

Web-based Intrusion Detection System Using Neural Network

Oluwabunmi Titilope Oladele¹

Ganiyyat Bolanle Balogun²

Peter Ogirima Sadiku³

Michael Abayomi Olatunji⁴

&

Tinuke Omolewa Oladele⁵

¹Department of Information Technology, University of Ilorin, Ilorin, Kwara State, Nigeria.

^{2,3,4,5}Department of Computer Science, University of Ilorin, Ilorin, Kwara State, Nigeria.

⁴Department of Information Systems and Technology, Kings University, Ode-Omu, Osun State, Nigeria.

oladele.ot@unilorin.edu.ng¹, balogun.gb@unilorin.edu.ng², sadiku.po@unilorin.edu.ng³,
mikeolaus@gmail.com⁴ and oladele.to@unilorin.edu.ng⁵

Abstract

As numerous businesses go online, there has been a rise in the use of web-based applications which, therefore, require taking further security. As a result of the large number of users on the internet, people are vulnerable to malicious intruders trying to negotiate the integrity, confidentiality, and availability of resources. The online pattern of a user has to be analyzed to determine if any inconsistency occurs in comparison with the usual pattern. Hence, an Intrusion Detection System is proposed to detect the attacks. The intrusion detection system makes use of an Artificial Neural Network (ANN) which adopts a neural network algorithm. Streamlit was used in the development of the system. For the neural network, rules like; brute force attack, SQL injection, and XSS injection were used to train the model.

Keywords: Intrusion Detection, Artificial Neural Network,

Introduction

With the rapid growth of the information age, technology has become an important factor in our daily lives. There has been a significant rise in the use of web-based applications and they are getting more important as numerous applications are designed to deal with a specific area of interest. Web-based applications such as online blogs, social networking sites, e-banking and e-commerce have become a regular platform for transmitting information and delivering online services. Even though web-based application provides great digital experiences, it does however have some side effects and only the secured ones deliver the services safely (Agarwal & Hussain, 2018). According to Keturahlee (2020), web-based application security is one of the important issues in the era of information security due to the growth of web attacks. To detect this, an Intrusion Detection System is proposed. Intrusion Detection Systems (IDSs) can assist organizations prevent cyber-attacks by detecting malware. Different types of intrusions include

attempted break-in/masquerade attacks, Denial of Service attacks, Cross-Scripting attacks and

soon. Another example is penetrating a security control system looking for security weaknesses and potentially gaining access to the system's features and data.

Intrusions are malicious attacks that spread swiftly over networks, and detecting them is crucial owing to the potential damage they can inflict. Security in networks is currently a worldwide major issue in computer security and defence. Breaches, threats, or intrusions in network infrastructures typically result in tremendous sensitive leaks of data, lowering an organization's efficiency and productivity quality (Zouhair et al., 2018). As reported by Devikrishna and Ramakrishna (2013), there are two intrusion detection-based methods; anomaly-based intrusion detection which can also be known as behavior-based detection. It detects malware by searching for any abnormal network traffic in the system. It detects normal activities of the system and renders any other activity as abnormal. Another method is the knowledge-based detection. It searches for activities that are similar to registered signatures of intrusions. It detects any unusual activities and renders any other activity in the system as normal. A Neural Network Intrusion Detection system can be trained to learn patterns in a system to detect intrusions by recognizing patterns of intrusions. It has been established (Om & Sarkar, 2010), that there are three steps involved in making a neural network; pre-processing the data, training the network and testing the data.

Agarwal and Hussain (2018) claimed that identifying or protecting against cyber threats is a never-ending battle as web applications are increasing rapidly and many business owners are starting to deploy web applications and replace traditional applications. Data stored in the database has increased over time as a result of technology growth. Intrusion detection methodology is still immature in the web application security domain. As a result, Intrusion Detection Systems (IDS) are regarded as critical tools for continuously monitoring malicious activities and detecting threats that could jeopardize the network's integrity, privacy, or availability (Kajal & Nandal, 2020). This paper therefore focuses on developing an Intrusion Detection System to detect web application intrusion using a neural network. The system was developed using a neural network algorithm to detect malware on the dataset. This paper has combined intrusion detection systems and neural networks to detect malware.

Literature Review

Applications that are web-based are now a favourite way to give access to information that is produced and created dynamically. Security flaws that are web-related portray a large number of vulnerabilities and this is based on the analysis of vulnerabilities that have become common in the years past. However, it is not always possible to prevent attacks as such because even when acceptable devices and machines are provided, developers with little time often disable the security devices just to get the job done. In other words, Intrusion Detection Systems (IDSs) come into play as mechanisms of prevention to detect these web-based attacks (Emmanuel, 2015). A limited amount of research has been carried out on using neural networks in detecting network intrusions. Artificial Neural Network is the most suitable approach for detecting network intrusions. It can solve major problems that other current approaches cannot deal with. Neural networks were mainly proposed to recognize the system users' characteristics and determine statistically considerable differences from the users' behaviour (Roheen & Baqar, 2023).

The proposed system consists of three composite areas of computer science; Web Application Development, Computer Security, and a branch of Artificial intelligence. For this system to be put together accurately all the subsidiaries must be extensively researched. They will be taken individually so they can be understood as a unit before they can be understood as an entity.

