Deep Auto-encoder based Intrusion Detection with Machine Learning Classifiers

Dhanush Vasa
IIT2019208
6th Semester
IT

Mitta Lekhana Reddy
IIT2019204
6th Semester
IT

Meghana Santhoshi Kandagatla
IIB2019030
6th Semester
IT-BI

***Abstract*—The convergence of Industry 4.0 offers unparalleled manufacturing agility, optimizing operations and reducing waste through digitized machine connectivity. However, this transformation also ushers in heightened cyber threats, necessitating a reevaluation of security measures. Amid escalating cyberattacks, intrusion detection systems, notably Network IDS, become pivotal in mitigating advanced persistent threats overlooked by traditional safeguards. As these threats evolve, developing novel techniques for effective intrusion detection becomes imperative. This paper introduces an innovative approach to enhance the accuracy and performance of existing IoT network intrusion detection through a robust unsupervised learning method. The proposed research work employs deep auto-encoders (DAEs) to detect anomalies by minimizing reconstruction errors in encoding and decoding input data. The comparative analysis of two DAE types assesses their capability to accurately discern abnormal patterns while reconstructing standard network traffic. By evaluating three machine learning models (Random Forest, Support Vector Machine, and K-Nearest Neighbors) using the Stacked Non-Symmetric Auto Encoder Algorithm, our study aims to offer significant insights into diverse DAE architectures for effective anomaly detection. These findings are crucial in guiding the development of robust security solutions in the future.**

***Index Terms*—Intrusion detection,auto-encoders, Industrial IoT, Industry 4.0**

I. Introduction

A. *Industry 4.0:*

Industry 4.0, also known as the Fourth Industrial Revolution, is a term used to describe the integration of digital technologies into the manufacturing industry. One of the key benefits of Industry 4.0 is the ability to create a more flexible and agile manufacturing process. With smart machines and real-time data collection, companies can respond more quickly to changing market demands, adjust production processes on the fly, and optimize their supply chains. This level of flexibility can help companies reduce waste, improve efficiency, and minimize downtime.

Industry 4.0 technologies can also enable companies to improve their sustainability and environmental impact. By monitoring and analyzing energy usage and other resource consumption, companies can identify areas where they can reduce their environmental footprint. This not only benefits the environment but can also lead to cost savings through reduced energy and resource consumption.

Another key benefit of Industry 4.0 is the ability to create new business models and revenue streams. With the ability to collect and analyze large amounts of data, companies can gain insights into customer behavior and preferences, enabling them to create new products and services that better meet customer needs. This can also lead to new revenue streams, such as subscription-based services or product customization options.

However, the adoption of Industry 4.0 technologies also presents challenges. One of the biggest challenges is the need for skilled workers who are capable of operating and maintaining these new machines, as well as analyzing the data they generate. This requires companies to invest in training and education programs to ensure that their workforce has the necessary skills and knowledge.

B. *Cybersecurity Challanges in Industry 4.0:*

Another challenge is the need for cybersecurity measures to protect against the potential for cyber attacks on connected machines and networks. Companies need to ensure that their systems are secure and that they have contingency plans in place in the event of a breach.

Despite these challenges, the benefits of Industry 4.0 make it a compelling proposition for companies looking to stay competitive in the global marketplace. With the potential to improve efficiency, reduce costs, and create new business models and revenue streams, Industry 4.0 represents a major opportunity for companies that are willing to invest in the necessary technologies and adapt their business models accordingly.

C. *Intrusion Detection System (IDS):*

Intrusion Detection Systems (IDS) have evolved significantly over the years to keep up with the changing threat landscape. Today's IDS can detect advanced persistent threats (APTs), which are sophisticated attacks that can bypass traditional security measures such as firewalls and antivirus software. APTs often rely on social engineering techniques to trick users into downloading malware or revealing sensitive information. IDS can detect these attacks by analyzing network traffic and looking for unusual patterns or behavior.

IDS can also be used to monitor cloud-based environments, which are becoming increasingly popular among organizations. Cloud-based environments pose unique security challenges, as they are often accessed from multiple devices and locations. IDS can be used to monitor cloud-based environments and detect threats such as data breaches, unauthorized access, and malware infections.

Another benefit of IDS is that it can be used to detect zero-day vulnerabilities. Zero-day vulnerabilities are previously unknown vulnerabilities that can be exploited by attackers to gain access to a network or system. IDS can detect zero-day vulnerabilities by looking for unusual behavior or traffic patterns that may indicate an attack.

IDS can also be used to comply with regulatory requirements. Many industries are subject to strict regulatory requirements regarding data privacy and security. IDS can help organizations meet these requirements by providing real-time monitoring and incident response capabilities. IDS can also generate reports that can be used to demonstrate compliance with regulatory requirements.

IDS can also be used to reduce the impact of security incidents. When a security incident occurs, the key to minimizing the impact is to detect it early and respond quickly. IDS can help organizations achieve this by providing real-time alerts and automating incident response. IDS can also be used to isolate infected devices or networks to prevent the spread of malware.

Finally, IDS can be used to improve the overall security posture of an organization. By providing real-time monitoring and incident response capabilities, IDS can help organizations identify and mitigate security risks before they can be exploited by attackers. This can help organizations reduce the likelihood and impact of security incidents, improve compliance with regulatory requirements, and ultimately improve the security of their network and data.

In summary, Intrusion Detection Systems are a critical component of a comprehensive security strategy. IDS can detect advanced threats, monitor cloud-based environments, detect zero-day vulnerabilities, comply with regulatory requirements, reduce the impact of security incidents, and improve the overall security posture of an organization. While IDS has its limitations, it remains an essential tool for organizations looking to protect their networks and data from cyber threats.

D. Limitations and Proposed Solution:

Machine learning approaches have been widely used to detect network intrusions in recent years. However, these approaches have limitations, such as the requirement for human experts to process data, identify important data and patterns, and train models. This has led to the development of deep learning approaches, which can automatically learn complex patterns and correlations from input data without human intervention. The proposed use of stacked non-symmetric autoencoders for NIDS is a prime example of this deep-learning approach.

E. Benefits of the Proposed Model and Key Research Contributions:

Stacked non-symmetric autoencoders can learn a compressed representation of the input data by encoding and decoding it. This technique can be applied for intrusion detection by using the encoded representation of the input data to identify anomalies by learning the normal behavior of the network and detecting any deviations from it.

The proposed model also helps to improve the accuracy of NIDS while reducing the false positive rate. False positives are a significant challenge for most intrusion detection systems, as they can cause unnecessary alerts which leads to a waste of time and resources. The proposed NIDS can be implemented in different industries and applications, such as banking, healthcare, and e-commerce, to safeguard sensitive data from cyber threats.

