International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Detection of SSH Password Guessing Attacks using Classification Algorithms

Golla Giridhara Maanas
Department of CSE PES University Bengaluru, India
giridharmanas123@gmail.com

Neeraj GS
Department of CSE PES University Bengaluru, India
gsneeraj2002@gmail.com

Nithish S
Department of CSE PES University Bengaluru, India
nithishanusri0@gmail.com

Pratham Hegde
Department of CSE PES University Bengaluru, India
prathamhegde8@gmail.com

Gokul Kannan Sadasivam
Department of CSE
PES University Bengaluru, India gokul@pes.edu

Abstract— The usage of SSH protocol has gained popularity among users due to its secure nature in recent times. Nevertheless, the SSH protocol can be susceptible to exploitation by hackers, who can access SSH servers without permission by exploiting vulnerabilities. SSH attacks cannot be completely detected using state-of-the-art security solutions like Firewall, Intrusion Detec- tion Systems, and so on. Malicious SSH traffic is created by malware and contains password guessing attacks. These attacks can result in compromising the security of servers and lead to the theft of private data. We aim to develop a robust and accurate SSH attack detection system that uses classification algorithms that can effectively differentiate between malicious SSH traffic and legitimate SSH traffic. In this paper, we have selected 14 classification algorithms like CNN, LSTM, Logistic regression, Deep Belief Networks, Auto Encoders, and so on. The process involves organising and preparing the data, extracting relevant features, and application of an ensemble learning approach with the selected classification algorithms. XGBoost is employed for model integration. The ensemble model achieves improved accuracy, successfully classifying between legitimate SSH traffic and SSH password guessing attacks.

Index Terms—password guessing attacks, classification algo- rithms, ensemble learning, SSH, and feature engineering.

I. INTRODUCTION

SSH protocol is widely used for securely accessing remote servers and systems. SSH provides encryption and authen- tication mechanisms to ensure confidentiality and integrity. However, SSH faces various security threats such as brute- force attacks, port scanning attacks, and so on. These threats may compromise the security of SSH servers and expose them to unauthorised access, data theft, or malicious commands.

Password guessing is a type of attack where the attacker uses a tool to guess the password of different users. The attacker tries to access the credentials of any legitimate user and then assumes their identity. There are different types of attacks under password guessing attacks like dictionary attacks and brute-force attacks. In SSH brute-force and dictionary attacks, an attacker tries different usernames and passwords repeatedly until they can access the server. These attacks can also be automated with the help of tools like Hydra and John the Ripper. These tools contain commonly used usernames and passwords.

Protecting the SSH servers from these types of attacks involves many methods like using complex passwords which makes guessing a password difficult, login attempts that focus on the specified number of attempts a user can attempt, and two-factor authentication. However, these techniques may not be sufficient or effective in detecting and preventing attacks that exploit the vulnerabilities of SSH.

Algorithms such as CNN, LSTM, Logistic Regression, Deep Belief Network, Auto Encoders, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Ridge Classifier, Deep Feed- Forward Neural Network, KNN, SGD Classifier have been used and XGBoost ensemble technique was used to combine the results of the other machine learning algorithms to attain higher accuracy.

Training a machine learning model that can detect malicious traffic can significantly improve the security of SSH com- munication. Employing machine learning models specifically trained to detect SSH malicious activity could reveal if there are any unusual traffic patterns or if any attacks are being made.

Therefore, there is a need to develop more advanced and robust techniques for the detection of SSH attacks. In this project, we propose an approach to classify whether the SSH traffic is a malicious password guessing attack or legitimate traffic using classification algorithms.

In Section II, a literature survey related to attack classification and machine learning algorithms is done. Section

III has two sub-sections highlighting the data collection and Feature Engineering for creating the dataset. Section IV has three sub-sections explaining the training of the model and the evaluation. In Section V, the results obtained are discussed. Section VI consists of the conclusion for the paper.

