TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52020

AI-Powered Network Intrusion Detection System (NIDS): An Intelligent Machine Learning Framework for Real-Time Network Threat Detection

Ayushi Srivastava[1], Avais Khan[2], Harshit Chaurasia[3], Homa Rizvi[4]

[1,2,3]Department of Computer Science, Shri Ramswaroop Memorial University, Lucknow, India

[4]Assistant Professor, Department of Computer Science, Shri Ramswaroop Memorial University, Lucknow, India

ayushisrivastava070805@gmail.com, faizahmad756987@gmail.com, harshitchaurasia23082005@gmail.com, homarizvi731@gmail.com

KEYWORD

Honeypot, intent analysis, machine learning, remote shell access, threat detection.

ABSTRACT

With the rapid growth of internet services, cloud computing, and interconnected devices, cybersecurity threats have increased dramatically. Organizations today face numerous network attacks such as Denial-of-Service (DoS), Distributed Denial-of-Service (DDoS), malware injection, phishing, and brute-force attacks. The proposed framework demonstrates how artificial intelligence can significantly enhance modern cybersecurity systems by enabling intelligent and adaptive threat detection. The work develops a network threat detection system, AI@NTDS, that uses the behavioral features of attackers and intelligent techniques. The proposed AI@NTDS system combines data analysis, feature extraction, and feature evaluation to construct a detection model, which supports a more straightforward strategy by which the operating system or its operators can defend against network attacks. The Linux system interaction information of SSH (Secure Shell) and Telnet are obtained from the Cowrie Honeypot and labeled according to Enterprise Tactics of MITRE ATT&CK to ensure dataset credibility. The proposed AI@NTDS system has three levels, depending on the attacker's attacks and the user's risk of damage. Fiftytwo features are used to detect the network threat level. AI-based algorithms LightGBM, Random Forest and the K-NN algorithm are used to verify the identification of the custom features. Finally, the detection model that is trained using the best combination of features is used to predict the test dataset. The accuracy of the proposed AI@NTDS system reaches 99%, 95.66%, and 94.08% with the LightGBM, Random Forest, and K-NN algorithms, respectively. The mutual dependencies of features and network threats are evaluated. Results of a performance analysis reveal that the proposed AI@NTDS system has an accuracy of 99.20% and an F1-score of 99.80%. It is superior to existing detection mechanisms, which it outperforms by 4% and 1% in accuracy and F1-score, respectively

I. INTRODUCTION

The rapid advancement of digital technologies has transformed the way organizations operate. Internet connectivity, cloud services, mobile devices, and Internet of Things (IoT) systems have become an integral part of modern infrastructure. However, this increased connectivity has also introduced significant cybersecurity risks.

Corresponding Author: Ayushi Srivastava, Department of Computer Science, Shri Ramswaroop Memorial University, Lucknow, India
Email: ayushisrivastava070805@gmail.com

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52020

Network attacks such as Distributed Denial of Service (DDoS), malware propagation, data breaches, and unauthorized access have become more frequent and sophisticated.

Network Intrusion Detection Systems (NIDS) are designed to monitor network traffic and detect suspicious activities. Traditional intrusion detection systems are mainly based on signature-based techniques that compare network traffic with a database of known attack patterns. Although these systems are effective against previously known attacks, they fail to detect new or unknown threats. Artificial Intelligence (AI) and Machine Learning (ML) provide powerful solutions to overcome these limitations. Machine learning algorithms can analyze network traffic patterns, identify anomalies, and detect malicious activities even when attack signatures are unknown. By learning from large volumes of network data, AI-based systems can adapt to evolving threats and improve detection accuracy. Venafi, Inc. collects real-world examples of SSH threats [2]. For example, Sony Pictures was hacked in 2014 and SSH keys were stolen, leading to leaks of executive salaries and copies of unreleased Sony movies. The 2019 Kinsing Malware included several shell scripts that download and install, remove, or reinstall various services and programs. The 2020 Kaiji malware detected poorly configured SSH services and performed a brute force attack [3]. The above examples show that SSH attacks involve various behaviors. Therefore, SSH security is critical and user access to remote systems must be carefully monitored. Commands that are executed by a remote connection must be analyzed. This research focuses on designing an AI-powered Network Intrusion Detection System capable of analyzing network traffic and identifying potential threats in real time. The proposed system leverages machine learning algorithms to enhance detection capabilities and provide a more robust cybersecurity solution. In this study, AI-powered techniques are used to solve the command-based content problem and design a network threat detection system, AI@NTDS. Since an enormous amount of information is collected daily, the manual defense of the remote connection threats may cause an irreversible situation. The malicious command dataset for AI Model training is collected and organized by the Honeypot. Most importantly, the problem of detecting malicious commands is solved herein.

