Network Intrusion detection system using artificial intelligence

Clementine Mamogale - 216025117

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
clementinemamogale@gmail.com

Abstract. The widespread use of the internet and communication devices and the development of new technologies that can connect to one other has advanced, which increased the volume of data transfer over the internet. Simultaneously, this has opened many novel attacks which are beyond the human control capabilities, which also poses challenges to the current network securities. One tool that is used by the network securities to prevent possible network intrusion is an Intrusion Detection System (IDS), which works by monitoring the traffic over a network to ensure the availability, integrity, and confidentiality of the network. This tool used to be effective over the past years, but as the internet grows, the IDS has become less and less effective. Hence in this paper, I am introducing a Network Intrusion detection system (NIDS) architecture that uses Artificial Intelligence (AI) as its backbone. This architecture will feature deep learning and machine learning techniques to develop a system that is adaptive and resilient to network intrusions, including zero-day intrusions. To demonstrate the effectiveness of this model, we use the old KDD+ and the new CICIDS 2017 datasets. The reason for these two datasets is because the KDD+ is widely used because it has well-known intrusions and can be used to create the baseline of the NIDS. Then the CICIDS is new and has new attacks that happened recently. Hence it can be used for training for zero-day attacks.

Keywords: Network Intrusion Detection System (NIDS), machine learning, deep learning, Network Security, Support Vector Machine, Naïve Bayes, Neural Networks.

1 Introduction

In this modern life, our daily activities are influenced by the widespread use of computer devices, which are connected to one another and share data and information. Using the information and communication technology (ICT) and its resilience, businesses and individuals were able to offer real-time global business continuity and a frontier solutions of interoperability [1]. The rise of this interoperability and data exchange among computers on the internet has opened vulnerabilities that are exploitable, and they can result in harmful effects on the end-users of the computers. Hence a resistant and resilient intrusion detection system (IDS) is needed, an IDS which will be able to maintain and provide availability, integrity, and confidentiality of network. After all, an IDS is recognized as the first layer of defense among the defensive mechanisms that address all the attack vectors [1].

A Network Intrusion Detection System (NIDS) works by scanning a network traffic as it comes and identify any violations if they exist based on customized detection levels that are preconfigured on the network and then report them, then block them from going through the network before they cause any damage to the data that is being protected by the network. IDS use the idea of behavioral features to differentiate between a legitimate and non-legitimate traffic. Hence the development of Intelligent Intrusion Detection system is needed, which will be able to learn about different behaviors of normal and abnormal network and use that learned behaviors to detect intrusions [2]. In this paper I propose a Network Intrusion Detection System (NIDS) that uses AI as its backbone to detect intrusion on a network in real time, automatic and fast, which aims to provide minimum false alerts. The KDD+ and CICIDS 2017 datasets will be used to demonstrate the effectiveness of the system and it will make use of the Naïve Bayes classification technique, Support Vector Machine and the Neural Networks which will work together to complement each other for maximum security of a network.

****************talk about the layout of the paper here*********************

2 Literature review

The concept of intrusion on a network have been around for a very long time and having intrusion detection systems which worked by monitoring networks for any anomalous behaviors or misuse. It is only now that there is a rise of this intrusions on networks, and mainly because of the rise of the internet with Internet of Things connecting and sharing data over networks [4].

James Anderson wrote a paper in 1980 for a government organization, which is accredited for introducing automatic IDS, a paper that was used to build the first IDS [4]. With the rise of the internet, some problems for that system arose, since it was using the signature based algorithm, it was a bit slow in terms of registering new signature from new threats and the system was only effective against known threats. Zero-day attacks could compromise it, easily [4].

Since then, other models have been developed in the industry to try and fix the problem that were present on the system that was built using the paper from James Anderson. Until the year 1999 when Todd Heberlein introduced a different kind of system called Network Security Monitor (NSM) which helped in monitoring and collecting data about network traffic [4]. This opened new ideas and there were more interests in the intrusion detection environment and new investments were made towards the market. This project featured some of the other projects that were made before it, such as the Distributed Intrusion Detection System which introduced the idea of hybrid intrusion detection. As a results, the IDS field was revolutionized and brought to the commercial world [4].

Currently, by statistics. It shows that in the market of security vendors, IDS remains one of the top selling ones. Some of the commercial IDS that were introduced in the early years includes Network Flight Recorder (NFR) which uses libpcap, it was introduced in 1999, then 1998 APE which was later changed to Snort which is a packer sniffer that was also developed. It uses libpcap also. And it remained the world's largest IDS that was used, with active users of over 300 000 [4]. However, due to fast growth of the internet and almost everything connecting to the internet. These methods become ineffective. at least, on their own. Hence the introduction of NIDS which uses AI which features more than one technique to get the work done.