Intrusion Detection Systems

Intrusion detection is used to catch intrusions or attacks into computer and network systems when they want to violate the system's security and privacy (Roheen & Baqar, 2023). Several defence mechanisms have been adopted by organizations to ensure adequate security for web applications. Web Application Firewall (WAF) is the most prominent defence mechanism used by organizations to protect the web application after deployment. Configuring WAF is like designing an alarm system for a building without considering the blueprint of the building. IDS, primarily designed to protect web applications, is a potential solution. IDS is the defence tool which is used to detect and report intrusions to the administrator (Agarwal & Hussain, 2018). Generally, Attacks that can be detected by IDS are categorized into Denial-of-Service attacks, unusual usage patterns, Impersonation and man-in-the-middle attacks and Worms (Chen, Blasco, Alzubi, & Alzubi, 2014; Alex, David, & Aladdin, 2018).

The process of intrusion detection and prevention is broken down into four major phases as follows:

- a. Pre-processing:** This is the first phase of the Intrusion detection system. The collection of activity from an IDS sensor is done in this phase. After the collection of activity, data is organized in a particular pattern for classification. It also helps to determine the format in which to put the data, which is usually a canonical format or a structured database.
- b. Analysis:** After the pre-processing phase is completed, the analysis stage begins. The data record is then compared with the knowledge base which will either be logged as an intrusion event or it will be dropped and the next data record is analyzed.
- c. Response:** In the intrusion detection system, information is received passively after the fact, resulting in an alert. The response can be automatically performed or can be done by manually analyzing the situation.
- d. Refinement:** This is the stage where fine-tuning is done, based on the previous usage and detected intrusions. This helps to reduce false positive levels and to have more security tools. These are tools like CTR (Cisco Threat Response) that help with the refining stage by actually making sure that an alert is valid by checking whether one is vulnerable to the attack or not, rule-based detection, even known as signature detection, pattern matching and misuse detection (Alex et. al., 2018).

Neural Networks as an Aspect of Artificial Intelligence (AI)

Neural Networks are new systems and computational methods for machine learning, knowledge demonstration, and finally the application of knowledge gained to maximize the output responses of complex systems (Chen et al., 2014). Neural networks were discovered in the early 1940's by Warren Mcculloch and Walter Pitts with the intention of calculating nearly any Arithmetic and logical functions. Then after this, some improvements were made. In 1947, they were able to make it recognize special patterns. It was finally at Dartmouth that some Professors and Scientists

conducted research and discussed the idea of how to simulate a brain since that is what is connected to the nervous system (Kriesel, 2005). Neural Networks is an aspect of AI because it deals with the simulation of human thinking and behavior into computer systems but the aspect is limited to the brain and the nerve cells.

Paradigms of Learning in Neural Networks

- i. **Supervised learning or Associative learning:** In supervised or associative learning, the network is trained by providing the input and matching output patterns which can be provided by an external factor or by the system which contains the network (Self-Supervised).
- ii. **Unsupervised learning or Self-organization:** In this paradigm, an output unit is trained to respond to clusters of patterns within the input. The system is supposed to discover statistically salient features of the input population. Unlike the case of the supervised learning paradigm, where there is a prior set of categories to which the patterns are to be classified rather the system must be able to develop its representation of the input stimuli.
- iii. **Re-enforcement learning:** Reinforcement learning is a machine learning technique that sets parameters of an artificial neural network, where data is usually not given, but generated by interactions with the environment. Reinforcement learning is frequently used as a part of an artificial neural network's overall learning algorithm (Singh, Srivastava, & Kumar, 2024).

Algorithms used to implement Neural Networks

1. **Pattern Matching:** Pattern matching algorithms are the algorithms that are used to figure out whether a specific string pattern occurs in a string text. This algorithm was adopted in this study. Two of the most widely used pattern-matching algorithms are the Naïve Algorithm for pattern matching and the pattern matching algorithm using finite automata. It involves taking two sets of data and comparing them to one another and if the outcome is positive then the test has been passed but if it is not then the test has failed (JingLu, 2015).
2. **Back-propagation:** is a supervised learning method used for feeding forward networks that are implemented by learning in batches. This is the most common algorithm used for neural network training. The main reason why it is so commonly adopted for neural networks is that it can train both arbitrary linear and non-linear functions as well as regression/classification of problems (Brownlee, 2015).
3. **Learning Vector Quantization:** It is also a supervised learning algorithm that is strongly related to the Self-Organizing Maps which are unsupervised learning algorithms. It processes information by preparing prototype vectors in the domain of the observed input data samples and using these vectors to classify unseen examples.
4. **Hopfield Network:** This is a neural network algorithm that processes information by associating the components of the inputs pattern with a pattern called Content Addressable Memory (CAM). It is recursive and can solve problems by matching cues for an input pattern to an associated pre-learned pattern (Brownlee, 2015).

5. **Perceptron:** The Perceptron algorithm is a single-layer feed-forward neural network that is inspired by the information processing of a single neuron. The perceptron is made up of a data structure and separate procedures for training and applying the structure which is a vector of weights. It is very good for approximating arbitrary linear functions and classification or regression problems but it cannot learn a non-linear mapping between input and output values (Brownlee, 2015).

Related work

Intrusion detection using fuzzy clustering and neural network

Surana (2012), proposed a hybrid neural network based on fuzzy clustering and neural networks for an intrusion detection system and this was done to eliminate the limitations involved in using only neural networks which are the requirements of high computational resources. The implementation of this system starts with the breaking down of the dataset into smaller subsets using the fuzzy clustering technique and consecutively each ANN on each of the subsets and then aggregating the output from each of the subsets to form the final output. Lee et. al. (2001), posited data mining approaches for detecting intrusions. One of the major approaches mentioned is association rules. The association rule describes user behaviour by learning the record patterns or signatures.