As cyber threats continue to evolve, it is essential to develop new and effective methods for detecting and preventing network intrusions. The proposed NIDS provides a more efficient and accurate approach for intrusion detection that can benefit various industries in the integration with Industry 4.0.

F. Manuscript Organization:

The present research paper is arranged in the following manner: Section II duly presents the contextual information pertaining to the models that were adroitly employed. In Section III, a meticulous explication of the findings and observations of related studies has been provided. A comprehensive account of the dataset is discussed in Section IV. The proposed methodology is outlined in Section V. The results and experimental setup have been cogently discussed in Section VI. Lastly, Section VII discusses the culminating conclusion.

II. Background

Complex relationships and concepts requiring multi-level representations can be handled by the models under consideration. Using supervised and unsupervised learning techniques to construct abstraction layers, these models are able to encapsulate abstract characteristics of the input. This enables the models to discover previously concealed significant patterns and connections in the data. These abstraction layers frequently include highly interpretable low-level output functions, allowing scientists to better comprehend how a model derives its predictions. The models' combination of high-level abstraction and low-level interpretability renders them applicable to a vast array of data analysis tasks.

Autoencoder and stacked autoencoder are prominent neural network topologies that excel at unsupervised learning and feature extraction. Autoencoders are a type of feedforward neural network that can be trained to compress and then reconstruct input data, whereas stacked autoencoders are a type of deep learning model that consists of numerous layers of autoencoders, enabling even more complex data representation and abstraction. Both techniques can be utilised to analyse images and signals, detect anomalies, and compress data more efficiently.

A. Auto-Encoder:an unsupervised learning technique

The objective of auto encoders, which are a type of neural network model, is to produce an output (x) that is highly accurate with respect to the input (x). In the majority of instances, component elements consist of an input layer, an output layer, and a concealed layer. The output of the auto encoder is obtained by transmitting the input data through a concealed layer, where it is compressed and shrunk in size using an encoder function denoted f(x). The decoder function d(x), which also accepts the data in their low-dimensional form, transforms the data from their low-dimensional form to their high-dimensional form. The reconstruction error L(x,d(f(x)) is computed by subtracting the data that was input from the data that was reconstructed. Using back propagation, the network can be refreshed and errors like this one reduced.

Encoders and decoders are utilised by auto encoder models to reduce the dimensionality of the data. This may facilitate the identification of prominent characteristics that distinguish the data distribution. Because of this, auto encoders are most effective in situations that require unsupervised learning and the extraction of features. Due to the reduced dimensionality of the concealed layer, auto encoders perform exceptionally well in applications such as image and signal processing, anomaly detection, and data compression. Because auto encoders can learn meaningful representations of data and compress that data effectively, they have applications in a variety of research and business fields.

B. Stacked Auto-Encoder

Since the deep auto-encoder consists of two symmetrical deep-belief networks, it is more complex than a traditional auto-encoder. Each of the encoding and decoding layers in a typical shallow architecture has between four and five levels. The research of Hinton and Salakhutdinov suggests that a deep auto-encoder may be used to effectively convert high-dimensional input into a low-dimensional representation. To do this, a higher-dimensional encoder might be used.

The stacked auto-encoder method uses deep learning to construct hidden structures with both one and several layers. It is feasible to improve accuracy while decreasing processing time and training data needs by raising the network's depth. First-order characteristics of the raw input data are typically learned by the first layer of an auto-encoder. The first-order features are used to build a pattern recognition framework, which is then used to learn the second-order features. Ultimately, the network's subsequent layers are responsible for imparting higher-level characteristics onto it. The technique's capacity to acquire broader and more complex attributes is directly responsible for the model's much improved performance in all scenarios.

III. Related Work

To highlight the large reconstruction error and additional training time associated with employing SNDAE as a feature extraction approach for intrusion detection, Zhaojun et al. [1] presented an Adam Nonsymmetric Deep Autoencoder (ANDAE) based on SNDAE with an Adam Optimizer. The low-dimensional noteworthy characteristics acquired by ANDAE are then classified using a random forest. intrusion detection.

Ghulam Muhammad et al.[2] suggested an intrusion detection system (IDS) in which sandwiched AE unsupervised learns the features of the input network record to reduce feature width.

Lee et al. [3] by combining pre-training, dropout and error forgetting, developed a deep neural network-based approach to simplify self-training. A comparison with conventional methods has demonstrated that throughout different use cases the approach increases accuracy by up to 14%. in different use cases. They also concluded that among the SdAs analysed (1–4 layers) those with 4 layers produced the efficient results.

Binghao Yan et al. [4] used a stacked sparse autoencoder to extract appropriate features. Used a deep learning method to extract high-level feature representations of invasive data. These low-dimensional features are classified with classifiers. This method proved to give feasible and efficient feature extraction compared to other techniques.

J. Zhang et al. [5] They presented new systematic frameworks for misusing, anomalous, and hybrid-network-based IDSs that leverage a data mining approach called random forests. The random forests technique is used to build patterns of incursions automatically over training data in abuse detection. We apply the abuse technique with the created patterns to detect invasions across the test set. The attack prediction rate is 94.2 percent along with the 1.1 percent as its FP rate.%

Hossain et al. [6] devised a strategy for enhancing the efficiency of clustering and classification techniques in order to increase IDS performance. The goal of this research is to improve the KNN classifier in an existing intrusion detection problem that involves K-MEANS,clustering and KNN classification.

As In the course of the research, it was discovered that the basic architecture of DL, unknown attacks were not detected by the security systems. Quite high false positives and false negatives occurred even if they did. To tackle this problem K Saurabh et al. [9] introduced a methodology using the variations of LSTM, which is a special form of RNN, since the most distinguishing feature of it is to keep the information for later use in the environment. In this case, they can handle time series data, which changes over time. The proposed method of this paper out performes the classic ML approaches, along with the extant approaches of Deep Neural Network.

It is possible, however, to use simple DL-based methods While classifying attacks, IDSs face a number of inherent challenges. A system for detecting intrusions in the Internet of Things. Dataset with high number of features and high amount of data increases the time required to train on a DL model. Some features can negatively impact the training process of the model which reduces the model's attack detection capability.

K Saurabh et al. [10] proposed an approach a lightweight and optimized Artificial Neural Network (ANN) based Distributed Denial of Services (DDoS) attack detection framework with mutual correlation as a feature selection method that produces a the superior result when compared with Long Short Term Memory (LSTM) and simple ANN.

Firuz Kamalov et al. [8] proposed auto encoder based intrusion detection system for detecting DDoS attacks.Essentially, the proposed approach uses the fact that anomalous traffic flows will have a higher reconstruction loss that can be used to flag intruders.

