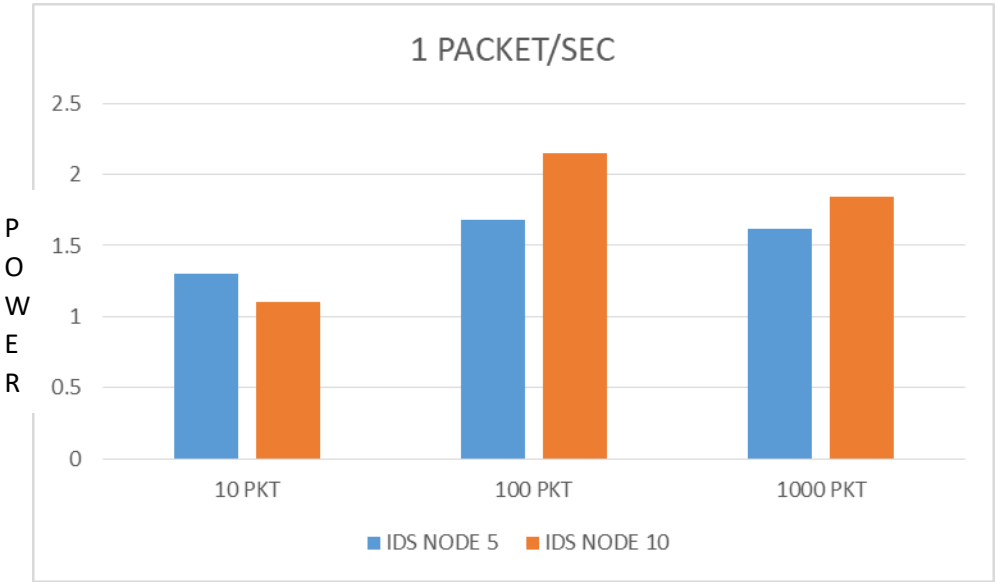

Figure 8: Power Consumption of an IDS node power consumption with respect to received number of adversarial packets

Figure 8 illustrates the consumed power of the intrusion detecting nodes for these times i.e. 5, 10, 15 and 20 minutes running time. From Figure 8, it was observed that the consumed power progressively reduces as the size of IDS alert from the IDS nodes increase. This can be attributed to the reduced size of the data packets resulting from higher packet size of alert control packets as the communication channel has a fixed bandwidth, i.e. as the alert control packets increases the actual data packets reduces and vice-versa.

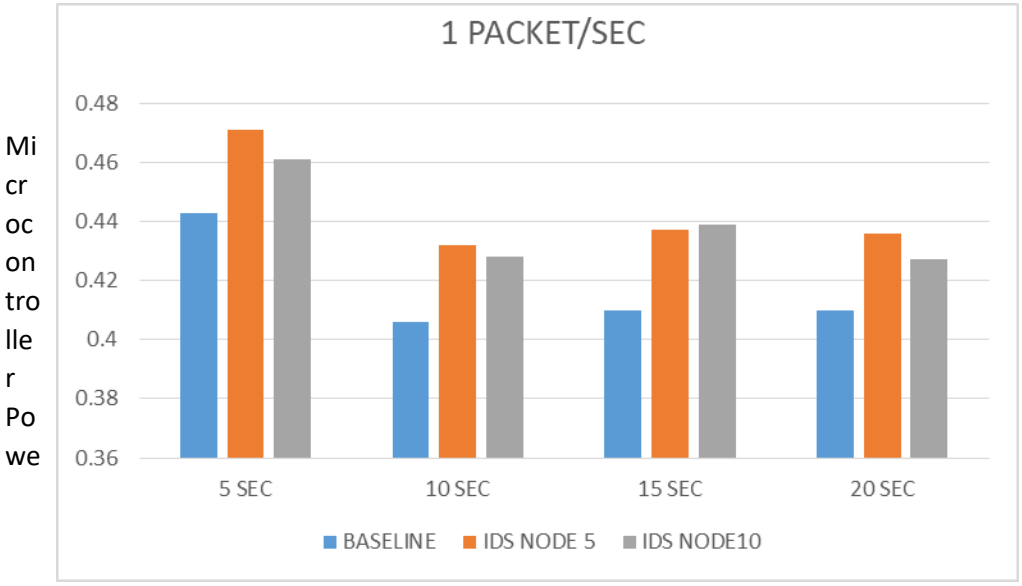

Figure 9a: Power consumption of microcontroller with respect to time.