Neural Network IDS implementation according to attacks

Ramakrishna (2013), classified attacks and deduced how well the neural network intrusion detection system will be used to detect them. He presented the multilayer perceptron approach for this research and also classified the types of intrusion detection systems; Host-based, Network-based, misuse-based, anomaly-based and finally neural network-based. The neural network intrusion detection system in their study was analyzed as a pattern recognition technique because it is an information processing system that imitates the brain and the biological neural network. The system architecture that was implemented had four compartments; the packet monitor: which as its name implies monitors and captures packets that will serve as data to the intrusion detection system, the pre-processor: which collects and processes network traffic, feature extractor- which extracts feature vectors from the network traffic that has already been pre-processed and the Classifier- which recognizes and classifies the patterns of attacks.

Neural Network IDS implementations with other algorithms

Harbi and Rahman (2010), developed a new learning algorithm for adaptive network intrusion detection using naive Bayes and decision trees. It has several advantages over other approaches like reduced false positive rates, reduced noise in data training, elimination of redundant attributes and so on. It can be used in situations of limited computational resources like space, CPU time, IO resources and so on. The algorithm's performance was tested with other existing learning algorithms by using the KDD99 benchmark intrusion detection dataset and this proved its high detection rate and low false positive rate. Shah (2014), developed a Real Time Intrusion Detection System using Artificial Neural Networks (ANN). Real-time has to do with frequent updates that the system will get and, in this case, it is a real-time intrusion system that receives updates on any new intrusions coming into the system. As it is known there are several algorithms for different neural network approaches. Jianfa (2015) authored an article called Network Intrusion Detection

Based on a General Regression Neural Network Optimized by an Improved Artificial Immune Algorithm. It illustrates a general regression neural network (GRNN) based on the artificial immune algorithm with elitist strategies (AIAE).

Anomaly Intrusion Detection Using Neural Network Reviews

Another recommended literature is ‘Anomaly-based network intrusion detection with unsupervised outlier detection’ (Zhang, & Zulkerine, 2006). It entails the supervised training method that uses an approach called the outlier’s approach. Although this kind of training data is difficult to obtain in real-world environments it has a high false positive rate. It is very good at detecting outliers especially when using it in a scenario of random forests. According to Schwab (2015), the IDS journey started thirty years ago when increasing enterprise network access spawned a new challenge; the need for user access and user monitoring. As day-to-day operations grew increasingly is dependent upon the shared use of information systems, levels of access to these systems and clear visibility into user activity were required to operate safely and securely. In the 1990s, IDS technology improved to address the increasing number and sophistication of network attacks. This new method, named anomaly detection, relied on identifying unusual behavioral patterns on the network and provided alerts for any identified abnormality. Muthukumar (2015) developed an anomaly detection system using an approach called the Outlier Detection Approach international proceeding. Here, it is said that the anomaly dataset is measured by the neighbourhood outlier factor (NOF) and the trained models consist of storage environments for big datasets.

Implementation of Neural Networks with Other AI Fields On Intrusion Detection

Neural networks can also be used alongside machine learning according to Vinchurkar and Reshamwala (2012). Machine learning was said to be based on statistical analysis of data and may be able to use patterns found in the previous data to make decisions about the new ones and this helps in the shortcomings of neural networks which are reduced performance due to increasing data and network traffic. To make use of machine learning, the system would have to undergo vector regression. Algorithms that support vector regression are; the Naive Bayes classifier and genetic algorithms. Surana (2015), disclosed that a hybrid Artificial Neural network based on fuzzy clustering and neural network can be made to detect intrusions because neural networks though very effective in capturing non-linearity in data recognition, require the use of a lot of resources. That is where fuzzy clustering comes in and it can be defined as a technique that divides data into smaller groups or portions and this can help the proposed system to reduce the time spent in detecting and achieving a higher detection rate. Firstly, the Fuzzy c-means (FCM) clustering algorithm was used to divide the data into clusters which enables the backpropagation algorithm to train the ANN with the 1999 KDD Cup dataset. Then the ANN is aggregated to detect the five classes of attacks at the output layer; Normal, Dos, U2R, R2L and Probe.

Framework of the Proposed System

Figure 1 shows the framework of the system. It depicts the process flow from Data collection, to Data Preprocessing, Data analysis, Feature Engineering, and Data Slitting. Model training, Model Evaluation, Model Deployment, down to Model Prediction on data.