IV. Data Set Description

NetFlow records were extracted from the ToN-IoT dataset's pcaps to create the NF-ToN-IoT-v2 dataset [11]. Applying NetFlow analysis to the raw pcap files yielded a database of records suitable for network traffic analysis and associated research. The pcaps that were used to create the NetFlow records may be found online, making the dataset fully reproducible and open to scrutiny. With its comprehensive and trustworthy depiction of network traffic patterns and behaviour, the NF-ToN-IoT-v2 dataset is an invaluable tool for academics and practitioners in the area of IoT network analysis.

ToN-IoT datasets are state-of-the-art data samples from the IoT and IIoT that may be used to test out different artificial intelligence (AI) based cybersecurity approaches. Here you may find ToN-IoT datasets. This information is generated by a broad variety of systems and applications, including computer operating systems, transport layer security, network traffic, and sensors built into IoT and IIoT devices. We were able to gather information from a large number of IoT networks by using a processing method known as parallel processing. The data collected is useful for more than just spotting prospective break-ins; it also records a broad range of regular activities. [12]

In order to test and evaluate different security solutions in the context of IoT and IIoT networks, academics and practitioners in the area of cybersecurity have a useful resource in ToN-IoT datasets. Using AI for the rapid and accurate analysis of massive datasets and by combining data from a range of sources may guarantee a complete and nuanced understanding of network activities. There has been a major step forward in the study of cybersecurity thanks to the ToN-IoT datasets. This is because they include a wealth of information from which novel inferences might be drawn.

The dataset's quality for future analysis and classification was improved by adding an extra label feature and a type feature. When applied to an event like a cyberattack, the label function will reveal whether or not it was beneficial. Cyberattacks that went unnoticed earlier may now be found because to the type feature's ability to categorise occurrences into several groups.

Nine such attacks were carried out by the researchers during the course of this investigation on a wide range of IoT and IIoT devices. More insight into the network's activity could be gleaned via including label and observation type data. With this data in hand, we can strengthen our defences against cyber threats to these vital infrastructures. threats.

A. NF-ToN-IoT-v2 consists of the following attacks:

[13]

- **Benign** - generally, information or activity that poses no threat or danger. This might comprise regular network traffic and data transfers between Internet of Things devices and servers. Some examples of benign traffic include updates, configuration changes, and other operations required for IoT devices to function as intended. [13] [17]
- **Backdoor** - is a method of access to a device or system that has been purposefully concealed or is not documented. The creation of a backdoor so that repairmen may get access to a device for routine maintenance and troubleshooting is an example of a lawful application. However, they may also be exploited for harm, such as when an attacker gains access to a system or network where sensitive information is stored. [13] [17]
- **DoS** (Denial-of-Service) - is a kind of cyberattack in which the target is inundated with so much data or traffic that it becomes unusable to its intended users. [13] [17]
- **DDoS** (Distributed Denial-of-Service) - is a kind of cyber assault in which the attacker utilises a network of hacked devices (a botnet) to overload a target system with traffic or requests, making it inaccessible to normal users. [13] [17]
- **Injection** - entail inserting malicious code or data to obtain unauthorised access, execute unauthorised instructions, or steal sensitive data by exploiting vulnerabilities in IoT devices, networks, or apps. SQL injection, command injection, and code injection are just a few examples of injection attacks. [13] [17]
- **MITM** (Man-in-the-Middle) - An attack is a sort of cyberattack in which one party attempts to spy on or steal information from another by manipulating or intercepting their communication. [13] [17]
- **Password** - used to safeguard entry to the devices themselves, the IoT network, or the web-based interfaces via which the devices are managed. However, hackers may easily obtain access to IoT devices and networks by using weak or default passwords. [13] [17]
- **Ransomware** - Data or firmware on susceptible Internet of Things devices, such as smart home gadgets, industrial control systems, or medical equipment, may be encrypted in an attack. The user might be locked out of their home automation system, vital infrastructure could be disrupted, and patient safety could be jeopardised in healthcare settings. [13] [17]
- **Scanning** - means locating and charting Internet of Things (IoT) devices and networks in order to understand their structure and potential weak spots. Security audits, device detection, and stock taking are just some of the many uses for scanning. [13] [17]
- **XSS** (Cross-Site Scripting) - is a kind of cyberattack in which harmful code is inserted into legitimate web sites visited by unsuspecting victims. Web-based interfaces for managing and controlling Internet of Things devices, including smart home systems or industrial control systems, are vulnerable to XSS attacks. [13] [17]

TABLE I
COMPARISON OF VARIOUS INTRUSION DETECTION APPROACHES

S. No.	Author	Proposed Work	Methodology Used	Advantages	Research Gap
1	Zhaojun et al. [1]	Adam Nonsymmetric Deep Autoencoder (ANDAE)	SNDAE with Adam Optimizer	Low-dimensional features, Random Forest	Reconstruction error, Training time
2	Ghulam Muhammad et al. [2]	IDS with Sandwiched AE	Unsupervised feature learning	Feature reduction	Feature width
3	Lee et al. [3]	Deep Neural Network	Pre-training, dropout, error forgetting	Improved accuracy	Layer optimization
4	Binghao Yan et al. [4]	Stacked Sparse Autoencoder	Deep learning	Efficient feature extraction	Comparison to other techniques
5	J. Zhang et al. [5]	Data Mining IDS with Random Forests	Data mining for abuse detection	High attack prediction rate	Systematic frameworks
6	Hossain et al. [6]	Enhanced Clustering and KNN	K-MEANS clustering and KNN classification	Improved IDS performance	Clustering and classification
7	K Saurabh et al. [9]	Variations of LSTM	Special form of RNN	Outperforms classic ML approaches	Handling time series data
8	K Saurabh et al. [10]	ANN-based DDoS Detection	Feature selection and optimization	Superior result compared to LSTM	Lightweight approach
9	Firuz Kamalov et al. [8]	Autoencoder-based IDS	Detecting DDoS attacks	Flags intruders based on reconstruction loss	Anomalous traffic flows