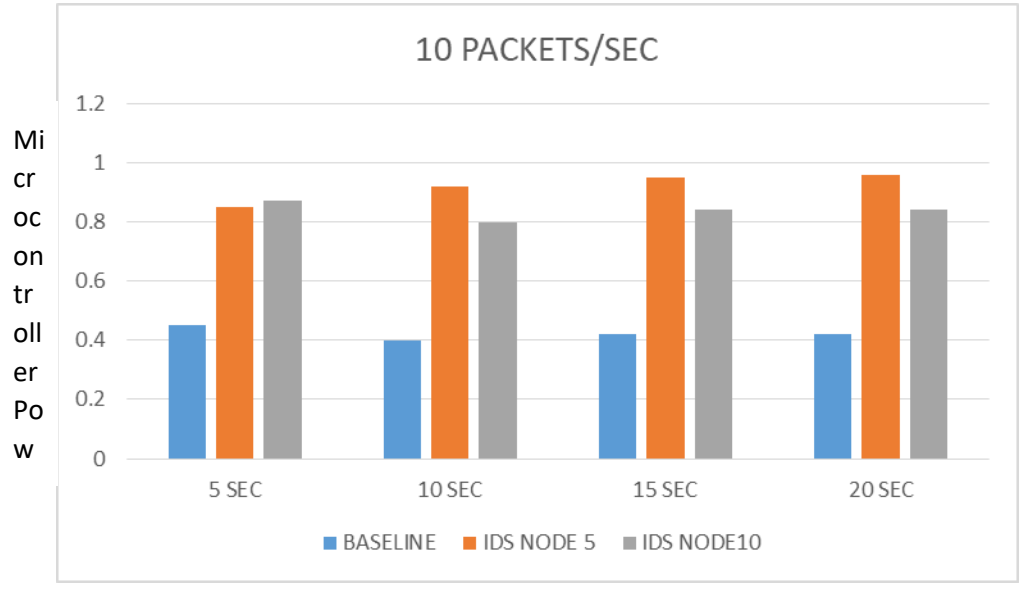

Figure 9b: Power consumption of microcontroller with respect to time

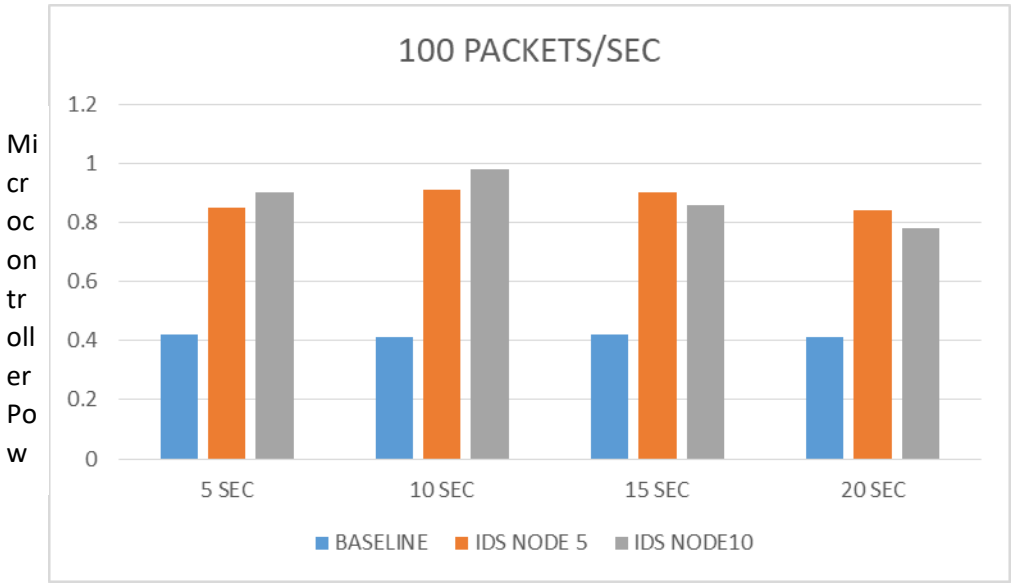

Figure 9c: Power consumption of microcontroller with respect to time

Secondly, the consumed power of the Tmote in relation with the number of adversarial data packets was measured. The result for the consumed power for 5, 10, 100 and 1000 adversarial data packets received is shown in Figure 9. It was observed that the consumed power analysis show similarity with the results in Figure 7 where the consumed power of the intrusion detecting nodes with running time was analysed.