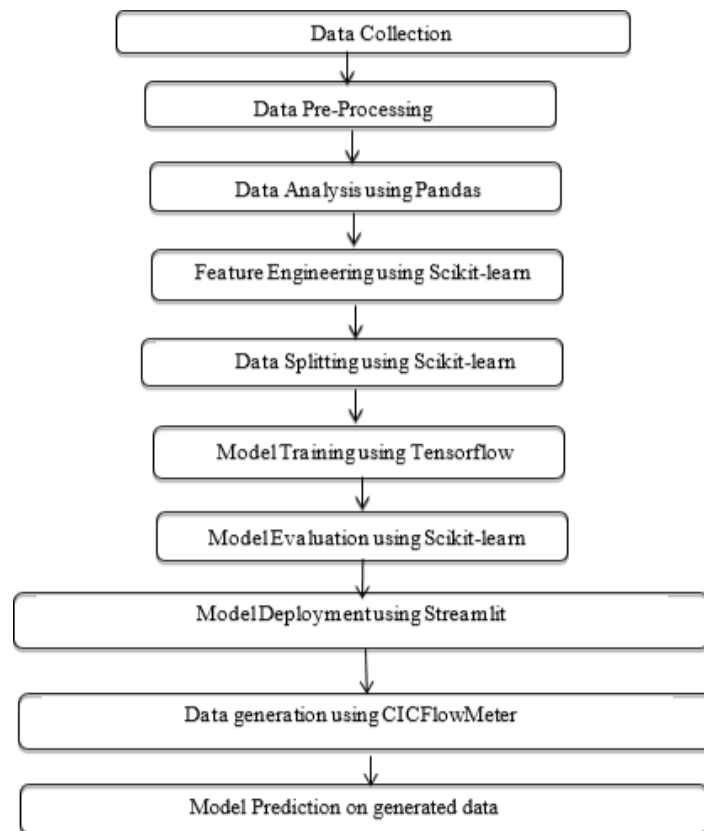


Figure 1: Framework of the Proposed System

Data Collection and Preprocessing

The creation of a reliable web application intrusion detection system depends heavily on data collection and preprocessing. To train and evaluate the system effectively, high-quality and representative data are required. The dataset for this research was obtained from the Canadian Institute for Cybersecurity (CIC) IDS-2017. The Intrusion Detection Evaluation Dataset (CICIDS2017) is a network traffic dataset comprised of both normal traffic and simulated abnormal data caused by intentional attacks on a test network. It comprises a variety of techniques, including SQL injection, cross-site scripting (XSS), and other widespread vulnerabilities in web applications. To help the intrusion detection system, learn from various attack situations and generalize those lessons, the dataset is carefully curated to encompass a wide range of attack patterns and network behaviours. After the dataset was gathered, Filtering, normalization, and feature extraction are a few of the crucial methods used during the preprocessing stage. Filtering

is done to get rid of useless or noisy data that can make the intrusion detection system work less

effectively. The data is next subjected to normalization to ensure that it is uniform and that each feature has a similar magnitude and distribution. The neural network can learn more skillfully by normalizing the data since it reduces biases brought on by changing feature magnitudes. The convergence and stability of the training processes are both enhanced by normalization.

A critical step in getting the data ready for analysis is feature extraction. It entails taking the raw network traffic data and extracting interesting and useful information. Techniques like statistical analysis, frequency analysis, and time-series analysis may be used during this procedure to identify pertinent patterns and traits in the network traffic. The neural network learns from the extracted features and uses them as input variables to produce precise predictions based on the supplied data. Preserving the dataset's integrity and representativeness is given considerable consideration during the preprocessing stage. To find any anomalies or inconsistencies that can impair the quality of the training and evaluation procedures, data validation approaches are used. To guarantee the consistency and dependability of the dataset, all problems are addressed and fixed. The model is trained and tested on a high-quality and diversified dataset. After the data collection and preprocessing stages, the dataset is turned into an appropriate format for further analysis by employing stringent filtering, normalization, and feature extraction procedures. With the help of this reliable preprocessing methodology, the model can learn from sample data and make accurate judgments when looking for suspected intrusions.

Data Preprocessing

Data preprocessing includes the method of cleaning, encoding, or changing data to make it interpretable and easily understood by the machine. This is achieved using the following three steps:

- i. Data Cleaning
- ii. Data Integration
- iii. Data Transformation

Neural Networks Design

There were several levels in the neural network architecture, including input, hidden, and output layers. The preprocessed data, which represented different attributes gleaned from network traffic, was delivered to the input layer. The majority of the computations and feature modifications were carried out by the hidden layers, which were positioned between the input and output layers. Since the number of neurons in these hidden layers varied, the network was able to learn and express complex relationships in the data.

The neural network can model complex decision boundaries. This is attributed to the carefully chosen activation functions that were used to introduce non-linearity. The sigmoid, ReLU (Rectified Linear Unit), and tanh (hyperbolic tangent) functions are the often-employed activation functions. The model's specific needs as well as the anticipated neuronal activity were taken into consideration while choosing the activation function. The neural network's learning process was greatly aided by connection weights, which control how strongly and influentially neurons are connected. During the training phase, these weights were iteratively changed using optimization strategies like backpropagation and gradient descent after being originally randomized. The link

weights were altered through intensive experimentation and optimization to reduce the discrepancy between the predictions of the network and the labels of the ground truth.

Combining domain knowledge and empirical experimentation, the design decisions for the neural network topology, activation functions, and connection weights were made. The performance of various architectures and hyperparameter combinations was assessed using the proper assessment measures. The goal was to strike a compromise between precision, computational effectiveness, and good generalization to new data. The effectiveness of the network was assessed using some parameters, including detection rate, false positive rate, precision, and recall. The neural network design was adjusted to achieve high accuracy and effective training, allowing the system to successfully recognize and classify probable intrusions in online applications. This was done through repeated refining and fine-tuning.

Training and Decision-Making

The neural network was trained using the preprocessed dataset, employing techniques such as backpropagation and gradient descent. During the training process, the network learned to recognize patterns and differentiate between normal and malicious network behaviour. The decision-making process involved setting appropriate threshold values and defining rules for classifying network traffic as normal or malicious based on the network's output. Careful evaluation and validation of the trained neural network ensured its robustness and reliability.