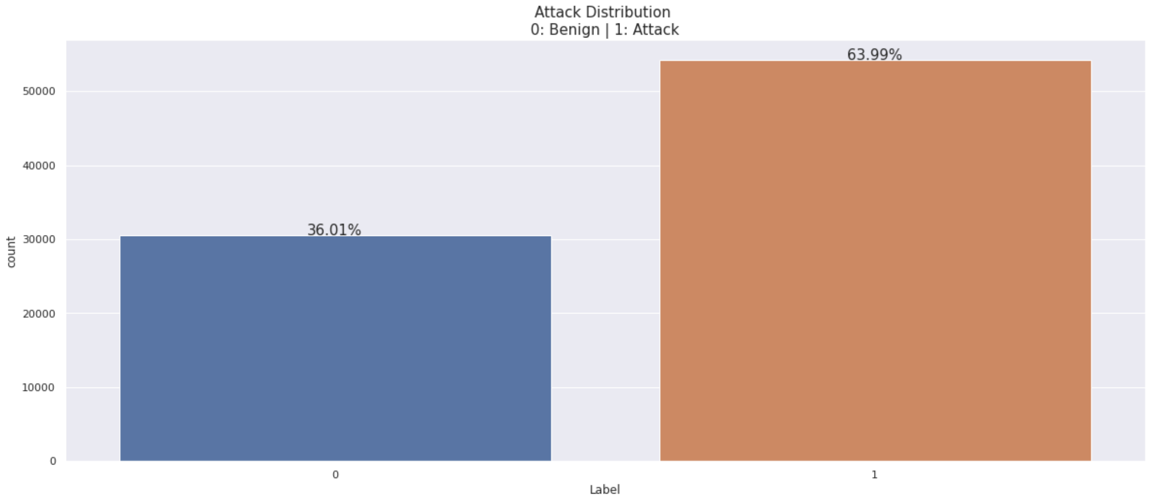


Fig. 1. Benign vs Attack

B. Dataset Comparision

V. PROPOSED METHODOLOGY

We investigate the performance of non-symmetric and stack non-symmetric deep auto-encoders (DAEs) for anomaly detection in IoT networks. By simulating normal network activity, we will evaluate the ability of two DAEs to detect and report anomalies in IoT network traffic and compare their performance.

In recent years, the use of deep auto-encoders (DAEs) has exploded for tasks as diverse as feature extraction, data compression, and anomaly detection. DAEs are neural networks designed to learn to encode data into a lower-dimensional representation and decode it back into the original data while minimising reconstruction error.

DAEs can be categorised as either symmetric or asymmetric layers. One encoding layer, one decoding layer, and a bottleneck layer distinguish non-symmetric DAEs. In contrast, the encoding and decoding levels of a symmetric DAE stack are positioned symmetrically relative to the bottleneck layer.

Despite the prevalence of stack symmetric DAEs in supervised learning and image recognition applications, non-symmetric DAEs are frequently employed when learning fea-

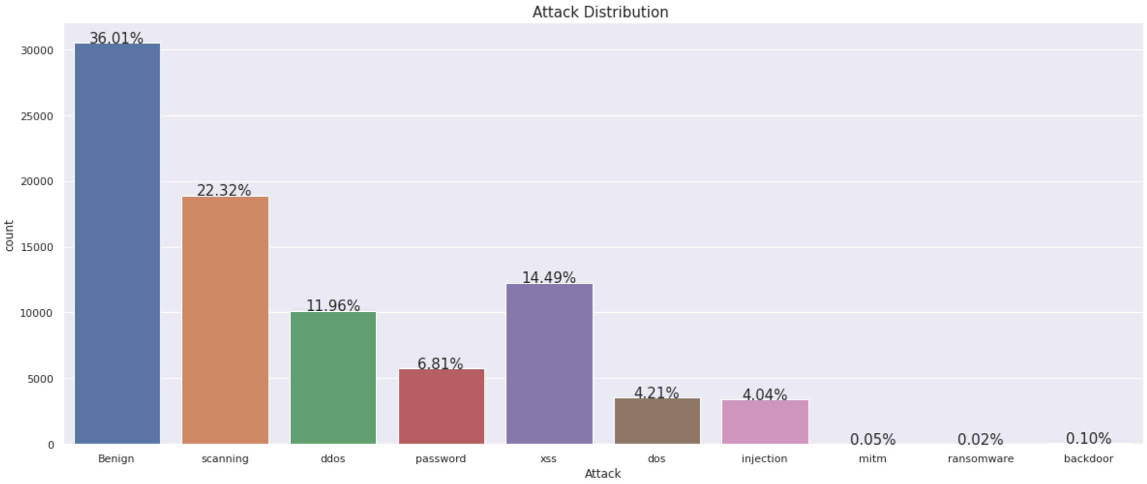


Fig. 2. Attack Distribution

TABLE II
COMPARISON OF DATASETS FOR ATTACK DETECTION IN INDUSTRIAL IOT

Dataset	**Description**	**No. of Features**	**Size**	**Attack Scenarios**	**Data Collection Period**	**Source**
NF ToN-IoT V2	A comprehensive dataset specifically designed for IIoT networks, capturing realistic industrial control systems (ICS) network traffic.	80+	Large	8	2021	CARRV Research Group, University of New Brunswick
CICIDS2017	A dataset based on real network traffic captured in an enterprise network environment.	80+	Large	16	2017	Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick
UNSW-NB15	A network traffic dataset collected from a controlled network environment.	42	Large	9	2015	University of New South Wales (UNSW)
IIoT-IDS	A dataset specifically designed for IIoT network intrusion detection.	30+	Moderate	12	2021	University of Tartu and Tallinn University of Technology
AID 2.0	A large-scale IoT dataset capturing multiple IoT protocols, including MQTT, CoAP, and HTTP.	20+	Very Large	7	2021	Shanghai Jiao Tong University
IoTPOT	A dataset comprising various IoT protocols and attack scenarios, collected from a real-world IoT honeypot.	22	Moderate	6	2020	CARRV Research Group, University of New Brunswick
IoT-23	A comprehensive dataset that covers various IoT device types and different network traffic patterns.	Varies	Large	9	2016-2017	Networked Systems Security Group, University of Twente
N-BaIoT	A dataset specifically designed for IoT network intrusion detection.	28	Moderate	5	2018	Institut Mines-Telecom, France
WADI	A dataset collected from a water treatment testbed that includes both normal and attack traffic.	11	Large	3	2017	Qatar University

tures in an unsupervised environment. Non-symmetric DAEs are simpler to train and comprehend, whereas stack-symmetric DAEs may be capable of capturing more complex feature hierarchies.

section¿Auto-Encoder, Non-Symmetric Deep/section¿ As the frequency and intricacy of attacks increase, it becomes more difficult for human operators to monitor and protect network infrastructure. To prepare NIDS for the future, it is essential to reduce reliance on human operators by automating attack detection and response.

We present a technique that has the potential to enhance the accuracy and efficacy of extant feature learning methods in NIDS by providing a reliable unsupervised learning method. Deep auto-encoders (DAEs), which are neural networks that can be trained to encode and decode input data with minimal reconstruction error, form the basis of the proposed method.