II. Related Work

S. S. Panwar, Y. P. Raiwani, and L. S. Panwar [1] used the dataset CIC-IDS 2017 and used eight classification algorithms and analysed how they perform on this dataset. The dataset has different types of attacks divided into five days. They have done pre-processing steps like data cleaning, normalising, and then feature extraction on the dataset before training it with the models. They have taken files such that four files have binary labels and three have multi-class labels. Then the performance is checked using metrics like accuracy. They also found that when performing data-preprocessing and feature selection these performance metrics are increasing thus proving that these steps are necessary. Random Forest gave the highest accuracy of 99.98%.

Junwon Lee and Heejo [2] addressed security threats from Advanced Persistent Threats (APTs) targeting institutions, focusing on challenges related to confirming IP addresses due to Infrastructure as a Service (IaaS) usage. Their study introduced synthetic training samples, utilising packet-to- session conversion and a deep generative model, featuring the Average Inter-Packet Arrival Time. The WGAN-GP algorithm was employed for sample creation, filtered using softmax and generator loss. Combined with the original dataset, these samples trained a Random Forest based model for SSH communication detection, evaluated on the DARPA-99 dataset. Results showcased the efficacy of the synthetic dataset, significantly improving recall by 11.9% and precision by 50.1% compared to prior work. The enhanced model, using generated samples, further improved precision by 13.5%. However, the study underscores the nuanced process of adjusting synthetic datasets for real-world SSH detection, emphasising considerations like SSH communication ratios and varied sample requirements.

Stephen Kahara Wanjau, Geoffrey Wambugu, and Gabriel Kamau [3] addressed the problem of detecting SSH brute force attacks. The paper claims that these attacks are difficult to detect because they are like legitimate network communications and can bypass some of the security solutions. The paper tells a way to detect these types of attacks using on CNN model. The paper uses a dataset from (CIC-IDS 2018) collected from a university network in six months, which contains both normal SSH communications and SSH brute force attacks. The paper shows that the CNN model performs better compared to the other models in terms of all metrics, getting 94.3% accuracy, 92.5% precision, 97.8% recall, and 91.8% F-measure. The paper also shows that the CNN model has a less false positive rate and a higher true positive rate than the other models, indicating it detects SSH brute force attacks more accurately and reliably and concludes that the CNN model is better for SSH brute force attack detection.

M. D. Hossain, H. Ochiai, F. Doudou, and Y. Kadobayashi

[4] focused on detecting SSH and FTP brute force attacks using the LSTM model. They have also used other machine learning models and compared them to see which algorithm is effective. They have used the CIC-IDS 2017 dataset and in that, they have used two days of data which has attacks like Web Attack-brute force, SQL injection, and many more. For the implementation, they have used

optimisers like RMSprop with a learning rate of 0.0001 and categorical cross entropy as the loss function. They found that LSTM was better at detecting these attacks than other models and also had an accuracy of 99.8%. Therefore, LSTM was successful in classifying the attacks efficiently.

R. Vinayakumar, K. P. Soman, and P. Poornachandran [5] assessed the performance of different types of shallow and deep neural networks and categorised encrypted network traffic, including SSH traffic. Statistical feature sets are derived from flow-based features of SSH traffic, such as Packet Length, Inter-Arrival Time, duration, and so on. This contrasts the performance of different shallow and deep networks, such as Multi-Layer perceptron, CNN, RNN, and so on. They proposed various private and public network traffic for training and testing the networks, such as NIMS, MAWI, NLANR AMP, and so on. Different deep learning networks with various network parameters, structures, and topologies are carried out. They have transformed the SSH traffic feature sets into 1D vectors that become the input for the CNN model. Experiments were conducted for a maximum of 1000 epochs and changed the learning rate between 0.01 and 0.5.