Performance Analysis

The performance of the system was thoroughly evaluated to determine its effectiveness. Metrics such as True Positive Rate, False Positive Rate, True Negative Rate, and False Negative Rate were analyzed.

- a. **True Positive Rate (TPR):** also referred to as recall or sensitivity. It is used to measure the percentage of actual positives that are correctly identified by the model.

$$\frac{\text{number of true positives}}{\text{number of true positives} + \text{number of false negatives}}$$

- b. **True Negative Rate (TNR):** also called specificity, which is the probability that an actual negative will test negative. It represents when the outcome is No and the actual value is No.

$$\frac{\text{number of true negatives}}{\text{number of true negatives} + \text{number of false negatives}}$$

- c. **False Positive Rate (FPR):** The model's prediction is based on how many actual negatives the model predicted incorrectly. FPR represents when the outcome is No and the actual value is Yes.

$$\frac{\text{number of false positives}}{\text{number of false positives} + \text{number of true negatives}}$$

- d. **False Negative Rate (FNR):** It shows the count of instances wrongly classified in the negative class. The outcome depicts an incorrectly classified class denoting a negative class. FNR represents when the outcome is Yes and the actual value is No.

$$\frac{\text{number of false negatives}}{\text{number of true positives} + \text{number of false negatives}}$$

System Analysis

Even though many systems have shown considerable improvements in terms of detection rates and accuracy, problems still exist with false positives, scaling, and adaptability to changing attack vectors. The use of conventional rule-based or signature-based detection techniques are typical drawbacks found in existing systems.

Analysis of the Proposed System

The proposed web application intrusion detection system, which makes use of neural network algorithms to increase its efficacy, is thoroughly examined.

1. **Pattern Matching Algorithm:** The proposed system employs pattern matching algorithms, specifically designed to analyze network traffic and detect patterns associated with malicious activities. Pattern matching offers several benefits, including the ability to identify both known attack signatures and novel attack patterns.
2. **Accurate Detection:** One of the key advantages of the proposed system is its ability to provide accurate intrusion detection. By utilizing advanced pattern-matching algorithms, the system can effectively identify anomalous network behaviour indicative of potential intrusions. The use of machine learning techniques, such as neural networks, further enhances the system's accuracy by enabling it to learn from historical data and adapt to evolving attack patterns.
3. **Reduced False Positives:** The proposed system aims to minimize false positives by employing sophisticated pattern-matching algorithms and machine learning techniques. By accurately distinguishing between normal and malicious network behaviour, the system reduces the number of false alarms, thereby improving the efficiency of incident response and reducing the workload on security analysts.
4. **Real-Time Detection and Response:** The efficient implementation of pattern-matching algorithms, coupled with optimized data processing techniques, enables the system to analyze network traffic in real time, facilitating proactive incident management and minimizing the impact of security breaches.
5. **Integration and Scalability:** The proposed system is designed to integrate seamlessly into existing web application infrastructures. By adopting a modular design approach, the system can be easily integrated with various components of the web application stack, ensuring compatibility and minimal disruption to existing operations. Additionally, the system is designed with scalability in mind, allowing it to handle increasing network traffic volumes without sacrificing accuracy or real-time analysis capabilities.
6. **Ease of Configuration and Maintenance:** To facilitate ease of use and adoption, the proposed system prioritizes simplicity in configuration and maintenance processes. The neural network algorithms are designed to automatically adapt to changing network conditions and update detection rules as needed.

Software Requirements

The software requirements for an intrusion detection system with an artificial neural network are:

-
1. Operating System (OS): Windows 7 or higher version

2. Programming language: Python
3. Machine learning library: TensorFlow

Hardware Requirements

The hardware requirements for an intrusion detection system with an artificial neural network are:

1. Processor: 2.4GHz processor speed.
2. Memory (RAM): 256 MB (2 GB recommended)
3. Disk space: 60GB of hard disk memory
4. Graphics Processing Unit (GPU): HD graphics 4000 and above

Neural Network Intrusion Detection System (nnids) Operation

In detecting intrusion, the developed NNIDS was designed to embed the approach of the neural network algorithms. Distinctive rules were embedded into the model to enhance its intrusion detection features. It also provides the total number of malware found in the dataset. The system uses a neural network which enables the system to learn and if there is any unknown attack it will be defined as unknown.

System Deployment

This section contains the step-by-step procedure for operating the system. Streamlit was chosen for this study. It helps us create web applications. The User will be prompted to upload the chosen dataset into the system. After uploading the dataset, the system begins to test the dataset for intrusion.

Dataset

The Dataset is shown in Figure 2. It is made up of 2313810 rows and 78 columns which cannot be captured on the computer screen at once. The first five rows, numbered 0 to 4, show the first five rows in the dataset, and the remaining five rows, show the last five rows in the dataset.