2. Literature Review

Several researchers have explored the use of machine learning and deep learning techniques for network intrusion detection. Many early intrusion detection systems relied on rule-based or signature-based detection methods. These systems compared network traffic against predefined patterns of known attacks. While effective for known threats, they lacked the capability to detect new or unknown attacks.

Recent studies have shown that machine learning techniques can significantly improve intrusion detection performance. Algorithms such as Decision Trees, Random Forest, Support Vector Machines (SVM), and Neural Networks have been successfully applied to classify network traffic and detect malicious activities.

Research using the NSL-KDD dataset demonstrated that Random Forest and Decision Tree models achieved high detection accuracy with lower false positive rates. Deep learning models such as Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks have also been used for detecting complex attack patterns in large datasets. However, challenges remain in terms of model accuracy, real-time processing, and reducing false alarms. Therefore, there is a need for more intelligent systems that can efficiently analyze network traffic and detect evolving cyber threats.

3. Problem Statement

Traditional Network Intrusion Detection Systems primarily rely on signature-based detection methods. These systems are limited in their ability to detect new or previously unknown cyber attacks. As cyber threats continue to evolve, signature-based approaches become insufficient for ensuring network security. Additionally, modern networks generate large volumes of data, making it difficult for traditional systems to analyze traffic efficiently. High false positive rates also create challenges for network administrators. Therefore, there is a need for an intelligent intrusion detection system capable of automatically analyzing network traffic, identifying abnormal patterns, and detecting malicious activities with high accuracy. Many researchers have presented solutions to protect users against the command-line-based threat. The main task that will be addressed in this work is the detection of the hacker's malicious intent; 52 features will be provided for the analysis of the AI model. These include message-based, host-based, and geography-based features.

CONTRIBUTIONS

This work contributes to the field by developing an AI-powered network threat detection system, AI@NTDS, which has three levels. The system provided 52 features for the AI-based threat detection datasets. The three main types of features are message-based, host-based, and geographybased. A feature importance analysis demonstrates that Message_Length,

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52020

Execution_File, and Received_Size features for the malicious behavior are more critical than other features. The features proposed herein are effective in detecting remote network connection threats. The performance analysis results herein were much better than those in other studies, revealing that the model in this study had an accuracy of 99.2% and a performance of F1-score of 99.8%.

RELATED WORK

The section will review the latest SSH-based intrusion systems, techniques, and experiments. Descriptions of the experiments have been published in different scientific articles, and various threats have been detected.

ANALYSIS OF ATTACKERS' BEHAVIORS BASED ON SSH SESSIONS

Following the above definition of Honeypot, this subsection will discuss the use of the information collected for analysis of the collected information. The definition and analysis of the behavior in Honeypot using previously developed research methods are described. Esmaeil et.al. proposed a Honeypot technique to investigate violent SSH attacks on academic networks [8]. The most common attack is the strong guess-password attack that targets SSH, FTP, and Telnet servers. Experimental results demonstrate that preset lists of user names and passwords are widely shared and form the basis of violent attacks. Valli et al. [9] used the Kippo SSH Honeypot system to identify the activity in the Honeypot. The system runs on the same hardware and software configuration as above. Data over 75 days were collected as experimental data. An analysis yields the attackers' behaviors and patterns. The experimental results show that the number and range of attacks are different so that the content can be further discussed. Kambourakis et al. [10] discusses the current state of botnets affecting the Internet of Things and the reasons for causes of the success of attacks. They provided detailed information on the operating principles of malware in the Internet of Things, examined their interrelationships, and proposed preventative strategies against malware. Critical steps concerning the operation and communication of botnets have been proposed and six sets of features of Mirai botnets have been identified [11]. That study used the above features to secure IoT devices and protect Internet infrastructure from destructive distributed denial-of-service attacks. Bajtos et al. [12] observed botnets and described the behavior of the first two stages of their life cycle, which are initial infection and secondary infection. They resolved identified the behavioral attributes in each stage and designed a model to determine whether a threat is a botnet. They found that some network sessions and credential guesses are easily collected and usable attributes of the features in profiling threat agents.