Table 7: IDS functionality induced memory overhead to both IDS node and the Edge router.

	Intrusion Detection size	**Overhead on RAM**	**Overhead on ROM**
IDS Node	5	370	275
	10	730	275
Edge Router	5	200	210
	10	380	210

The memory overhead occasioned by the inclusion of the IDS protocol on the IDS nodes is shown in Table 7. It can be observed from table 7 that the overhead due to ROM memory consumption did not change appreciably in relation to an increase with the IDS alert size, this can be attributed to the fact the ROM program is not dependent on the IDS alert size. On the other hand, the RAM consumption was increased in relation with the IDS alert size, as information concerning the alert is usually saved in memory

6. CONCLUSION

An anomaly detection system was proposed in this paper for detection of threats in IIoT networks using data from TCP/IP packets. It works on unsupervised deep learning techniques which are variants of rule based selection strategy. The model uses a combination of the variational auto-encoder with the recursive feature elimination for reducing the size of the datasets, this subset of the dataset serves as a good representation for unsupervised learning of standard network. The proposed VAE-DFBNN using recursive selection strategy has capability to extract important features from the datasets necessary for detecting anomaly in IIoT networks. By using the important subset of the datasets features, the detection rate was higher as well as the detection accuracy as there was no problem of dataset redundancy,

When the VAE-DFBNN model proposed in this paper was compared to other recent intrusion detection techniques, it has the highest detection rate (DR) of 99.2% and the lowest incorrect prediction of 1.0%. It has an improvement of 7% on the closest performing technique of DAE-DFFNN when tested on both NSL-KDD and NSW-NB15 datasets which is being used by many researchers in intrusion detection design. The use of the recursive feature elimination was to select the most appropriate dataset features necessary for the detection of threats in IIoT networks. This together with the variational auto-encoder technique helps to improve the threat detection rate and decrease the detection latency. For the purpose of future analysis, a testbed will be designed using actual data generated from IIoT networks. An attempt will also be made to combine more than one technique of data intrusion for enhanced threat detection performance.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

REFERENCES

1. Abdulraheem M, Awotunde J. B, Jimoh R. G and Oladipo, I. D (2022) “An efficient lightweight cryptographic algorithm for IoT security,” in Communications in Computer and Information Science, pp. 444–456, Springer..
2. Adeniyi E. A, Ogundokun R. O and Awotunde J. B (2022) “IoMT- based wearable body sensors network healthcare monitoring system,” in IoT in Healthcare and Ambient Assisted Living, pp. 103–121, Springer, Singapore.
3. Ambika P (2021) “Machine learning and deep learning algorithms on the Industrial Internet of Things (IIoT),” Advances in Computers, vol. 117, no. 1, pp. 321–338,