In [4]: df

Out [4]:

	Protocol	Flow Duration	Total Fwd Packets	Total Backward Packets	Fwd Packets Length Total	Bwd Packets Length Total	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	Fwd Seg Size Min	Active Mean	Active Std
0	6	4	2	0	12	0	6	6	6.000000	0.000000	20	0.0	0.0
1	6	1	2	0	12	0	6	6	6.000000	0.000000	20	0.0	0.0
2	6	3	2	0	12	0	6	6	6.000000	0.000000	20	0.0	0.0
3	6	1	2	0	12	0	6	6	6.000000	0.000000	20	0.0	0.0
4	6	609	7	4	484	414	233	0	69.142860	111.967896	20	0.0	0.0
2313805	6	49	3	0	76	0	45	0	25.333334	23.028967	32	0.0	0.0
2313806	6	1286687	41	42	2664	6954	456	0	64.975609	109.864571	32	0.0	0.0
2313807	6	181	3	1	18	6	6	6	6.000000	0.000000	20	0.0	0.0
2313808	6	217	2	1	31	6	31	0	15.500000	21.920311	32	0.0	0.0
2313809	6	1387547	41	46	2728	6634	456	0	66.536583	110.129944	32	0.0	0.0

2313810 rows x 78 columns

Figure 2: Dataset

The Home Page

The Home page of the system is shown in Figure 3. It is a Streamlit interface that allows the user to drag and drop the data file for analysis. The dataset with the file name, intrusion.xlsx is then uploaded on the home page.

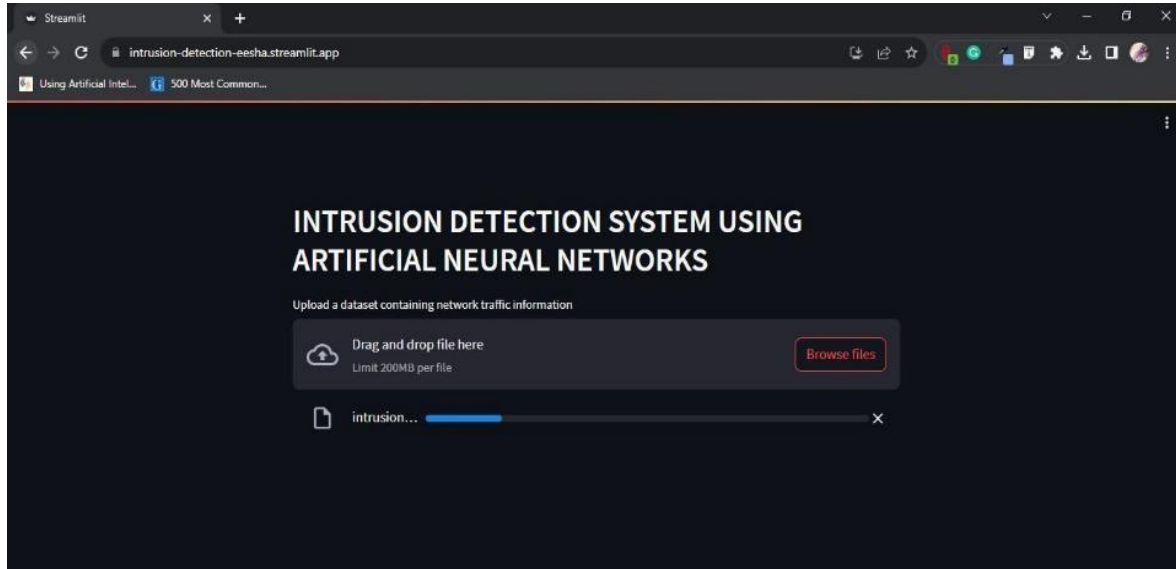


Figure 3: Uploading the dataset downloaded for testing

Model Training

The model was trained using 100 epochs (iterations) as shown in Figure 4. Only the training of the first eleven epochs is shown in Figure 4. The values for accuracy, precision and recall are displayed for each epoch of the model training.

```
with tf.device('/gpu:0'):
    model.fit(X_train,
              y_train,
              epochs=100,
              batch_size=2048,
              validation_batch_size=2048,
              callbacks=[callbacks],
              validation_split=0.2,
              verbose=1)
```

Epoch 1/100
211/211 [=====] - 7s 35ms/step - loss: 0.0611 - accuracy: 0.9729 - precision: 0.9878 - recall: 0.9576 - val_loss:
Epoch 2/100
211/211 [=====] - 9s 44ms/step - loss: 0.0607 - accuracy: 0.9732 - precision: 0.9879 - recall: 0.9579 - val_loss:
Epoch 3/100
211/211 [=====] - 9s 43ms/step - loss: 0.0599 - accuracy: 0.9736 - precision: 0.9883 - recall: 0.9585 - val_loss:
Epoch 4/100
211/211 [=====] - 9s 41ms/step - loss: 0.0598 - accuracy: 0.9735 - precision: 0.9879 - recall: 0.9586 - val_loss:
Epoch 5/100
211/211 [=====] - 9s 42ms/step - loss: 0.0595 - accuracy: 0.9737 - precision: 0.9882 - recall: 0.9587 - val_loss:
Epoch 6/100
211/211 [=====] - 8s 40ms/step - loss: 0.0591 - accuracy: 0.9736 - precision: 0.9877 - recall: 0.9591 - val_loss:
Epoch 7/100
211/211 [=====] - 8s 39ms/step - loss: 0.0582 - accuracy: 0.9741 - precision: 0.9884 - recall: 0.9594 - val_loss:
Epoch 8/100
211/211 [=====] - 8s 40ms/step - loss: 0.0579 - accuracy: 0.9741 - precision: 0.9881 - recall: 0.9598 - val_loss:
Epoch 9/100
211/211 [=====] - 8s 40ms/step - loss: 0.0582 - accuracy: 0.9738 - precision: 0.9880 - recall: 0.9592 - val_loss:
Epoch 10/100
211/211 [=====] - 8s 39ms/step - loss: 0.0576 - accuracy: 0.9741 - precision: 0.9880 - recall: 0.9599 - val_loss:
Epoch 11/100
211/211 [=====] - 8s 38ms/step - loss: 0.0577 - accuracy: 0.9740 - precision: 0.9884 - recall: 0.9591 - val_loss:

Predicting the Malware Present in the Dataset

Streamlit was used for deployment of the model as shown in Figure 5. The total number of malware present in the dataset is 6465 as shown in Figure 5.