2.1 Different types of intrusion detection systems

- **Host based intrusion detection system (HBIDS)** runs and monitors the network of machines, at the endpoints on which they are installed on. NBIDS analyzes and log any behaviors that are malicious. It can be used to view what is happening on security systems that are critical [5][7]. But it is not enough on its own since it does not give a bigger picture of the whole system, hence it must be used in conjunction with other IDS [7].

- **Network intrusion detection system (NIDS)** runs on the network and monitors the whole system at the network level, it complements the flaws of HBIDS to securely protect an organization's system. Unlike the NBIDS, they scan data packets and check for any signs of malicious activities [5]

- **Network node intrusion detection system (NNIDS)** works like to NIDS, but on micro level. It checks for malicious activities and threats on each node that is connected to the network [5]. NNIDS adds more security to the NIDS to protect a network system.

All these types of intrusion detection systems can be approached using Signature-based IDS (SIDS), Anomaly-based IDS (AIDS) or a hybrid of both Signature and Anomaly-based IDS. Their differences are that SIDS is knowledge based, it has signatures of known attacks stored in databases and use them to compare with the data patterns on the network as they come [3]. SIDS is very good for known attacks, but very poor for zero-day attacks. On the other hand, AIDS is based on machine learning algorithms, a profile is built for a normal activity, and any activity that does not satisfy that profile, it is considered abnormal and get flagged [3]. AIDS is good for detecting not known intrusions, but is it is not perfect since it can misclassify activities. A hybrid of both SIDS and AIDS is combining both SIDS and AIDS for maximum protection of systems.

2.2 Recent similar work

Ensemble Core Vector Machine (CVM) which is a better version of Support Vector Machine (SVM), which are a supervised learning machines. They are used for regression analysis and classifications [5]. The algorithms that are used in CVM and SVM are based on concept of minimum Enclosing Ball basis, which is an adopted function from statistic. The function helps in producing less false-positive rates and improvement in performance. Hence, they are considered the best prediction methods. However, unlike the SVM, the CVM has low overhead computation which makes it better than the SVM [2].

Internet of Things (IoT) is another field of study that has shown some interesting studies in IDS. IoT is a new and growing technological advancement, whereby the day-to-day “smart” devices, from different domains, connect and communicate to each other over the internet via data sharing [6]. These smart devices are embedded with sensors which generate high volume of data on the internet [5]. IoT brings lots of benefits to out everyday lives, but they also have vulnerabilities that can be exploited over the internet. These vulnerabilities on these devices exist because most of them lack hardware security support, and their gateways also does not support such securities. IDS require high computational power which these smart devices gateways do not provide. They also have their own specifications and protocols that they follow. Implementing these complicated hardware and software security features on these smart devices, can cause them to misbehave or break. As a results, it remains a challenge to secure the environment of these smart devices while having them working perfect [6].

There is a huge gap between security capability and requirements, which puts these smart devices at risks of cyber-attacks such as DoS and spoofing attacks. Many records of IoT devices being attacked are recorded, and it shows how bad they can affect these devices, such as damaging the hardware, or interrupting the system. The Marian malware is one example that was recorded to be the most famous attack that happened in 2017 which affected 300 000 IoT devices, which further turned the devices into zombies to create DDoS attack [6].

For security purposes, IDS for smart devices are placed at the gateways to monitor the network for incoming and outgoing traffic and checks for malicious activities. This is still not enough since it does not check the internal interconnected traffic of the devices, which might cause internal attacks [6]. A better solution that was proposed for IoT was the use of Deep Neural Network (DNN) which will learn and create profiles for abnormal and normal networks from already existing datasets and based on that, it can find abnormalities in a network easily, and it get deployed live on the IoT. This method worked so well because it could identify even non-linear complex features which are used in real life attacks [6].

3 Proposed Method

This paper proposes a network intrusion detection system that uses artificial intelligence, the processes are automated to efficiently detect intrusions on a network and act accordingly. Different machine learning (ML) algorithms are used and compared to pick one that is the best. Deep neural networks (DNN), Support vector machines (SVM) and Naïve Bayes classifier are the three types of ML algorithms that are being compared. All of them are supervised learning machines. It can be said that DNN create the best model for our NIDS. This might be influenced by the fact that DNN uses unlimited hidden layers which helps the model to learn and give better predictions.

