A System to automate the development of anomaly-based network intrusion detection model

B Padmaja1, K Sai Sravan2, E Krishna Rao Patro3, G Chandra Sekhar4

1Associate Professor, Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India
2Student, Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India
3Assistant Professor, Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India
4Assistant Professor, Department of Computer Science and Engineering, Institute of Aeronautical Engineering, Hyderabad, India
Email: b.padmaja@gmail.com

Abstract. Cyber security is the major concern in today’s world. Over the past couple of decades, the internet has grown to such an extent that almost every individual living on this planet has the access to the internet today. This can be viewed as one of the major achievements in the human race, but on the flip side of the coin, this gave rise to a lot of security issues for every individual or the company that is accessing the web through the internet. Hackers have become active and are always monitoring the networks to grab every possible opportunity to attack a system and make the best fortune out of its vulnerabilities. To safeguard people’s and organization’s privacy in this cyberspace, different network intrusion detection systems have been developed to detect the hacker’s presence in the networks. These systems fall under signature based and anomaly based intrusion detection systems. This paper deals with using anomaly based intrusion detection technique to develop an automation system to both train and test supervised machine learning models, which is developed to classify real time network traffic as to whether it is malicious or not. Currently the best models by considering both detection success rate and the false positives rate are Artificial Neural Networks(ANN) followed by Support Vector Machines(SVM). In this paper, it is verified that Artificial Neural Network (ANN) based machine learning with wrapper feature selection outperforms support vector machine (SVM) technique while classifying network traffic as harmful or harmless. Initially to evaluate the performance of the system, NSL-KDD dataset is used to train and test the SVM and ANN models and finally classify real time network traffic using these models. This system can be used to carry out model building automatically on the new datasets and also for classifying the behaviour of the provided dataset without having to code.

1. Introduction
In today’s modern world, the Internet is omnipresent. So, the increasing access to the web pages via the internet creates many serious security breaches. Nowadays, such types of attacks occur frequently and it is not easy to detect and terminate such attacks with proper efficiency. One of the most widely seen attacks employs sending a huge number of client requests to the server and the server, which is unable to handle such huge requests, will make the entire client-server architecture passive by shutting down the server’s functionalities. This attack is also called a distributed denial of service (DDOS) attack, which accounts for a major security threat to internet services. Detection and termination of DDoS is one of the most challenging tasks for today’s organizations using the internet. There are different strategies adopted to handle and process the network traffic information collected by
monitoring stations, i.e., Servers and Routers, to distinguish malicious traffic entering the system such as DDoS attacks, Probing attacks, Root to Local attacks, and User to Root attacks from safe and normal traffic in Intrusion Detection Systems (IDS). Generally, in Anomaly based IDS—which is the current paper’s adopted technique, Machine learning techniques are designed and implemented with the intrusion systems to safeguard the organizations from notorious and harmful traffic. Specifically, supervised clustering techniques allow to effectively distinguish normal traffic from malicious traffic with good accuracy. In this paper, the most established and efficient machine learning algorithms, SVM and ANN, are used to automatically train and detect these attacks collected from the “NSL-KDD” dataset using an automation system that has been developed using python libraries and Mysql. Proper pre-processing and wrapper feature selection technique is employed on the dataset to enhance the performance of the classifiers and reduce the detection time. The classification algorithms like C4.5 decision trees, K-Nearest Neighbors, and Navie Bayes which was previously applied on the training dataset [1] has shown much lower accuracy when compared to SVM and ANN. The performance comparison of algorithms for every training and testing iteration is also shown using bar graph and it is found that Artificial Neural Networks (ANN) is more efficient in detection of these attacks even though SVM was already established with better accuracy because of it’s high false positive rate This automation system that has been developed can be used as proper Intrusion detection system with a friendly GUI.