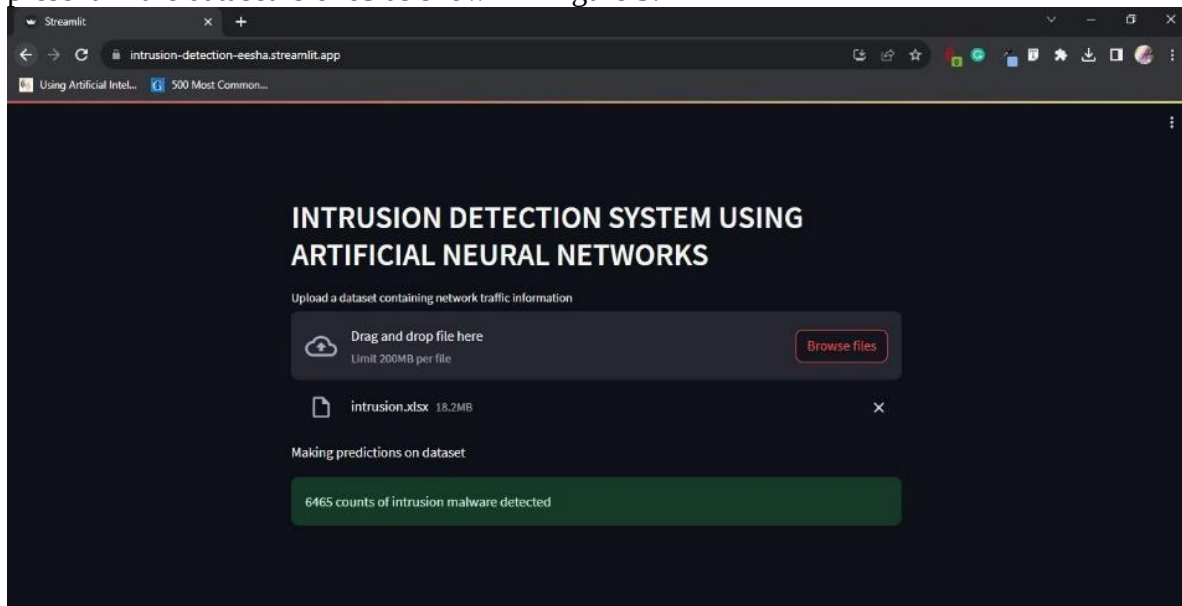


Figure 5: Predicting malware present in the dataset

This study is a detective rather than preventive and corrective method of managing a system's network.

Conclusion

In conclusion, the development and implementation of a web-based intrusion detection system using artificial neural networks represent a significant achievement in the cybersecurity field. This study has demonstrated the power and potential of artificial intelligence, specifically artificial neural networks, in detecting malware. The developed system is capable of analyzing vast amounts of network traffic in real time, identifying subtle patterns, and swiftly detecting malware.

Future Research Direction

For future research, more rules should be implemented to give the system a broader scope and improve the intrusion detection capability. Other algorithms can be used especially unsupervised learning algorithms like Self-Organizing Maps, Multilayer Perceptron and so on, to enable the system to learn better. The integration of neural network algorithms into web applications does not have to be limited to detection, prevention of the system from the web attack is also an important task in making a system free from threat.

References

- Agarwal, N., & Hussain, S. Z. (2018). A closer look on Intrusion Detection System for web applications. *ArXiv*, *abs/1803.06153*.
- Alex, S., David, D., & Aladdin, A. (2018). Intelligent Intrusion Detection Systems Artificial Neural Networks, *ICT Express*, Volume 4, Issue 2, (pp. 95-99).
- Brownlee, J. (2015). *Neural Algorithms*.
- Chen, T., Blasco, J., Alzubi, J., & Alzubi, O. (2014). Intrusion Detection. *IET Digital Library*. 10.1049/etr.2014.0007
- Devikrishna, K. S., & Ramakrishna, B. B. (2013). An Artificial Neural Network based Intrusion Detection System and Classification of Attacks. *International Journal of Engineering Research and Applications (IJERA)*, ISSN: 2248-9622, (pp. 1959-1964).
- Harbi, N., & Rahman, M. Z. (2010). Combining Naive Bayes and Decision Tree for Adaptive Intrusion Detection. *International Journal of Network Security & Its Applications*., 12-25.
- JingLu, S. X. (2015). A Neural Network-Based Interval Pattern Matcher. *Information*, 388-398.
- Keturahlee, C. (2020). An overview of Intrusion Detection and Prevention Systems. *Computer Security and Reliability*.
- Kriesel, D. (2005). *A Brief Introduction to Neural Networks*. Bonn.
- Lee, W., Stolfo, S., & E.Eskin, P. C. (2001). Real time data mining-based intrusion detection. *DARPA Information Survivability Conference & Exposition II, DISCEX '01 Proceedings* (pp. 89-100). Anaheim, CA: IEEE.
- Muthukumar, J. J. (2015). Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach. *In proceedings of International Conference on Intelligent Computing, Communication & Convergence*, (pp. 338-346). Odisha.
- Om, H., & Sarkar, T. (2010). Designing Intrusion Detection System for Web Documents Using Neural Network. *Communication and Network*, 54-61.