3.1 Deep Neural Network (DNN)

DNN is a better version of Neural Network (NN), which is a supervised type of machine learning algorithm that teaches computers to process data, learn from it and make predictions. The NN is inspired by how the human nervous system of the brain works. NN is made up of processing elements called nodes or neurons, which are connected to one another to form layers of the NN system. These layers are organized to form input layer, many hidden layers, and output layer [1,3].

DNN is a creative system, unlike the NN because it can process complex data and make predictions on non-linear data. It can recognize sounds, voices, and graphics, then do some expert review on those data, and also do other functions such as creative thinking and predictions from those complex data [1].

The neurons in the DNN are nothing but mathematical functions which, when give some input data via the input layer, they process it through the hidden many layers and then give out an output via the output layer. The inputs of the neurons and its parameters will determine the output. And these parameters can be adjusted to produce desired outputs from the network [3]. The movement of data from input, processing and output is called forward propagations, how it works is that outputs of one layer, becomes inputs to the next layer, until they reach the final output layer to make a prediction. Sigmoid function which, given an input produces an output between zero and one, is used to normalize the data as it moves from one layer to another, and the higher the value the sigmoid function produces, will determine which node to be activated in the next layer [3].

Once a prediction is made, we can make a cost function, which is a function to check how correct is the prediction, which is compared with the known truth to see the difference between the predicted value and the truth value. The aim is to make sure that the cost function value is as low as possible. This can be achieved by using an algorithm that will update the parameters' values of the network to produce const function's value that is as low as possible [3].

A gradient descent is used as the algorithm to produce such minimal value of cost function. The gradient descent function will update the values of the biases and weights of the network to produce the minimum value of the cost function. This process if called back propagation [3].

3.2 Support Vector Machines (SVM)

SVM are types of supervised learning methods used for regression, classifications and outlier detections. SMV are mathematical equations that aim to give the most accurate answers in machine learning (ML). SVM are based on the vector machine (VC) theory which is from the statistical learning frameworks which was proposed by Vapnik and Chervonenkis [8]. SVM uses classes to categorize data given as inputs during training and testing. They use a decision boundary to choose a maximized distance from the classes using their closest data points. This boundary line is called a maximum margin hyper line.

A simple linear SVM classifier make use of classes that are separated by a straight line between them and data on one side of the straight line are categorized as one and data on the other side of the line are categorized as different from the data on the other side of the staright line []. This means that there can be infinit number of lines to choose from. The best line is the line that is the farthest away from the closet data points, or a line that maximize the margins.

Different kernel functions exist for SVM, this makes it easier to choose which function to pick based on the type of data that will be processed.

- **Linear functions** – These functions are commonly used in text classification because usually these kind of problems can be solved linealy. Linear functions are fast and they work well when there are lot of features in the problems. The formula for this function is: **f(x) = w^T * X * b**, where by w is the weight of the vector that need to be minimized. X is the data that need to be classified and then b is the estimated linear coefficient from the training data.
  In **linear function**, hyperline can be defined using some set of points of **x** such that **w * x + b = 0** This equation is useful because any point of **x** that does not satisfy this equation it mean that point belongs to one of two classes, where by the other class will satisfy this equation **w * x + b = -1** for any values of x and the other class will satisfy this eqation **w * x + b = 1** for any x values that are encountered. This means that the hyberline devides the data points according to their classes in this range {-1, 1}
- **Polynomial function** – This function is not used often in practice because of its poor accuracy and the poor computation compared to the other kenerl functions. The formula for this function is: **f(x1, x2) = (a + x1^T * x2) ^b**
- **Guassian Radial Basis Function (RBF) –** This is the mostly used kernel function and it is so powerful. It is used usually for non-linear data processing. The formula for the function is: **f(x1, x2) = exp(-gamma * ||x1-x2|| ^2)**
  The gamma in this equation represent how much influence does one single training point has on other data points around it. The dot product between the features is represented by ||x1 – x2||

There are other kernel functions that are used in SVM such as the sigmoid, the hyberbolic tangent, ANOVA radial basis, and many more. The selection is based on the computational speed and the accuracy they produce.

3.3 Naïve Bayes

Naïve bayes classifiers are family of classification algorithms that are based on the Bayes' theorm. The algorithms share a common principle that is based on the fact that features that are being classified are independent of each other. This means that the presence of a particular feature in one category is unrelated to any feature that might be present on the data on the other gategory that is being classified.