As shown in Fig.1, the machine learning system is built using SVM and ANN by following the above architecture to make use of the best of both models at different instances and with different datasets. The NSL-KDD dataset containing a total of 41 refined features is initially normalized. Later, the wrapper feature selection strategy is applied on the normalized dataset to pick up the most relevant and important features as discussed in [1]. In Wrapper feature selection technique, a classifier model is initially selected. Then, a subset of features from all of the available features are sent into this model in a recursive way. Based on the model’s classification accuracy, the usefulness of a particular feature is determined. At the end, top N features with high usefulness are selected for model training. NSL-KDD dataset has 43 attributes for each of the connection records which include class labels containing attack types. The attack types are categorized into four attack classes as described in [12].
1. Denial of Service (DoS): is an attack in which the hacker sends too many requests to the server and makes the server overloaded with requests, after which the server would become unresponsive to any further request and makes the system passive.
2. Probing Attack (Probe): is used for probing a network of computers so that it would gather all information to be used to compromise its security controls.
3. User to Root Attack (U2R): It is one of the classes of system exploits in which the hacker starts out with having access to a normal user account on the system which is achieved either by password sniffing, dictionary attacks, or social engineering. Then the hacker is ultimately able to exploit the vulnerability of the system to gain root access to the system.
4. Remote to Local Attack (R2L): This attack generally occurs when a hacker who has got the ability to send packets to any machine over a network but who does not have any record on that machine tries to exploit a certain vulnerability of the system to gain local access as a user of that specific machine.

Out of these 43 attributes, 41 attributes refer to the traffic inputs. 42nd and 43rd attributes refers to the output labels (normal or attack) and score—the traffic input’s severity, respectively.

The feature types in this dataset can be categorized into 4 different types as mentioned in [20]: Categorical, Binary, Discrete and Continuous.

This dataset is an improved version of “KDDCUP99” dataset, in which all the redundant records and inconsistent data has been removed to help create much more efficient machine learning models. Most of the redundant data was found in the features which fall under U2R and R2L. So, the NSL-KDD dataset is later found to have very few redundant records in these two categories because of the removal of a lot of redundant data from these categories.[21]

Figure 2, depicts the entire flow of events in the proposed system. It starts at the authentication point, where the user needs to type in their credentials to get into the system. Once logged in, they have an option to either classify the data by using previously saved models or train and test the new model using both SVM and ANN algorithms. Finally the user can use Accuracy Graph to analyze the performance of both the algorithms.
2. Related Work

Signature based IDS is by far, the only well established technique to detect the network intrusions. This technique was initially used only by the system antivirus developers. A signature-based Intrusion Detection System approach will typically keep monitoring the network traffic to find the patterns and
sequences that match any particular attack signature. Attack signatures are the files that contain very specific arrangements of information that provide an identity to the known attacks that have been stored in the Signature-based IDS database. These signatures will help the IDS system to detect if any attacker attempts to attack the system. These will be found within the HTTP headers of the network packets as well as in the sequences of data that match known viruses or other patterns which are malicious. An attack signature is also found within the source or destination network addresses as well as in very specific data sequences or a series of packets. Any signature-based detection uses a very well known list of IOCs, where IOC stands for Indicators of Compromise. These IOC’s usually contain specific network attack behaviors, malevolent domains and known sequences of bytes. They might also include file hashes and subjects of the email lines.

From Fig 3, it is evident that the traffic collector acts as the heart of the signature based IDS system by collecting the real time traffic and reshaping the traffic for the Signature based IDS block in the system. Once the traffic is sent to the Signature based IDS block, it matches the signature with the already stored signatures in the Database. If there is a mismatch, the system automatically reports the intrusion.

![Signature Based Intrusion Detection System Architecture as depicted in [8]](image)

The major limitations of signature-based IDS approach is its incapability to detect the attack signatures that were never seen or stored before. Notorious attackers can simply modify their attack sequences within the malware that they are trying to inject and other types of attacks, just to avoid being noticed by the IDS. In this way, the shield that this IDS is providing for the users can easily be broken if the attackers simply adapt different set of attacks which are completely different from traditional ones.

2.1. Signature based IDS vs Anomaly based IDS

As mentioned earlier, the major disadvantage with signature based IDS is that it cannot work on unseen attacks. The methodology used in this paper is completely anomaly based Intrusion detection, i.e, this paper uses anomaly based intrusion detection techniques to eliminate the major disadvantage of signature based IDS. The anomaly based techniques generally involve using the machine learning
techniques to train the system in such a way that it could work well not only with known attacks but also with unknown attacks. Anomaly based IDS is still growing and there is no concrete system that uses 100% anomaly based intrusion detection technique [1].