Gokul Kannan Sadasivam, Chittaranjan Hota, and Bhojan Anand [6] have addressed the vulnerability of SSH servers to various attacks, such as Brute Force attacks, Port scanning, and key-related compromises. They implemented a honeynet system on a DELL PowerEdge Server, collecting continuous network traffic with a public IP address. By categorising SSH attacks as severe and not-so-severe, they have utilized machine learning algorithms such as Naive Bayes, Logistic Regression, J48 Decision Tree, Support Vector Machine, OneR, KNN, and PART to distinguish between these categories. They selected features based on domain knowledge and literature survey. The J48 and PART algorithms were effective in classifying SSH attacks, the PART algorithm achieved an overall accuracy of 0.9999 and an F2 score of 0.9420. Similarly, the J48 decision tree algorithm demonstrated excellent performance with an overall accuracy of 0.9999 and a corresponding F2 score of 0.9420.

Laurens Hellemons et al. [7] main focus is to introduce a new system called SSHCure, a flow-based IDS to overcome the packet-based IDS that can’t scale high in current high- speed networks. It provides an efficient algorithm that de- tects currently ongoing brute-force attacks and also identifies compromised attacks. For the implementation, a prototype of SSHCure was integrated as a plugin for NfSen(NetFlow Sensor), a popular network monitoring tool. Two datasets UT2008 and UT2012 were used. The validation results were with 0 false positives and 1 false negative in both the datasets determining the reliability of the system

III. DATASET

A. Data Collection

To train the model and evaluate the results, legitimate SSH traffic and password guessing attack PCAP files are collected from credible online sources like AZSecure-data, CIC-IDS 2019, and ISCX-IDS 2017 for legitimate traffic, and malicious traffic is collected from sources like ISCX-IDS 2012 and CIC-IDS 2017. The legitimate traffic files contain genuine SSH interactions from a normal user, while the malicious files represent real instances of attack scenarios. PCAP files are then split into individual TCP flow PCAP files using the SplitCap tool. SplitCap is a free tool to split PCAP files into smaller files based on criteria like port number, IP address, and so on.

B. Feature Engineering

Essential features necessary for binary classification are extracted from the PCAP files. Each set of features extracted from an individual PCAP file represents a row in the CSV file. The following set of features were extracted from each PCAP file.

1) Average Payload Length of a TCP flow: Legitimate SSH traffic packets may exhibit diverse payload lengths, varying based on the complexity and volume of transmitted commands and outputs. In contrast, SSH password guessing activities often employ techniques such as port scanning to minimise data transfer, resulting in shorter payload lengths to evade detection through a reduced network footprint. SSH password guessing attempts may manipulate payload lengths to blend with normal traffic patterns and further avoid detection mechanisms.

2) Average Time Difference between Two Packets at server in a TCP flow: The average time difference between two packets at the server in regular SSH traffic exhibits a more predictable pattern, reflecting the systematic nature of legitimate interactions. In contrast, during password guessing attacks, the arrival of packets tends to deviate from this regularity, introducing unpredictable time intervals as attackers intentionally disrupt the expected timing patterns to hinder detection mechanisms. *Average Time Difference between Two Packets at client in a TCP flow:* The average time difference between two packets at the client in regular SSH traffic tends to be more extended, reflecting

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

the natural pacing of user-initiated commands and responses. Conversely, in the context of password guessing attacks, attackers often exhibit a sense of urgency, resulting in shorter time intervals between packets as they rapidly attempt to breach security. This deviation from the usual time gaps in legitimate traffic can serve as an indicator for detecting malicious SSH activities.

3) Standard Deviation of Inter Arrival Time of a TCP flow: In legitimate SSH traffic, a relatively even distribution of inter-arrival times is typical, reflecting the consistent nature of legitimate command executions. However, in the case of SSH password guessing activity aimed at avoiding detection, irregular patterns of packet arrivals may emerge as attackers strategically alter timing intervals to mimic normal user behaviour.