The objective is to use non-symmetric DAEs in an unsupervised environment to discover feature representations of network traffic data. These feature representations may be employed to train classifiers to recognise anomalies in network data. Due to unsupervised learning's reduced reliance on labeled data, our technique can be applied to larger datasets and utilised in more generic network topologies.

The proposed method has the potential to considerably enhance the accuracy and performance of existing feature learning algorithms and reduce the need for human personnel to manually deploy and update network intrusion detection systems (NIDS). As a result, NIDS systems may be better able to withstand future cyberattacks because they can adapt to them automatically and with little human intervention.

NDAE and comparable neural network designs are used to extract features from high-dimensional input data. In contrast to the two-phase approach used by conventional deep autoencoders, the ”asymmetric” design only employs the encoder phase. By encoding the input data in a lower-dimensional space, the network could learn to extract more valuable information from the input data.

Training each layer of the NDAE architecture to extract more abstract and generalizable features from the input data yields a hierarchical representation of the data. Because of this, it is an excellent instrument for unsupervised feature learning in complex datasets, such as those frequently encountered in IoT systems. If you lack access to labelled training samples, you can still utilise NDAE as an unsupervised feature extractor to gain insights from your data.

NDAE minimises additional labour by focusing solely on the encoding stage. Therefore, intrusion detection in real-time IoT networks becomes more accurate and efficient. Similar to traditional deep autoencoders, the NDAE training strategy aims to minimise the reconstruction error between the input data and its encoded representation.

The NDAE architecture is a potent instrument for unsupervised IoT network feature learning. The method's structure for rapid learning, precision, and ability to construct hierarchical representations of input data make it a promising candidate for use in intrusion detection systems.

The latent representations are calculated via NDAE $h_i \in R^{di}$ (with d being the vector's dimension) given an input vector x in Rd, using the following deterministic function description.

$$h_i = g(W_i.h_i - 1 + b_i); i = 1,2,3,4,5,6,7,8...n \qquad (1)$$

here $h_0 = x$

g ReLu function

Without a decoder, it is equal to a standard deep auto-encoder, as shown below. The formula resembling a latent representation is used to produce the output vector:

$$y = g(W_(n+1).h_n + b_(n+1) \qquad (2)$$

Having minimised the square reconstruction error over m training samples $(x_(i), y_(i))^m, i = 1$, one can obtain the estimator of the model $\theta = (W_i, b_i)$.

$$E(\theta) = \sum_{i=1}^{m} (x_(i), y_(i))^2 \qquad (3)$$

A. *Stacked Non-Symmetric Deep Auto-Encoder*

The proposed model utilizes the Stacked Non-Symmetric Deep Auto-Encoder (NDAE) [15] as a deep learning method to extract features from the input data. The NDAEs are stacked to create a deep learning hierarchy, which enables the model to learn complex relationships between features through layer-wise representation learning. By stacking NDAEs, the model can extract the most descriptive features, which can be emphasized to improve the model's performance.

However, the classification ability of the model with a typical softmax layer is found to be less than that of other discriminative models such as K nearest neighbor, random forests, and support vector machines. To address this issue, the deep learning power of the NDAE layer is combined with a shallow learning classifier.

The proposed model consists of two NDAEs arranged in a stack, combined with the classifiers. Each NDAE comprises three hidden layers, each with 14 neurons, 28 neurons, and 28 neurons, respectively. The model is trained using a similar training technique as traditional deep autoencoders, which involves minimizing the reconstruction error between the input data and its encoded representation.

Overall, the proposed model combines the deep learning power of the NDAE layer with the classification power of the shallow learning classifier to make intrusion detection systems in IoT networks more accurate and work better.

Below are the classifiers being used in our research:

1) Random Forest: The Random Forest algorithm [16] is a widely used and powerful machine learning technique that can handle complex data and improve the model's generalization capability. Its ability to handle missing data, estimate feature importance, and generate accurate predictions has made it a

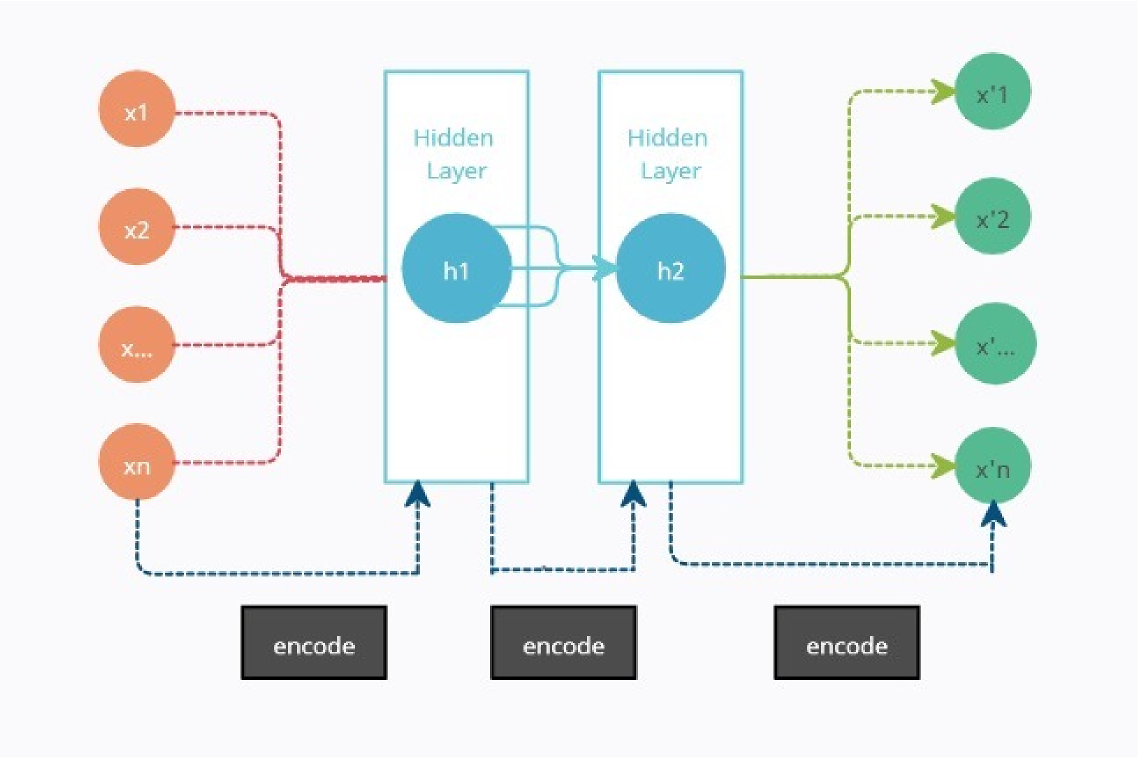


Fig. 3. Non-symmetric Autoencoder

preferred choice among data scientists and machine learning practitioners for a variety of applications. The provided report is based on voting techniques like popular voting or averaging.