3. Proposed Method

There As mentioned earlier, the major disadvantage with signature based IDS is that it cannot work on unseen attacks. The methodology used in this paper is completely anomaly based Intrusion detection, i.e., this paper uses anomaly based intrusion detection techniques to eliminate the major disadvantage of signature based IDS. The anomaly based techniques generally involve using the machine learning techniques to train the system in such a way that it could work well not only with known attacks but also with unknown attacks. Anomaly based IDS are still growing and there is no concrete system that uses 100% anomaly based intrusion detection technique [1].

3.1. Login/Signup

- Login: If the user is already registered with the system, the user can simply login to the home page using his ID and Password. The login credentials will be stored in the database once the user is registered and will verify the presence of those credentials entered by the user in the users table, every time the user logs in. If there is a match, a positive response will be generated by the Login system which will allow the user to enter the home page. Otherwise, the Login system will produce a negative response and will put the user in the same Login page.
• Signup: If the user is new to the system, he/she will register with the details that are asked in the registration form. These details will then be appended to the users table in MySQL database to register the user for future authentication.

• MySQL database: The Python programming language needs MySQL driver software to connect with the MySQL database. “MySQL Connector” is used in the proposed system to connect to the MySQL database.

The syntax of connecting the python script with the database using this driver is:

```python
from mysql import connector
mydb = connector.connect(
    host="Host address",
    user="User Id",
    password="User Password")
```

Here, the Host address is the location from where the SQL database is actually accessed. In general, it could be LocalHost(127.0.0.10) or the address of the web host who is hosting the database.

Figure 5: MYSQL driver connection with python
Figure 5 depicts the systematic procedure with which python scripts connect to a MySQL database using a database driver.

Procedure for connection:

- Mysql.connector.connect() will invoke a connection request
- Connection.cursor() will connect the MySQL database to the MySQL driver.
- Cursor.execute() will perform all CRUD operations on the actual database by using the database driver.

3.2. Authentication of the User:

As the user logs in to the system, the system sends a query request using the database driver to check if the person with the given username and password exists in the Users table or not. If the query returns a positive response, the system will redirect the user to the homepage of the system, otherwise the system will warn the user that the credentials are incorrect and make the user get stuck in the same login page until he/she enters correct credentials.

3.3. Dataset Uploading:

FileDialog from the TkInter library is used to upload the files onto the proposed system after which, the further processing and intrusion detection identification can take place.

3.4. Model Generation and Network Intrusion Detection:

Once the dataset is uploaded, the system can be used to train an IDS model using SVM algorithm and also using ANN algorithm. The Scikit learn package is used to train the model under SVM and the Keras package is used to train the model using ANN. Systematic Hyper-parameter tuning is done to find the best hyper-parameters that could give the highest accuracy rate and at the same time, low false positive rate. Once the models are trained, the models are stored in the local storage of the user system. This is done using the Pickle package. Once the model is saved, it can be accessed by the system using the same file location that was used during saving the model, and perform intrusion detection on the unseen data. This process is clearly mentioned in Fig 4.

4. Results and Discussions

As discussed earlier, the first state of the proposed system is the Login/signup page which is styled using TkInter library.
As shown in Fig 7, we use the button “Switch to Sign Up”, to go to the Sign Up screen which is shown in Fig 8, and “Switch to login” in the Sign Up screen to go back to the login screen.

After successful Login, the flow of events in the system will be as follows:
In Fig 9, ‘Upload NSL KDD Dataset’ button is clicked to upload the dataset.

Now, the “Preprocess Dataset” button is clicked to preprocess the uploaded dataset.
After preprocessing, all string values are removed and attack names of type String are converted to numeric values such that normal signature is mapped to ‘0’ and anomaly attack signatures are mapped to ‘1’. Also, all the numeric features are normalized using standard normal curves such that each feature set has mean of ‘0’ and standard deviation of ‘1’.

| Learning Algorithm | Number of features | Detection Accuracy |
|--------------------|--------------------|--------------------|
| SVM                | 17                 | 84%                |
| ANN                | 35                 | 85.5%              |
| SVM                | 17                 | 94.5%              |
| ANN                | 35                 | 83.5%              |

**Table 1** Results of classification with different number of features [1]

Table 1 shows the performance of the models with both 17 and 35 as feature numbers since these numbers have shown best results for ANN and SVM models respectively. Out of these two numbers, since 17 features gives the highest accuracy for ANN, top 17 features from a total of 41 features are selected using the wrapper feature selection technique. In this technique, Random Forest Classifier is used as a base model for figuring out the importance of all the features with repeated calls to the model for batches of 17 features in every iteration.

![Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection](image-url)

**Figure 11**: Dataset Preprocessing
Figure 1 depicts the results from the wrapper feature selection technique. In this figure, the most important feature was found to be ‘source bytes’, which is to say, the source from which the bytes are coming and the least important is found to be ‘is_host_login’, which is to say whether the user is logged in to the system while interacting with the internet or not. Top 17 features are selected from these results based on their importance values and only these features will be used in the further steps of the model building process.

Now, when ‘Generate Training Model’ is clicked, it splits the dataset into train and test data as mentioned in Fig 1. Traditional 80-20 split strategy is applied for splitting the dataset.

Figure 12: Feature importance graph

Figure 13: Train Test Split
In Fig 13 we can see a dataset containing a total of 1244 records out of which 995 are used for training and 249 are used for testing. Now ‘Run SVM Algorithm’ is clicked to generate an SVM model and model accuracy is calculated.

In Fig 14 we can see that with SVM we got 84.73% training accuracy. Now to train the dataset on ANN, ‘Run ANN Algorithm’ is clicked to calculate ANN accuracy.

In Fig 15, ANN model was trained with 96.88% accuracy. Now we can click on the ‘Upload Test Data & Detect Attack’ to make the models detect the attacks in unseen test dataset to evaluate the model performances.
Figure 16: Accuracy graph of SVM and ANN models

Figure 16 clearly shows that the testing accuracy of the SVM model is 84%, whereas the testing accuracy of ANN is 94.5%. This proves that when NSL-KDD dataset is used, the model trained of ANN would always give better predicting results compared to any other supervised learning model at present as mentioned in [1].

5. Conclusion

A system that automates the ML model building and testing comes handy for everyone irrespective of their skills in coding and knowledge in machine learning. Thus this system is built on two of the best performing algorithms to ease out the unnecessary tuning and rewriting the complex codes in the context of network intrusion detection. From Fig 16, it is clear that the accuracy of ANN in detecting the network intrusions is more when compared to the SVM algorithm, when the NSL-KDD dataset is used. But these observations are not constant and they change based on the dataset used and the techniques used to preprocess and prepare the data for model generation. So, the proposed system uses both the models to get the best results from both the models performances at different circumstances. As the anomaly based IDS is still under development, there is a lot of scope in coming up with a much more efficient and optimized Supervised Machine learning system and also there is a scope to find more efficient ways to select the best features for much higher accuracy in the models generated. Once such accomplishments are achieved, much more advanced systems can be generated to automate the entire complexity of the anomaly based IDS system.

Acknowledgement

We would like to thank Prof. B Padmaja, Associate Professor & Head, IARE, for her mentorship and guidance throughout the research and would also like to express gratitude to Dr. M Madhu Bala, Professor & Head, CSE department, IARE, for providing us relevant sources and help in documentation.

References

[1] K. A. Taher, B. Mohammed Yasin Jisan and M. M. Rahman, "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection," 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2019, pp. 643-646, doi: 10.1109/ICREST.2019.8644161.
[2] H. Song, M. J. Lynch, and J. K. Cochran, “A macro-social exploratory analysis of the rate of interstate cyber-victimization,” American Journal of Criminal Justice, vol. 41, no. 3, pp. 583–601, 2016.

[3] P. Alaei and F. Noor Bebehani, “Incremental anomaly-based intrusion detection system using limited labeled data,” in Web Research (ICWRI), 2017 3rd International Conference on, 2017, pp. 178–184.

[4] M. Saber, S. Chadli, M. Emharrar, and I. El Farissi, “Modeling and implementation approach to evaluate the intrusion detection system,” in International Conference on Networked Systems, 2015, pp. 513–517.

[5] M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, “Toward credible evaluation of anomaly-based intrusion-detection methods,” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 40, no. 5, pp. 516–524, 2010.

[6] A. S. Ashoor and S. Gore, “Importance of intrusion detection system (IDS),” International Journal of Scientific and Engineering Research, vol. 2, no. 1, pp. 1–4, 2011.

[7] M. Zamani and M. Movahedi, “Machine learning techniques for intrusion detection,” arXiv preprint arXiv:1312.2177, 2013.