4) Total Number of Packets with ACK Flag set in a TCP flow: Legitimate SSH sessions involve a bidirectional flow of data between the client and server. ACK packets acknowledge the receipt of data packets, confirming the normal, two-way communication pattern. In malicious password guessing attacks attackers repeatedly attempt to authenticate without the need for a significant back-and-forth data exchange. The bidirectional flow of data is less compared to legitimate.

5) Total Number of Packets with RST flag set in a TCP flow: In legitimate traffic, the number of packets with the RST flag set is controlled, as it occurs when a connection is about to be terminated which is a normal part of the process. In malicious traffic, especially during password guessing attacks, the number of packets with the RST flag set is higher. This increase is attributed to the multitude of failed login attempts, resulting in a greater occurrence of the RST flag being set.

6) Total Number of Packets with FIN flag set in a TCP flow: This feature counts the FIN packets in a bidirectional SSH session. Legitimate sessions conclude with a regular closure marked by FIN packets. Malicious attempts, especially with repeated authentication often lack substantial bidirectional exchanges and a notable number of FIN packets.

7) Average Packet Size of a TCP flow: Average packet size in a flow is a good feature because in SSH attacks attackers can manipulate the packet sizes to avoid detection, the packet sizes may also be small. Flows with a high average packet size are more likely to be a normal TCP flow, while a flow with a low average packet size is more likely to be an SSH attack.

8) Average Time Interval in a TCP flow: Calculating the time elapsed between the current frame and the previous frame in the same TCP stream. This measures the time interval between consecutive actions within the same session. Legitimate users typically exhibit a consistent and uniform pattern in the time intervals between their actions. Conversely, in the context of malicious attacks, the presence of rapid and constant time intervals between actions may raise suspicion of anomalous behaviour.

9) Time since First Frame in a TCP flow: This feature measures the time elapsed between the first frame and the current frame in an SSH session. Legitimate sessions have consistent intervals, reflecting normal server-client interactions. Malicious activity often introduces irregular time intervals, deviating from typical communication patterns.

10) Time Difference between First packet and Last packet in a TCP flow: In normal SSH Traffic, the time difference between the first and last packet in the TCP flow tends to be relatively consistent. Users establish a connection, perform their tasks, and then close the connection. But in malicious traffic, attackers performing SSH password guessing attacks keep their session durations short. Short session duration is characteristic of these attacks, as they involve rapid, short-lived connections with minimal interaction.

11) Standard Deviation of TTL values in a TCP flow: In legitimate SSH traffic TTL values generally remain stable or follow a consistent pattern because packets in a well-configured network traverse a predictable number of hops to reach their destination. In malicious traffic, TTL values may show greater variability. If the attacker is attempting to conceal the source of their traffic by using multiple proxy servers or relays, the TTL values can become more diverse.

12) Download/Upload ratio of a TCP flow: A high download/upload ratio in SSH traffic could suggest normal behaviour, where the client is receiving more data than it is sending, aligning with common patterns of command execution and output retrieval. Conversely, a low download/upload ratio may indicate potentially malicious activity, such as file uploads or the execution of commands that generate substantial output, signalling a possible attack scenario where the client is actively contributing more to the data exchange.

13) Average Rate of Window Size Growth in a TCP flow: In legitimate SSH traffic, the window size often remains relatively stable during a session. They change gradually as needed. In malicious SSH Traffic, attackers often attempt to blend in with normal traffic patterns to avoid detection. Anomalous or rapid changes in the window size might be indicative of malicious activity.

14) Total number of Packets with URG flag set in a TCP flow: Examining the frequency of URG-flagged packets, this feature observes normal SSH sessions with infrequent, balanced urgent data transfers. Malicious sessions show unexpected or frequent URG packets, indicating abnormal urgency or data manipulation.

15) Average Rate of Sequence Number Increase in a TCP flow: In legitimate SSH traffic, the rate of increase in the sequence numbers tends to follow a relatively stable pattern. They increase consistently and predictably. In malicious SSH Traffic, abnormally fast or erratic increases in sequence numbers can be indicative of malicious activity. Attackers may quickly send multiple login attempts, causing sequence numbers to rise unusually fast.