2) KNN(K Nearest Neighbors): K-Nearest Neighbors (KNN) is a simple yet effective machine learning algorithm used for classification and regression tasks. It operates by calculating the distance between the input data point and its K-nearest neighbors to determine its classification or regression value. KNN is a non-parametric algorithm, meaning it does not make any assumptions about the data distribution. This makes it useful for handling complex datasets with non-linear relationships.

KNN measures the distance between both the query and each data instance, selects a specified number of examples (K) where the query is closest, then votes for its most For classification, choose a frequent label, and for regression, provide an average of the labels.

3) SVM(Support Vector Machine): In order to classify information in n-dimensional space, the SVM method generates decision boundaries that may be used as cutoffs. Each feasible decision has an ideal boundary that may be represented as a hyperplane. This machine learning system generates hyperplanes by randomly selecting long and short vectors. Support vectors are what we call them when things get really bad, and the method at issue here is called the SVM classifier. The SVM methods depend on a kernel, which is a special mathematical operation and technique. We call this central part of a computer ”the kernel.”In order to enhance its output, the kernel's principal job is to gather data and transform it into the proper format. This is the kernel's primary role. We used a tool called a polynomial kernel, which is a feature map constructed using a polynomial of the available data, to aid in our investigation.

VI. Results and Experimental Setup

The research report is an important component of the overall project since it provides an in-depth discussion of the experimental design as well as the evaluation metrics that were used in order to verify the hypothesis. In this section, the most significant results and inferences drawn from the research are provided, along with a statistical analysis of the findings and a discussion of the importance of those findings. In addition to this, it provides a straightforward and condensed description of the experimental setting, which includes not only the apparatus but also the methods that were carried out throughout the experiment.

1) Method for evaluation of Models: The steps taken to evaluate the efficacy of the machine learning models Random Forest (RF), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) using the Stacked Non-Symmetric Auto Encoder Algorithm. The ”score” function from Scikit-learn is used to evaluate these models.

Utilizing the X and Y train parameters during the training process and Scikit-learn's score method to evaluate each model after training. The score function calculates the accuracy score of the models by default, which is the ratio

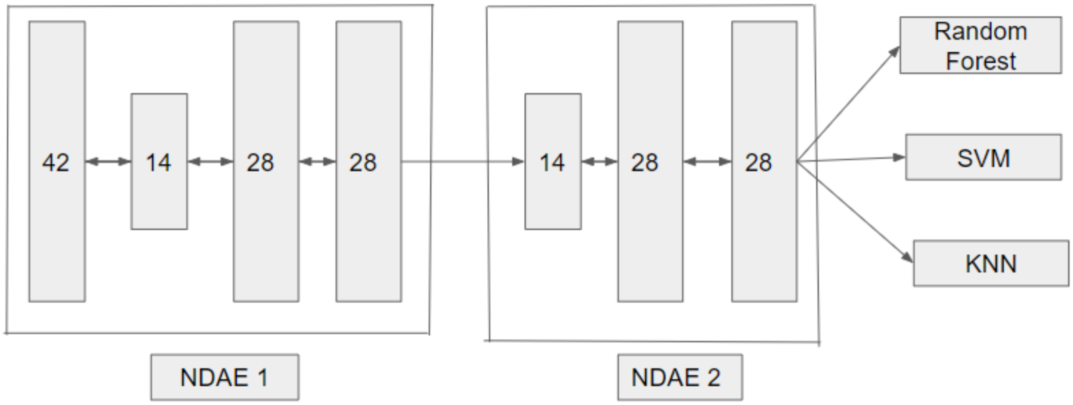


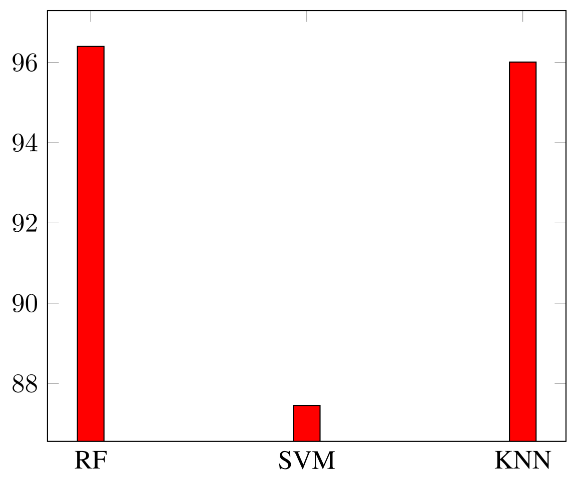
Fig. 4. Stacked Non-symmetric Autoencoder

of correct predictions to all predictions. The term ”scores” is used to refer to the accuracy results obtained from the classifiers.

The score method does not require any prior information about the actual predictions since it makes predictions automatically when called with the X test as input. The score function is a common metric for evaluating the performance of machine learning models, and it can be thought of as a shorthand for accuracy. Overall, the text provides insight into how machine learning models are evaluated and scored using commonly used algorithms and tools.

2) Evaluation of the results:

Classification Model	Accuracy(%)	Time Taken
Random Forest	96.3994	10.3115
SVM	87.4513	340.3042
KNN	96.0099	29.2771


Accuracy(%) Results of the Models

As many researches before us for classification models, we have used Keras, Tensorflow, and Sklearn for evaluating proposed models. To cross-reference the proposed model we have used the NetFlow ToN-IoT dataset to be evaluated with RF, SVM/SVC, and KNN as classifiers within the proposed model structure with two stacked non-symmetric autoencoders.

RF was the most successful. As RF is less susceptible to overfitting due to its ability to handle categorical features. Among the other models used in our methodology, Random Forest gives the best accuracy rate of 96.3994 percent with minimal runtime.

As we can see from the above table, the remaining models are significantly behind and will take much longer to reach fit. The SVM model with a ”poly” kernel has been the least effective in terms of outcome, requiring the most time for fitting. Whereas with the KNN, it has performed substantially better compared to the SVM but with minor improvements compared to the RF, with almost the same accuracy but requiring longer to fit the model for the KNN. From all this, we can conclude that with the stacked non-symmetric autoencoder in tandem with the RF classifier, we were able to achieve the most accurate and reliable results with minimal time.