4. Objectives of the Research

The main objectives of this research are:

1. To study the concept of network intrusion detection systems.
2. To collect and analyze network traffic datasets for cybersecurity research.
3. To preprocess and prepare data for machine learning algorithms.
4. To implement machine learning models for detecting network attacks.
5. To evaluate the performance of different models using appropriate metrics.
6. To design an AI-based framework for real-time intrusion detection.

5. Proposed Methodology

The proposed system follows a machine learning-based framework for detecting network intrusions. The methodology consists of several stages including data collection, preprocessing, feature selection, model training, and performance evaluation.

5.1 Data Collection

The system uses publicly available cybersecurity datasets such as:

- NSL-KDD Dataset
- CICIDS2017 Dataset
- UNSW-NB15 Dataset

These datasets contain labeled network traffic data representing both normal and malicious activities.

5.2 Data Preprocessing

Before training machine learning models, the dataset must be cleaned and prepared. Data preprocessing steps include:

- Removing missing or corrupted values
- Encoding categorical features
- Normalizing numerical data
- Splitting dataset into training and testing sets

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52020

Proper preprocessing ensures that machine learning models can learn meaningful patterns from the data.

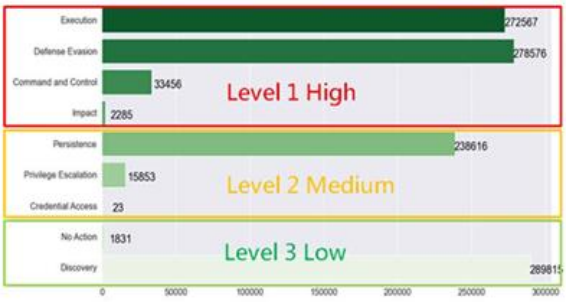

FIGURE 3. Data distribution of aggressive behavior.

5.3 Feature Selection

Feature selection helps identify the most important attributes that influence network behavior. Techniques such as correlation analysis, information gain, and principal component analysis (PCA) can be used to reduce irrelevant features and improve model performance.

Features Set : 52 Features					
F1	*Keyword_bash*	F18	*Keyword_cat /etc (Get_sys_info)*	F35	*Count_Hex*
F2	*Keyword_shell*	F19	*Keyword_uname (Get_sys_info)*	F36	*Count_url*
F3	*Keyword_exit*	F20	*Keyword_wc (Get_sys_info)*	F37	*Message_length*
				F38	*Messages / sec*
F4	*Keyword_help*	F21	*Keyword_crontab (Get_sys_info)*	F39	*Protocol*
F5	*Keyword_passwd [user] (Set_account)*	F22	*Keyword_w (Get_sys_info)*	F40	*Src_Port*
F6	*Keyword_chpasswd (Set_account)*	F23	*Keyword_ps (Get_sys_info)*	F41	*SSH_Client_Version*
F7	*Keyword_useradd -[suffix] (Set_account)*	F24	*Keyword_free (Get_sys_info)*	F42	*Username*
F8	*Keyword_./ [file] (Execution_file)*	F25	*Keyword_lscpu (Get_sys_info)*	F43	*Password*
F9	*Keyword_sh [file] (Execution_file)*	F26	*Keyword_nproc (Get_sys_info)*	F44	*Duration*
F10	*Keyword_./ [file] (Execution_file)*	F27	*Keyword_uptime (Get_sys_info)*	F45	*Received_Size (AVG.)*
F11	*Keyword_perl [file] (Execution_file)*	F28	*Keyword_wget (Network_connect)*	F46	*File*
F12	*Keyword_python [file] (Execution_file)*	F29	*Keyword_tftp (Network_connect)*	F47	*Continent_Code*
F13	*Keyword_/bin/ [file] (Execution_file)*	F30	*Keyword_scp (Network_connect)*	F48	*Country_Name*
F14	*Keyword_chmod [file] (Set_permissions)*	F31	*Keyword_ping (Network_connect)*	F49	*Region_Name*
F15	*Keyword_sudo su (Set_permissions)*	F32	*Keyword_kill (Shutdown_action)*	F50	*City_Name*
F16	*Keyword_rm [file] (Delect_record)*	F33	*Keyword_reboot (Shutdown_action)*	F51	*Longitude*
F17	*Keyword_history -[suffix] (Delect_record)*	F34	*Count_base64*	F52	*Latitude*

5.4 Model Development

Multiple machine learning algorithms are implemented and compared, including:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Deep Learning Models (ANN/LSTM)

These models analyze network traffic and classify it as normal or malicious.