16) Total number of Packets with PSH flag set in a TCP flow: Focusing on the PSH flag, this feature notes occasional PSH-flagged packets in legitimate SSH sessions for prompt data delivery. Malicious sessions often display a higher occurrence or unusual placement of PSH-flagged packets, suggesting potential data tampering.

TABLE I: Types of Traffic

TCP Flow Type	Number of TCP Flows
Legitimate SSH	7958
SSH password guessing attacks	9291

IV. Proposed Approach

The system needs a powerful processor with multiple cores to handle the heavy calculations in deep learning and other machine learning tasks. It also requires plenty of memory to manage the large amounts of data used during training and testing. Python programming language has been used. Deep learning frameworks like TensorFlow and Keras are used to build and train models. Other helpful tools include Python libraries like NumPy, Pandas, Matplotlib, and Scikit-learn for analysing and visualising data. The system should work with various operating systems like Windows, Linux, or macOS. Using an integrated development environment, such as Jupyter Notebook, helps write and test code.

A. Model Training

Some machine learning algorithms were chosen to evaluate the performance of classification. The algorithms are Long short-term memory, Bi-directional Long short-term memory, Stacked Long Short-term Memory, One Dimensional Convolutional Neural Network, Quadratic Discriminant Analysis, Deep Belief Network, Ridge Classifier, SGD Classifier, DFNN, KNN, Logistic Regression, Linear Discriminant Analysis performs well for a dataset with only numeric attributes.

LSTMs, designed for sequential data, efficiently capture long-term dependencies using memory cells for selective information storage and retrieval. 1D-CNN excels at identifying local patterns and variations in sequential data, making them effective in capturing specific features associated with SSH traffic, such as patterns related to password guessing attacks. Naive Bayes is simple, efficient with large datasets, and also at handling binary features, making BernoulliNB suitable for modelling binary nature aspects in TCP packet characteristics.

SGD is computationally efficient and performs well on large-scale data making it appropriate for the application of classifying SSH password attacks. The logistic regression model has an activation function called sigmoid which squashes the output between 0 and 1, thus helping in binary classification. Linear discriminant analysis model finds a linear combination of features that separates two or more classes. Training an autoencoder on legitimate traffic, the model can learn to reconstruct the normal patterns and the autoencoder can identify deviations from learnt normal patterns. Quadratic discriminant analysis is a statistical classification method that discriminates between different classes based on their statistical properties. This is advantageous for modelling the statistical characteristics of normal and malicious SSH traffic. It can capture non-linear relationships between features, allowing it to notice complex patterns in SSH traffic that may not be adequately represented by linear models.

B. Ensemble Model

An ensemble approach is proposed to enhance classification performance by leveraging diverse machine learning models. Predictions from different models are stacked horizontally to create a matrix. Subsequently, an XGBoost classifier is em- ployed to further enhance classification performance. XGBoost operates by iteratively constructing decision trees, sequentially correcting errors from preceding trees. The XGBoost model, configured with specific parameters like the number of es- timators, maximum depth, and learning rate, systematically refines predictions, yielding an improved and more accurate classification outcome.

C. Model Evaluation

The performance of the models is evaluated using metrics such as accuracy, precision, recall, and F1-Score. Accuracy represents the overall correctness of the model, considering both malicious and legitimate classifications. A higher accuracy shows that the model is more reliable. A higher precision signifies a lower rate of false classification of legitimate instances as malicious. Recall assesses the model's effective- ness in capturing all actual malicious instances. A higher recall implies a lower rate of falsely classifying malicious instances as legitimate. The F1-Score balances precision and recall, comprehensively measuring the model's performance. A higher F1-Score indicates a well-balanced trade-off between precision and recall.

V. RESULTS

The proposed ensemble model was used to classify SSH password guessing attack traffic and legitimate SSH traffic. The accuracy of 94% was achieved using an ensemble of classification algorithms. The obtained accuracy indicates that the model successfully distinguished between legitimate SSH traffic and SSH password guessing attack traffic.