Attack class	Precision	Recall	F1-Score
Benign	0.99	1.00	0.99
Backdoor	1.00	0.92	0.96
DDoS	0.97	0.97	0.97
DoS	0.89	0.88	0.88
Injection	0.85	0.76	0.80
MITM	0.72	0.82	0.62
Paasword	0.92	0.90	0.91
Scanning	1.00	0.99	1.00
XSS	0.91	0.95	0.93
Accuracy			**0.96**

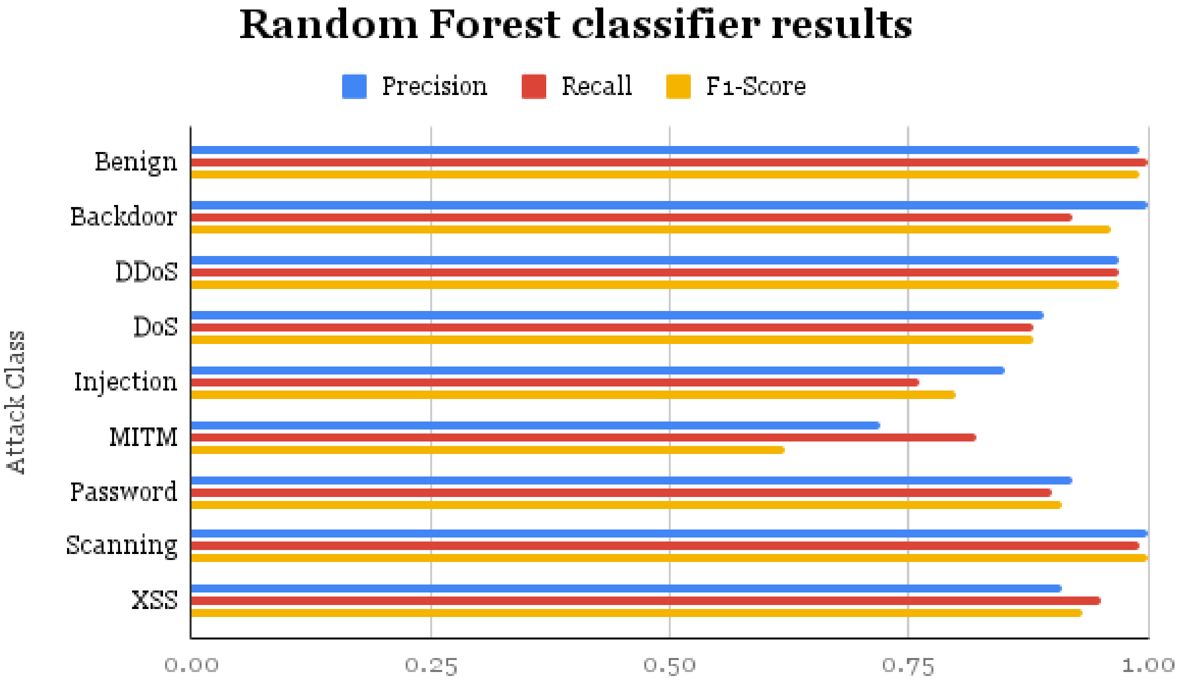


Fig. 5. Random Forest Classification Results

Attack class	Precision	Recall	F1-Score
Benign	0.97	0.96	0.97
Backdoor	1.00	0.92	0.96
DDos	0.84	0.92	0.88
Dos	0.77	0.14	0.24
Injection	0.89	0.24	0.38
MITM	0.56	0.39	0.63
Password	0.67	0.78	0.72
Scanning	0.98	0.96	0.97
XSS	0.68	0.92	0.78
Accuracy			**0.87**

Attack class	Precision	Recall	F1-Score
Benign	0.99	0.99	0.99
Backdoor	0.92	0.92	0.92
DDoS	0.96	0.96	0.96
DoS	0.88	0.89	0.88
Injection	0.86	0.74	0.80
MITM	0.72	0.82	0.62
Paasword	0.90	0.89	0.90
Scanning	0.99	0.99	0.99
XSS	0.91	0.96	0.93
Accuracy			**0.96**

VII. Conclusion

Designing good Intrusion detection systems are important. To address this problem, we combine the power of stacking NDAE and classification models. Using this type of auto-encoders, we are able to facilitate and improve classification as this provides non-symmetric data dimensionality reduction. By combining both stacked NDAE and classification algorithms. This allowed us to exploit their own strengths and reduce analytical overloads. Using Random Forest as the classification algorithm, we are able to achieve the best accuracy in less time as compared to SVM and KNN. Overall, we conclude that RF and KNN give us the best accuracy with RF doing it in the least amount of time.

References

[1] Zhaojun Gu, Liyin Wang, Chunbo Liu, Zhi Wang, ”Network Intrusion Detection with Non Symmetric Deep Autoencoder Feature Extraction”, Security and Communication Networks, vol. 2021, Article ID 2843856, 11 pages, 2021.

[2] G. Muhammad, M. S. Hossain and S. Garg, ”Stacked Autoencoder-based Intrusion Detection System to Combat Financial Fraudulent,” in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2020.3041184.

[3] H.-W. Lee, N.-R. Kim, and J.-H. Lee, “Deep neural network self-training based on unsupervised learning and dropout,” Int. J. Fuzzy Logic Intell. Syst., vol. 17, no. 1, pp. 1–9, Mar. 2017. [Online].

[4] B. Yan and G. Han, “Effective feature extraction via stacked sparse autoencoder to improve