[8] Khan, Khalid & Khan, Shafulloha & Altaf, Muhammad & Iqbal, Zeeshan &Mashwani, Wali. (2019). A survey on Intrusion Detection and Prevention in Wireless Ad-hoc Networks. Journal of Systems Architecture. 105. 101701. 10.1016/j.sysarc.2019.101701.

[9] N. Chakraborty, "Intrusion detection system and intrusion prevention system: A comparative study," International Journal of Computing and Business Research (IJCBR) ISSN (Online), pp. 2229–6166, 2013.

[10] Panda, Mrutyunjaya & Abraham, Ajith & Das, Swagatam & Patra, Manas. (2011). Network intrusion detection system: A machine learning approach. Intelligent Decision Technologies (IDT) Journal, IOS Press. 5. 347-356. 10.3233/IDT-2011-0117.

[11] Ahmad, Z, Shahid Khan, A, Wai Shiang, C, Abdullah, J, Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerging Tel Tech. 2021; 32:e4150 https://doi.org/10.1002/ett.4150

[12] Tavallaee, Mahbod& Bagheri, Ebrahim & Lu, Wei &Ghorbani, Ali. (2009). A detailed analysis of the KDD CUP 99 data set. IEEE Symposium. Computational Intelligence for Security and Defense Applications, CISDA. 2. 10.1109/CISDA.2009.5356528.

[13] Khraisat,A., Gondal, I., Vamplew, P. et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecur 2, 20 (2019), https://doi.org/10.1186/s42400-019-0038-7

[14] I. Butun, S. D. Morgera and R. Sankar, "A Survey of Intrusion Detection Methods and Systems," in IEEE Transactions on Software Engineering, vol. SE-22, no. 1, pp. 36-50, 1996.

[15] Z. Tan, A. Jamdagni, X. He, P. Nanda and R. P. Liu(2014), "A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis," in IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 447-456, Feb. 2014, doi: 10.1109/TPDS.2013.146.

[16] D. E. Denning, “An Intrusion-Detection Model,” in IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222-232, Feb. 1987, doi: 10.1109/TSE.1987.232894.

[17] A. A. Aburomman and M. B. IbnKeaaz(2016), “A novel SVM-kNN-PSO ensemble method for intrusion detection system,” Appl Soft Comput, vol. 38, pp. 360-372, 2016/01/01 2016

[18] Alazah, A., Hobbs, M., Abawajy, J., Khraisat, A, and Alazab, M. (2014), "Using response action with intelligent intrusion detection and prevention system against web application malware", Information Management & Computer Security, Vol. 22 No. 5, pp. 431-449.

[19] B Padmaja, B Shashirekha, Sasmita kumara pradhan, T Sahithi, E Krishna Rao Patro, G Akhila (2021), “OASNIDS: A Novel Optimal Acceptance Sampling based Network Intrusion Detection System,” in Journal of Xidian University, vol 15(3), pp. 508-519.

[20] Gerry Saporito (2019), “A Deep Dive in to NSS-KDD Dataset”, Retrieved May 1, 2021 from https://towardsdatascience.com/a-deeper-dive-into-the-nsl-kdd-data-set-15c753364657

[21] Sarika Choudhary, Nishtha Kesswani(2020), “Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT”, Procedia Computer Science, Volume 167, 2020, Pages 1561-1573, ISSN 1877-0599, https://doi.org/10.1016/j.procs.2020.03.367.

[22] B Padmaja, PV Narasimha Rao, M Madhu Bala, E Krishna Rao Patro(2018), “A novel design of autonomous cars using IoT and visual features”, Proceedings of the Second IEEE International conference on I-SAMC (IoT in Social, Mobile, Analytics and Cloud) (I-SAMC 2018), PP. 18 – 21, 2018, ISBN:978-1-5386-1442-6.

[23] B Padmaja, Narasimha Prasad, B Rajkumar Gatadi (2013), “Monitoring and Extracting Abnormalities in Land Surface Temperature Images for Automatic Identification of Forest Fires”, 2013 IEEE European Modelling Symposium, pp no. 202-206, DOI 10.1109/EMS.2013.37.

[24] B Padmaja, Myneni Madhu Bala, E Krishna Rao Patro (2020), “A comparison on visual prediction models for MAMO (multi activity–multi object) recognition using deep learning”, Journal of Big Data, vol 7, issue 24, Springer, 2020.