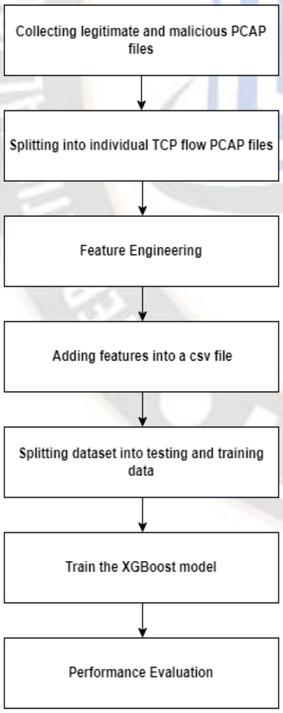

Collecting legitimate and malicious PCAP files
Splitting into individual TCP flow PCAP files
Feature Engineering
Adding features into a csv file
Splitting dataset into testing and training data
Train the XGBoost model
Performance Evaluation

Fig. 1: Model Architecture

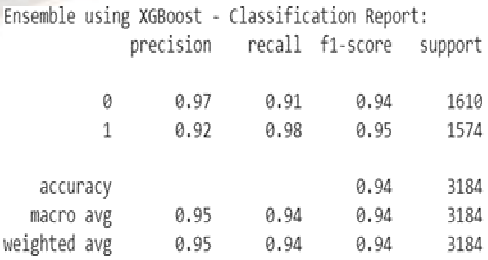
Quadratic discriminant analysis model had the least accu- racy and KNN had the highest among the models. XGBoost builds a stronger predictive model by combining predictions of multiple weak learner models and focuses on correcting errors in previous models.

Table II gives the accuracy metric of each classification model, tested to classify between SSH password guessing attacks and legitimate SSH traffic. Then predictions from each model are combined using the XGBoost technique which combines weak models sequentially, where each new model corrects the error of previous ones. Figure 2 shows the classification report of the ensemble model.

The comparison of the accuracy metrics of each classifi-

TABLE II: Model Performance

Model	**Accuracy(%)**
LSTM	93.18
Bi-LSTM	93.68
Stacked LSTM	93.75
1D-CNN	93.68
Quadratic Discriminant Analysis	77.10
Deep Belief Network	93.56
Ridge Classifier	84.70
SGD Classifier	82.47
Auto Encoders	82.50
DFNN	93.68
KNN	94.03
Logistic Regression	83
Linear Discriminant Analysis	84.70
Naive Bayes	81.50

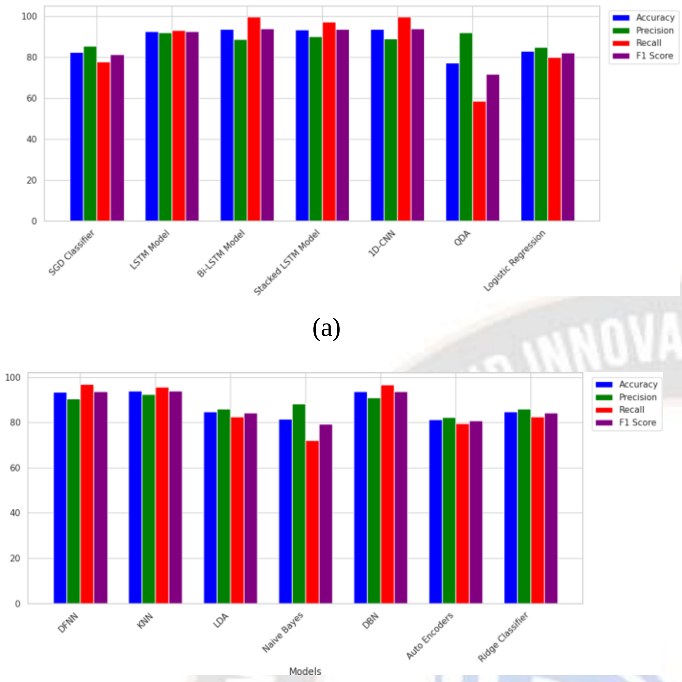


```
Ensemble using XGBoost - Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.91      0.94      1610
           1       0.92      0.98      0.95      1574

    accuracy                           0.94      3184
   macro avg       0.95      0.94      0.94      3184
weighted avg       0.95      0.94      0.94      3184
```