- Roheen, Q., & Baqar, Z., (2023). Artificial Neural Networks: An Overview. *Mesopotamian Journal of Computer Science*. 2023. 130-139. 10.58496/MJCSC/2023/015.
- Ramakrishna, K. S. (2013). An Artificial Neural Network based Intrusion Detection System and Classification of Attacks. *International Journal of Engineering Research and Applications (IJERA)*, 1959-1964.
- Singh, R., Srivastava, N., & Kumar, A. (2024). Anomaly detection framework for highly scattered and dynamic data on large-scale networks using AWS. *Int. j. inf. tecnol*.
- Schwab, P. (2015, September 9th). *The History of Intrusion Detection Systems*. Retrieved April 10, 2016, from Threat Stack: blog.threatstack.com/thehistory-of-intrusion-detection-systems-ids-part-1
- Shah, A. (2014). A real time intrusion detection system using artificial neural networks (ANN). *International Journal of Emerging Technology and Advanced Engineering* , 756-759.
- Surana, S. (2012). Intrusion Detection using Fuzzy Clustering and Artificial Neural Network. *Advances in Neural Networks, Fuzzy Systems and Artificial Intelligence*, 209-217.
- Surana, S. (2015). Intrusion Detection using Fuzzy Clustering and Artificial Neural Network. *Advances in Neural Networks, Fuzzy Systems and Artificial Intelligence*, 209-217.
- Vinchurkar, D. P., & Reshamwala, A. (2012). A Review of Intrusion Detection System Using Neural Network and Machine Learning Techniques. *International Journal of Engineering Science and Innovative Technology (IJESIT)*, 54-63.
- Zhang, J., & Zulkerine, M. (2006). Anomaly based network intrusion detection with unsupervised outlier detection. *IEEE International Conference on Communications* (pp. 2388-2393). Istanbul: IEEE.
- Zouhair, C., Abghour, N., Moussaid, K., El Omri, A., & Rida, M. (2018). A Review of Intrusion Detection Systems in Cloud Computing. In Y. Maleh, A. Ezzati, & M. Belaissaoui (Eds.), *Security and Privacy in Smart Sensor Networks* (pp. 253-283). IGI Global Scientific Publishing. <https://doi.org/10.4018/978-1-5225-5736-4.ch012>



Oluwabunmi Titilope OLADELE is a lecturer in the Department of Information Technology, University of Ilorin, Ilorin Nigeria. She holds a Masters of Technology degree in Computer Science from the Federal University of Technology, Akure. She is a member of the Nigeria Computer Society (NCS), the Organization for Women in Science for the Developing World (OWSD) and Nigeria Women in Information Technology (NIWIIT). Her research areas include: Data and Information Governance, Data Quality, Experience Management, and Management Information Systems. She can be reached via oladele.ot@unilorin.edu.ng



Dr. (Mrs) Ghaniyyat Bolanle Balogun is a Lecturer in the Department of Computer Science, the University of Ilorin, Nigeria. She obtained a Doctor of Philosophy (PhD) in Computer Science from the University of Ilorin. She also obtained a Master of Science Degree (M.Sc.) in Mathematics with a Computer Science option from the same university. She holds a professional qualification in Business Administration (MBA) which she obtained from the University of

	Ilorin. Her first Degree is a Bachelor of Technology (B.Tech) degree in Mathematics/Computer Science obtained from the Federal University of Technology, Minna, Niger State, Nigeria. Her research interest lies in the fields of Data Structures, Big Data and Data Mining, File Systems and Database Management Systems. She can be reached via email at balogun.gb@unilorin.edu.ng.
	Peter Ogirima SADIKU is a Lecturer at the Department of Computer Science, University of Ilorin, Nigeria. He holds a PhD and an MSc. Degrees in Computer Science, from the University of Ilorin. He has supervised students both at the undergraduate and postgraduate diploma levels. His research areas include: Machine Learning, Biometrics, Information Security and Artificial Intelligence. Email: sadiku.po@unilorin.edu.ng
	Michael Abayomi OLATUNJI is a lecturer at Kings University, Ode-Omu, Osun State. He holds a Bachelor of Science and a PhD in Computer Science from the University of Ilorin, as well as a Master of Science in Computer Science from the University of Hertfordshire, Hatfield, United Kingdom. His research interests include machine learning, cybersecurity, optimization, data science, and software engineering. He can be reached via email at mikeolaus@gmail.com or by phone at +234 903 492 1020.
	Tinuke Omolewa OLADELE is a Professor in the Department of Computer Science, at University of Ilorin, Ilorin, Nigeria. She holds a BSc. in Computer Science from the University of Benin, Benin City, Edo State, with an MSc. Mathematics (Computer Science Option) and PhD Computer Science from the University of Ilorin, Ilorin, Kwara State, Nigeria. She studied Post Graduate Program in Data Science and Business Analytics (PGP DSBA) at McCombs School of Business, University of Texas at Austin, USA. Her research interests include Bioinformatics, Artificial Intelligence - Machine Learning and Deep Learning, Data Science, and Cyber Security. She is an INSA TATA Fellow at the Bioinformatics Unit, Centre for Cellular and Molecular Biology (CCMB), Hyderabad, India. Member of Societe Marocaine de Bio-Informatique (SMBI). Member of Data Scientists Network (DSN). She can be reached via email: oladele.to@unilorin.edu.ng.