4. Amit K. and Chinmay C. (2022) "Artificial intelligence and Internet of Things based healthcare 4.0 monitoring system," Wireless Personal Communications, pp. 1–14.
5. Ashfaq R. A. R , Wang X, Z, Huang J. Z, Abbas H, and He Y. L, (2018) "Fuzziness based semi-supervised learning approach for ntrusion detection system," Information Sciences, vol. 378, pp. 484–492.
6. Ashima R Haleem A, Bahl S, Javaid M, Mahla S. K and Singh S (2022) "Automation and manufacturing of smart materials in Additive Manufacturing technologies using the Internet of Things towards the adoption of Industry 4.0," Materials Today: Proceedings, vol. 45, pp. 5081–5088.
7. Ayo F. E. Folorunso S. O, Abayomi-Alli A. A, Adekunle A. O and Awotunde J. B (2021) "Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection," Information Security Journal: A Global Perspective, vol. 29, no. 6, pp. 267–283.
8. Bakhtawar A, Abdul R. J, Chinmay C, Jamel N, Saira R, and Muhammad R, (2022) "Blockchain and ANFIS empowered IoMT application for privacy preserved contact tracing in COVID-19 pandemic," Personal and Ubiquitous Computing.
9. Cecchinel C, Jimenez M, Mosser S, and Riveill M, (2017) "An architecture to support the collection of big data in the internet of Wireless Communications and Mobile Computing things," in 2017 IEEE World Congress on Services, pp. 442–449, Anchorage, AK, USA.
10. Chakraborty B and Kawamura A, (2020) "A new penalty-based wrapper fitness function for feature subset selection with evolutionary algorithms," Journal of Information and Telecommunication, vol. 2, no. 2, pp. 1–18.
11. Chen R, Liu C. M, and Chen C, (2015) "An artificial immune-based distributed intrusion detection model for the internet ofbthings," in Advanced materials research, pp. 165–168, Trans Tech Publications Ltd.
12. Cruz T, Rosa L and Proenca J , (2019) "A cybersecurity detection framework for supervisory control and data acquisition systems," IEEE Transactions on Industrial Informatics, vol. 12, no. 6, pp. 2236–2246.
13. Dash S, Chakraborty C, Giri S. K, Pani S. K, and Frnda J,*2023) "BIFM: big-data driven intelligent forecasting model for COVID-19," IEEE Access, vol. 9, pp. 97505–97517.
14. Enache A. C and Sgârciu V, (2017) "Anomaly intrusions detection based on support vector machines with an improved bat algorithm," in 2017 22nd International

Conference on Control Systems and Computer Science, pp. 317–321, Bucharest, Romania.

15. Haghighi M. S, Farivar F, and Jolfaei A, (2023) "A machine learning-based approach to build zero false-positive IPSs for industrial IoT and CPS with a case study on power grids security," IEEE Transactions on Industry Applications, pp. 1–9.
16. Herrera-Semenets V, Andrés O, Pérez-García, R. Hernández-León, J. van den Berg, and Doerr C, (2019) "A data reduction strategy and its application on scan and backscatter detection using rule-based classifiers," Expert Systems with Applications, vol. 95, pp. 272–279.
17. Hodo E, X. Bellekens X and Hamilton A, (2019) "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2019 International Symposium on Networks, Computers and Communications (ISNCC), pp. 1–6, Yasmine Hammamet, Tunisia.
18. Hu Y, Yang A, Li H, Sun Y, and Sun L, (2021) "A survey of intrusion detection on industrial control systems," International Journal of Distributed Sensor Networks, vol. 14, no. 8.
19. Kabir M. R, Onik A. R, and Samad T, (2019) "A network intrusion detection framework based on Bayesian network using a wrapper approach," International Journal of Computer Applications, vol. 166, no. 4, pp. 13–17.
20. Khan P. W and Byun Y, (2022) "A blockchain-based secure image encryption scheme for the industrial Internet of Things,"Entropy, vol. 22, no. 2, p. 175 - 187.
21. Kirnapure W. K and Patil A. R. B, (2018) "Classification, detection and prevention of network attacks using rule based approach," International Research Journal of Engineering and Technology, vol. 4, no. 4, pp. 1453–1459.
22. Kushner D, (2015) "The real story of stuxnet," IEEE Spectrum, vol. 50, no. 3, pp. 48–53.
23. Leonard, L. C. 2018) "Web-based behavioural modelling for continuous user authentication (CUA)," in Advances in Computers, pp. 1–44,
24. Linda O, Vollmer T, and Manic M, (2012) "Neural network-based intrusion detection system for critical infrastructures," in 2012 International Joint Conference on Neural Networks, pp. 1827–1834, Atlanta, GA, USA.
25. Li Y, Ma R, and Jiao R, (2018) "A hybrid malicious code detection method based on deep learning," International Journal of Security and Its Applications, vol. 9, no. 5, pp. 205–216.