Fig. 2: Classification Report



(a)

(b)

Fig. 3: Performance metrics for each classification algorithmcation model is shown in Figure

3. This shows that accuracy and F1-score are higher for KNN. This graph gives the overall performances so that appropriate conclusions can be made.

VI. CONCLUSION

In conclusion, this project introduces an effective approach for detecting SSH password guessing attacks through machine learning techniques. The methodology involved the collection of SSH legitimate and password guessing attacks packet capture files from various sources. Essential features were extracted from PCAP files, and machine learning models were trained to differentiate between legitimate and malicious activities. The experimentation outcomes reveal our ensemble model's robust performance, achieving an impressive accuracy rate of 94%. This high accuracy highlights the efficacy of the proposed method in accurately detecting and classifying SSH password guessing attacks. The success of this approach holds significant implications for enhancing the security of systems relying on SSH protocols, providing a defence mech- anism against unauthorised access attempts. As cyber threats continue to evolve, the findings of this research contribute to the ongoing efforts to develop advanced and adaptive security measures to detect these kinds of malicious activities.

In the future, working with industries to collect various types of traffic will strengthen the model in dealing with new challenges. More studies could look into making the model, training it, and predicting the attacks in real-time to keep up with changing attack methods. Also, it would be helpful to check if the model can work well in a big, real-world network to see if it's scalable and useful in practice.

REFERENCES

[1] S. S. Panwar, Y. P. Raiwani and L. S. Panwar, "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," 2022 International Conference on Advances in Computing, Communi- cation and Materials (ICACCM), Dehradun, India, 2022, pp. 1-10, doi: 10.1109/ICACCM56405.2022.10009400.

[2] Junwon Lee and Heejo Lee. 2022. Improving SSH detection model using IPA time and WGAN-GP. Comput. Secur. 116, C (May 2022). https://doi.org/10.1016/j.cose.2022.102672.

[3] Wanjau, Stephen & Wambugu, Geoffrey & Kamau, Gabriel. (2021). SSH-Brute Force Attack Detection Model based on Deep Learning. In- ternational Journal of Computer Applications Technology and Research.
10. 42-50. 10.7753/IJCATR1001.1008

[4] M. D. Hossain, H. Ochiai, F. Doudou and Y. Kadobayashi, "SSH and FTP brute-force Attacks Detection in Computer Networks: LSTM and Machine Learning Approaches," 2020 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 2020, pp. 491-497, doi: 10.1109/ICCCS49078.2020.9118459.

[5] R. Vinayakumar, K. P. Soman and P. Poornachandran, "Secure shell (ssh) traffic analysis with flow-based features using shallow and deep networks," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 2026-2032,doi: 10.1109/ICACCI.2017.8126143.

[6] Sadasivam, Gokul Kannan & Hota, Chittaranjan & Bhojan, Anand. (2017). Detection of Severe SSH Attacks Using Honeypot Servers and Machine Learning Techniques. Software Networking. 2017. 79-100. 10.13052/jsn2445-9739.2017.005.

[7] Hellemons, L., Hendriks, L., Hofstede, R., Sperotto, A., Sadre, R., Pras, A. (2012). SSHCure: A Flow-Based SSH Intrusion Detection System. In: Sadre, R., Novotny´, J., Cˇ eleda, P., Waldburger, M., Stiller,
B. (eds) Dependable Networks and Services. AIMS 2012. Lecture Notes in Computer Science, vol 7279. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-30633-4_11