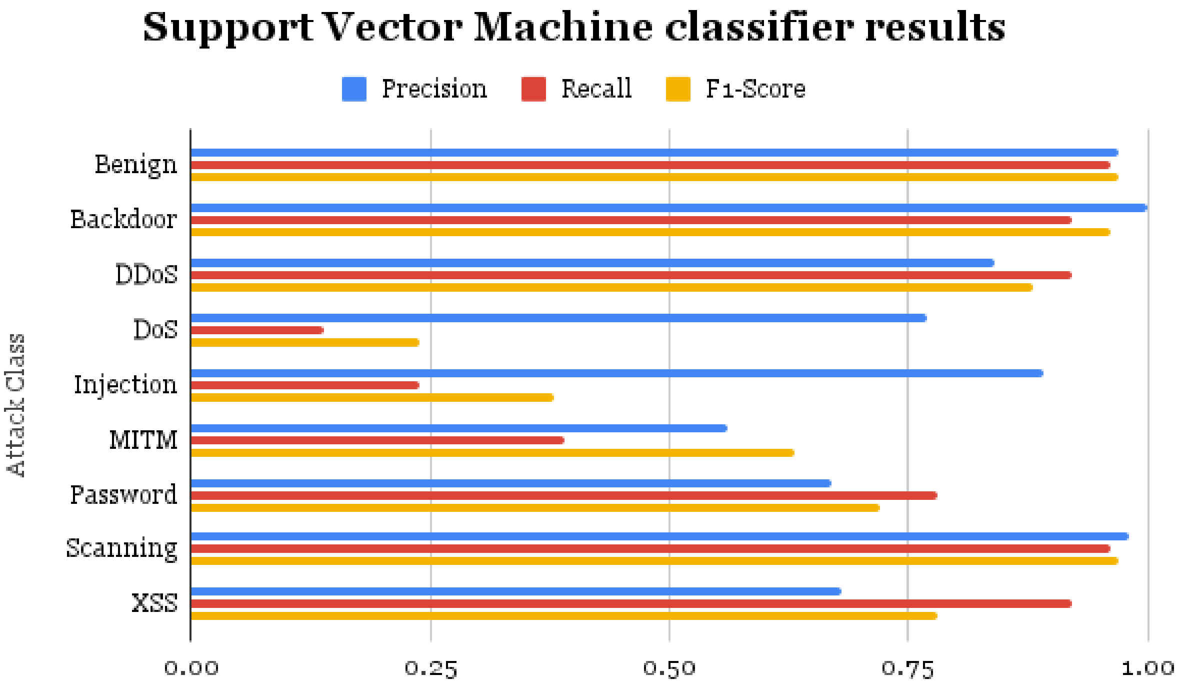


Fig. 6. Suppoprt vector Machine Classification Results

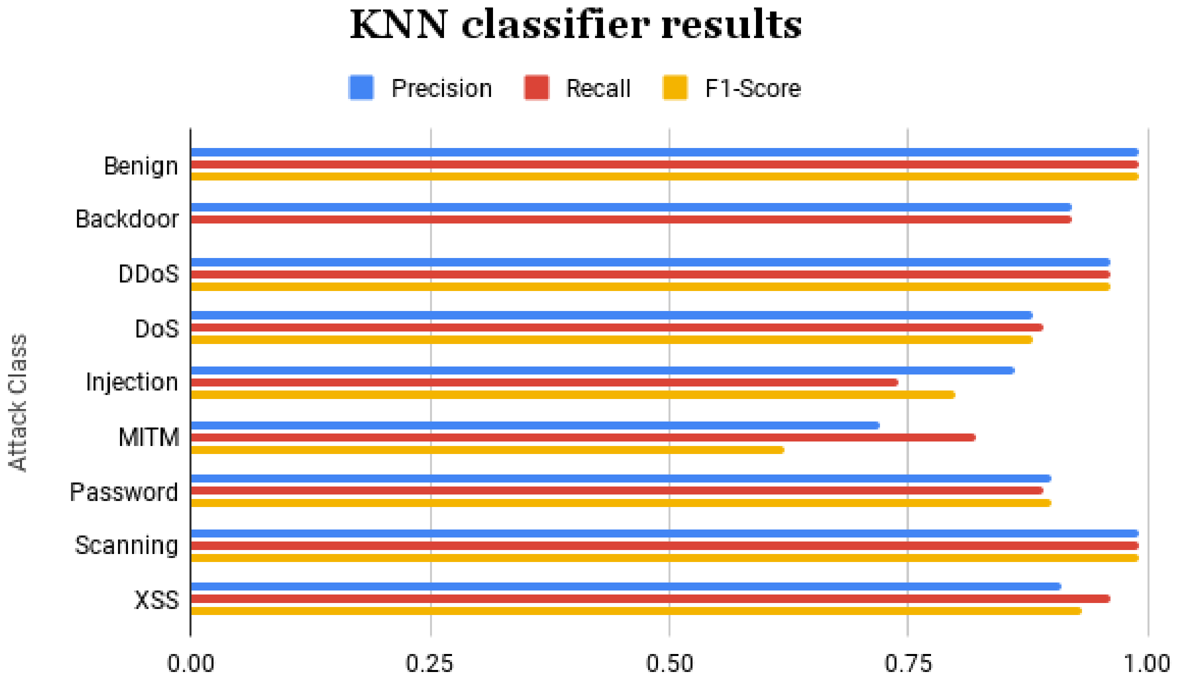


Fig. 7. K Nearest Neighbours Classification Results

intrusion detection system," IEEE Access, vol. 6, pp. 41238–41248, 2018.

[5] M. J. Zhang, M. Zulkernine, and A. Haque, "Random-forests-based network intrusion detection systems," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., vol. 38, no. 5, pp. 649-659, Sep. 2008.

[6] Shapoorifard, Hossein and Shamsinjead Babaki, Pirooz. (2017). "Intrusion Detection using a Novel Hybrid Method Incorporating an Improved KNN". International Journal of Computer Applications. 173. 5-9. 10.5120/ijca2017914340.

[7] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021.

[8] F. Kamalov, R. Zgheib, H. H. Leung, A. Al-Gindy and S. Moussa, "Autoencoder-based Intrusion Detection System," 2021 International Conference on Engineering and Emerging Technologies (ICEET), 2021, pp. 1-5, doi: 10.1109/ICEET53442.2021.9659562.

[9] K. Saurabh et al., "LBDMIDS: LSTM Based Deep Learning Model for Intrusion Detection Systems for IoT Networks," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 753-759, doi: 10.1109/AIIoT54504.2022.9817245.

[10] K. Saurabh, T. Kumar, U. Singh, O. P. Vyas and R. Khondoker, "NFDLM: A Lightweight Network Flow based Deep Learning Model for DDoS Attack Detection in IoT Domains," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 736-742, doi: 10.1109/AIIoT54504.2022.9817297.

[11] Moustafa, N. ToNIoT and unsw15 Datasets. Available online: https://research.unsw.edu.au/projects/toniot-datasets (accessed on 3 April 2022).

[12] Gad, A.R.; Nashat, A.A.; Barkat, T.M. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. IEEE Access 2021, 9, 142206–142217.

[13] Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions; Springer International Publishing: Cham, Switzerland, 2021; Volume 8.

[14] W. Zheng, "Intrusion Detection Based on Convolutional Neural Network," 2020 International Conference on Computer Engineering and Application (ICCEA), 2020, pp. 273-277, doi: 10.1109/ICCEA50009.2020.00066.

[15] A. Majumdar and A. Tripathi, "Asymmetric stacked autoencoder," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 911-918, doi: 10.1109/IJCNN.2017.7965949.

[16] Livingston F. Implementation of Breiman's random forest machine learning algorithm. ECE591Q Machine Learning Journal Paper. 2005:1-3.

[17] "IoT multiplies risk of attack," Network Security, vol. 2015, no. 5, p. 20, May 2015, doi: 10.1016/s1353-4858(15)30041-6.