Naïve Bayes are amongst the Bayesian network (BN) models which is a probabilistic graphical mode that represent knowledge about uncertain domain. It has nodes and edges, where by the nodes represent random variables and then the edges represents conditional probability for the corresponding nodes [9]. BN are sometimes called Bayes Nets or belief networks. NB are directed acyclic graph whereby self connections or loops are not allowed. This is due to the conditional probabilities and dependencies. BNs are ideal for situations where an event has occurred and finding the likelihood that one of the known causes from the several possibilities was the contributing factor [9].

Naïve Bayes classifies are easy to build, they are highly scalable and very useful for very large dataset. Along with that, they are known to outperform evn highly experienced classification methods that exists.

Below is the formula of Naïve Bayes

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

- *P(Y|X)* – posterior probability of class Y, which is the target, when predictor X, which is the attribute.
- *P(Y)* – prior probability of the class
- *P(X|Y)* – the likelihood. i.e. the probability of predictor, given the class
- *P(X)* – the prior probability of the predictor

How it works:

To train and create a model. Firstly a dataset is provided and it will be divided into two classes namely **feature matrix** and **response vector.** The dependent features are contained in the feature matrix rows (vectors) of the datasets and the class variable (output or prediction) of the datasets are contained in the response vector. The concept of assumption for Naïve Bayes features is that each of them contribute equally towards the outcome, each feature is given the same influence or importance, and all of them are independent of each other.

Using the above Naïve Bayes equation, the class ***P(Y|X)*** can be calculated given then features by firstly creating a **frequency table** for each attribute against the target. Then using the **frequency table** to create the **likelihood table**. Then by using the naïve Bayes equation to find the posterior probability for each class. Then finally checking the posterior values for each class and the class with highest outcome value is the prediction. When working with Naïve Bayes, a 'zero-frequency-problem' which iss when a test data has a category that was not present during training data and that data category will be given the value zero and hence when multiplied by the other features, it give the prediction of zero which will always not be taken as the final answer. And to solve that problem, a value of one is added on all categories attributes so that no category is zero on the datatsets.

The **Gaussian Naïve Bayes** is a proposal for continues data to complement the Naïve Bayes classifier. An assumption in continues data is that the continuous data that are associated with each class are distributed according to the the Guassian (Normal) distribution. The likelihood assumption for the features follows this formula:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Where the sigma is the variance and mu is the mean of the continuous variable Xi computed for a given X|Y

The formula calculate the friquencies for the probabilities for input values of each class. And for the entere distribution, the standard deviation and the mean for X's can be calculated as follows:

Mean: **m(x) = 1/n * sum(x)**

Standard deviation: **st(x) = sqrt(1/n * sum(xi-mean(x)^2))**

Where,

- x is the input variable,
- sum is the sum function
- sqr is the squar root function
- n is the number of instances
- xi is a specific x value

The Gaussian Naïve Bayes classifier is simple and quick classifier technique that works so well without putting to much effort and complexity. And it has better accuracy results.

https://reader.elsevier.com/reader/sd/pii/S1877050918321136?token=4606A27E76A37AFD6B26EC86334757868E24A175B8E3582D171D805F91F0EDD9458B69078AFBC2E04E5A6A540FB07321&originRegion=eu-west-1&originCreation=20220725201432

https://reader.elsevier.com/reader/sd/pii/S1877050921011078?token=077DC3F7EC539AD3916AF2141263849590B01B72E5D7E7CECED6B96C2ED1228835AD1572DF2B16A3151026CB414DB861&originRegion=eu-west-1&originCreation=20220725201650

https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

https://towardsdatascience.com/naive-bayes-explained-9d2b96f4a9c0

https://www.geeksforgeeks.org/naive-bayes-classifiers/

https://en.wikipedia.org/wiki/Naive_Bayes_classifier

4 First Section

4.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either.

Subsequent paragraphs, however, are indented.

Sample Heading (Third Level). Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings.

Sample Heading (Forth Level). The contribution should contain no more than four levels of headings. The following Table 1 gives a summary of all heading levels.

Table 1. Table captions should be placed above the tables.

Heading level	Example	Font size and style
Title (centered)	**Lecture Notes**	14 point, bold
1st-level heading	**1 Introduction**	12 point, bold
2nd-level heading	**2.1 Printing Area**	10 point, bold
3rd-level heading	**Run-in Heading in Bold.** Text follows	10 point, bold
4th-level heading	*Lowest Level Heading.* Text follows	10 point, italic

Displayed equations are centered and set on a separate line.

$$x + y = z \tag{1}$$

Please try to avoid rasterized images for line-art diagrams and schemas. Whenever possible, use vector graphics instead (see Fig. 1).