5.5 Model Evaluation

The performance of each model is evaluated using the following metrics:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix
- ROC Curve

These metrics help determine which algorithm performs best for intrusion detection.

6. System Implementation

The implementation of the proposed system is carried out using Python and various machine learning libraries. Tools and technologies used include:

- Python Programming Language
- Jupyter Notebook
- Pandas and NumPy for data processing
- Scikit-learn for machine learning algorithms
- TensorFlow or Keras for deep learning models

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52020

- Matplotlib and Seaborn for data visualization

The system processes network traffic data, applies machine learning models, and predicts whether the traffic is normal or malicious.

7. Expected Results

The AI-powered intrusion detection system is expected to achieve higher accuracy and better detection rates compared to traditional systems. Machine learning algorithms such as Random Forest and Deep Learning models are expected to provide strong performance in identifying complex attack patterns.

Example result comparison:

Algorithm	**Expected Accuracy**
Logistic Regression	88%
Decision Tree	91%
Random Forest	95%
SVM	93%
LSTM Neural Network	96%

The results will demonstrate the effectiveness of AI-based intrusion detection systems.

8. Advantages of the Proposed System

The proposed AI-powered intrusion detection system offers several advantages:

- Detects both known and unknown cyber attacks
- Reduces false positive rates
- Improves detection accuracy
- Supports real-time network monitoring
- Adapts to evolving cybersecurity threats

These benefits make AI-based intrusion detection systems more reliable for modern networks.

9. Conclusion

Cybersecurity has become a critical concern in the modern digital world. Traditional intrusion detection systems are no longer sufficient to handle the increasing complexity of cyber attacks. Artificial intelligence and machine learning technologies provide a powerful solution for improving intrusion detection capabilities. This research proposed an AI-powered Network Intrusion Detection System that uses machine learning algorithms to analyze network traffic and detect malicious activities. By using datasets such as NSL-KDD and CICIDS2017, the system can be trained to recognize patterns associated with network attacks. The proposed system demonstrates how AI can enhance network security by providing intelligent, adaptive, and efficient intrusion detection mechanisms. Future work may focus on integrating deep learning models and implementing real-time detection in large-scale enterprise networks.

References

[1] Profit From Tech. The Ultimate List of Internet of Things Statistics for 2021. Accessed: Mar. 7, 2021. [Online]. Available: https://www. profitfromtech.com/internet-of-things-statistics/

[2] Venafi. Secure Shell (SSH) Security, Vulnerabilities and Exploitation. Accessed: Apr. 19, 2022. [Online]. Available: https://www.venafi.com/ education-center/ssh/security-and-vulnerabilities/

[3] Fraunhofer. Kaiji (Malware Family). Accessed: Apr. 19, 2022. [Online]. Available: https://malpedia.caad.fkie.fraunhofer.de/details/elf.kaiji

[4] D. Fraunholz, M. Zimmermann, A. Hafner, and H. D. Schotten, ‘‘Data mining in long-term honeypot data,’’ in Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW), Nov. 2017, pp. 649–656.

[5] A. Kyriakou and N. Sklavos, ‘‘Container-based honeypot deployment for the analysis of malicious activity,’’ in Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS), Oct. 2018, pp. 1–4.

[6] S. Kumar, B. Janet, and R. Eswari, ‘‘Multi platform honeypot for gen eration of cyber threat intelligence,’’ in Proc. 9th IEEE Int. Conf. Adv. Comput., Dec. 2019, pp. 25–29.

[7] J. M. Pittman, K. Hoffpauir, and N. Markle, ‘‘Primer—A tool for testing honeypot measures of effectiveness,’’ 2020, arXiv:2011.00582.

[8] E. Kheirkhah, S. M. P. Amin, H. A. J. Sistani, and H. Acharya, ‘‘An exper imental study of SSH attacks by using honeypot decoys,’’ Indian J. Sci. Technol., vol. 6, no. 12, pp. 1–12, Dec. 2013.