26. Maglaras. and Jiang A, (2017) “Intrusion detection in SCADA systems using machine learning techniques,” in 2017 Science and Information Conference, pp. 626–631, London, UK.
27. Maglaras L. A and Jiang J, (2017) “OCSVM model combined with k- means recursive clustering for intrusion detection in scada systems,” in 12th International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, pp. 133-134, Rhodes, Greece.
28. Maglaras L/ A, Jiang J, and Cruz T. J, (2019) “Combining ensemble methods and social network metrics for improving accuracy of OCSVM on intrusion detection in SCADA systems,” Journal of Information Security and Applications, vol. 30, pp. 15–26.
29. Marsden T, Moustafa N, Sitnikova E, and Creech G, (2020) “Probability risk identification based intrusion detection system for SCADA systems,” in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 353–363, Springer, Cham, Switzerland,.
30. Moustafa N, Hu J, and Slay J, (2022) “Machine learning models for secure data analytics: a taxonomy and threat model,” Computer Communications, vol. 153, pp. 406–440.
31. Moustafa N, Hu J, and Slay J, (2022) “A holistic review of network anomaly detection systems: a comprehensive survey,” Journal of Network and Computer Applications, vol. 128, pp. 33–55.
32. Muna A. H, Moustafa N, and Sitnikova E, (2019) “Identification of malicious activities in industrial internet of things based on deep learning models,” Journal of information security and applications, vol. 41, pp. 1–11.
33. Rajendran R, S. Santhosh S. V. Kumar, Y. Palanichamy T, and Arputharaj K, (2019) “Detection of DoS attacks in cloud networks using intelligent rule based classification system,” Cluster Computing, vol. 22, no. S1, pp. 423–434.
34. Shang W, Cui J, Wan M, An P, and Zeng P, (2019) “Modbus comumunication behaviour modelling and SVM intrusion detection method,” in Proceedings of the 6th International Conference on Communication and Network Security pp. 80–85, Singapore.

    Shang W, Zeng P, Wan M, Li L, and An P, (2019) “Intrusion detection algorithm based on OCSVM in industrial control system,”Security and Communication Networks, Vol 9, no 10, pp 1049 – 1.

35. Sherasiya T, Upadhyay H and Patel H. B (2017) “A survey: intrusion detection system for internet of things,” International Journal of Computer Science and Engineering (IJCSE), vol. 5, no. 2, pp. 91–98.
36. Silva P and Schukat M, (2017) “On the use of k-nn in intrusion detection for industrial control systems,” in Proceedings of The IT&T 15th International Conference on Information Technology and Telecommunication, pp. 103–106, Dublin, Ireland.
37. Sivatha S, Geetha S, and Kannan A, (2014) “Decision tree based light weight intrusion detection using a wrapper approach,” Expert Systems with Applications, vol. 39, no. 1, pp. 129–141.
38. Snášel V, Nowaková J, Xhafa F, and Barolli L (2018) “Geometrical and topological approaches to Big Data,” Future Generation Computer Systems, vol. 67, pp. 286–296.
39. Soto J and Nogueira M, (2020) “A framework for resilient and secure spectrum sensing on cognitive radio networks,” Computer Networks, vol. 115, pp. 130–138. Stein G, Chen B, Wu A. S, and Hua K. A, (2008) “Decision tree classifier for network intrusion detection with GA-based feature selection,” in Proceedings of the 43rd annual southeast regional conference on - ACM-SE 43, pp. 136–141, Kennesaw, Georgia,. Stewart B, Rosa L and Maglaras L. A., (2020) “A novel intrusion detection mechanism for scada systems which automatically adapts to network topology changes,” EAI Endorsed Transactions on Industrial Networks and Intelligent Systems, vol. 4,no. 10. Van Dijk J. C and Williams P, (1995) “The history of artificial intelligence,” in Expert Systems in Auditing, pp. 21–26, Palgrave Macmillan, London.
40. Xenofontos C, Zografopoulos L, Konstantinou C, Jolfaei A, Khan M. K, and Choo K. (2022) “Consumer, commercial and industrial IoT (in) security: attack taxonomy and case studies,” IEEE Internet of Things Journal.
41. Yan Q and Yu F. R, (2018) “Distributed denial of service attacks in software-defined networking with cloud computing,” IEEE Communications Magazine, vol. 53, no. 4, pp. 52–59,.
42. Yin C, Zhu Y, Fei J, and He X, (2020) “A deep learning approach for intrusion detection using recurrent neural networks,” IEEE Access, vol. 5, pp. 21954–21961.
43. Zafar F, Khan A, and Malik S. U. R (2020) “A survey of cloud computing data integrity schemes: design challenges, taxonomy and future trends,” Computers & Security, vol. 65, pp. 29–49.

44. Zhang H, Yao D. D, Ramakrishnan N , and Zhang Z, (2018) “Causality reasoning about network events for detecting stealthy malware activities,” Computers & Security, vol. 58, pp. 180–198.