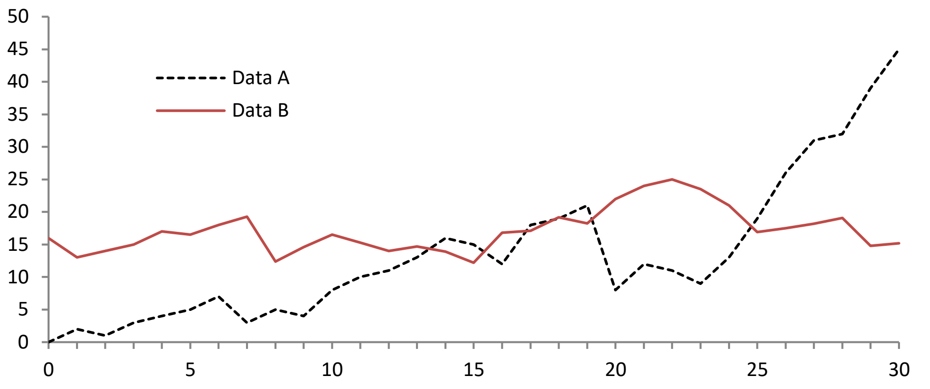

Fig. 1. A figure caption is always placed below the illustration. Short captions are centered, while long ones are justified. The macro button chooses the correct format automatically.

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], as well as a URL [5].

References

1) Author, F.: Article title. Journal 2(5), 99–110 (2016).
2) Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
3) Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
4) Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
5) LNCS Homepage, http://www.springer.com/lncs, last accessed 2016/11/21.

Klngfdnkddnkjnkngfd0

1. https://reader.elsevier.com/reader/sd/pii/S1877050921011078?token=B9BD6FFE3C86E2DD74A456A01956CF6DA067E92A474645A6F1360558C7F127895C1D0E990B5C4E33630BC1CF08F91064&originRegion=eu-west-1&originCreation=20220612193324
2. https://reader.elsevier.com/reader/sd/pii/S1877050918321136?token=71A22DCD7E8606EF7E519D3A1E7939A5D7DCC70FB252A9610845A124066DE2BA46EEC85A07DD74A4E29468BC9D812487&originRegion=eu-west-1&originCreation=20220612193328
3. https://www.researchgate.net/journal/Transactions-on-Emerging-Telecommunications-Technologies-2161-3915/publication/344726867_Network_intrusion_detection_system_A_systematic_study_of_machine_learning_and_deep_learning_approaches/links/61a604c86864311d938a92c5/Network-intrusion-detection-system-A-systematic-study-of-machine-learning-and-deep-learning-
approach-

es.pdf?_sg%5B0%5D=3jDD1L58VEnesSzxdTvYvwp0_61rq6ICOLKl2tjjxvTed3YAWsWX1Q2dcaSsqUPCQlbRSAcqxceJPx-oDt5DdSg.JQJoBClZHuZcttTiLuMOW9UQnisaHqaJSDDs2oMBM9v1hKbmtPj0EtMlE5h3jY_ZGyqDu7T6anRV87RsntUh0w&_sg%5B1%5D=uQS6nkX6_GaKkhJD6vUMzkZKM7ySW_iU1XkoxouJsHKLna-CYA4zsKXdCEG3H_Y9BkeIFaegXckRbCvEEqs_9a0Sdvs6hd82ofvwu1AYxRSlH.JQJoBClZHuZcttTi-LuMOW9UQnisaHqaJSDDs2oMBM9v1hKbmtPj0EtMlE5h3jY_ZGyqDu7T6anRV87RsntUh0w&_iepl=

4. https://reader.elsevier.com/reader/sd/pii/S1877705812021613?token=B75AFECDC757F2A8D54A0A4081F752ED705134BDCEB05B6A7237305EE9E84A908F48746A94C60ED84B82A7FCE295F847&originRegion=eu-west-1&originCreation=20220613230743
5. https://mdpi-res.com/d_attachment/mathematics/mathematics-10-00530/article_deploy/mathematics-10-00530-v2.pdf
6. https://iopscience.iop.org/article/10.1088/1742-6596/1518/1/012040
7. https://link.springer.com/article/10.1007/s13198-014-0277-7#Sec1
8. https://towardsdatascience.com/implementing-svm-from-scratch-784e4ad0bc6a
9. https://www.sciencedirect.com/topics/mathematics/bayesian-network