Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2026. Vol. 5. No. 2; e-ISSN: 2808-4519

Brute-Force Attack Detection on Computer Networks Using Artificial Neural Network

Ikhtiar Adli Wicaksono[1]*, Muhammad Iqbal Maulana[2], Bagus Nurrahman[3], Syifa Nur Rakhmah[4], Findi Ayu Sariasih[5], Imam Sutoyo[6]

[1,2,3,5] Informatika, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika
[4,6] Teknologi Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika
15230149@bsi.ac.id[1*], 15230211@bsi.ac.id[2], 15230308@bsi.ac.id[3], syifa.snk@bsi.ac.id[4], findi.fav@bsi.ac.id[5], imam.ity@bsi.ac.id[6]

Abstract

This research aims to develop a brute-force attack detection system on computer networks using the Artificial Neural Network (ANN) algorithm. This security problem is crucial, especially in the banking sector because it can threaten login systems and sensitive customer data. The research methods include data cleansing, feature selection using the Wrapper method, ANN model training, and performance evaluation using datasets from Kaggle which include four classes of network traffic, namely Normal, Brute-force FTP, Brute-force SSH, and Web Attack Brute-force. The test results showed that the ANN model achieved an accuracy of 95%, precision of 91%, and the best performance in the Brute-force FTP class with an accuracy of 98.3%. This system has proven to be effective in detecting brute-force attack patterns and can improve the security of banking networks adaptively. This research broadens the insights of the application of ANN in network security and provides a basis for the development of systems that are more responsive to cyber threats.

Keywords: *Artificial Neural Network; Brute-force Attack; Classification; Machine Learning; Network Traffic*

1. Introduction

The rapid advancement of information technology has brought major changes to the digital security system, especially in the banking sector which relies heavily on computer networks to manage customer transactions and data. The risk of cyberattacks is also increasing, one of which is brute-force attacks. This attack is a technique used by attackers to gain unauthorized access by repeatedly trying different username and password combinations until they successfully enter the system. This method can lead to significant losses such as theft of sensitive data and system damage [1]

Research by [2] in the journal *JIIP (Scientific Journal of Education Sciences)* entitled *"Consumer Protection in the Financial Services Sector in the Case of Ransomware Cyber Attacks Affecting Banks"* explains that ransomware attacks on Bank Syariah Indonesia (BSI) have a major impact on national financial services and reduce the level of public trust. The case shows that the digital security system in Indonesia's banking sector still holds loopholes that can be exploited by cybercriminals.

Another form of threat that also often occurs is *brute-force attacks*, which are attempts to gain illegal access to the system by repeatedly trying various password combinations [3]. Such attacks have the potential to cause significant financial losses and operational disruption to the banking sector, with losses worth millions of dollars due to unauthorized access and service interruptions [4]. In the context of banking, brute-force attacks are a serious threat because they can target customer login systems, internal applications, and authentication servers. If not detected early, these attacks can lead to sensitive data leaks, financial losses, and lower public trust in financial institutions. Therefore, a detection system is needed that is able to recognize attack patterns quickly and accurately.

Previous research emphasizes the effectiveness of machine learning-based approaches in detecting cyberattacks. [5] uses Deep Neural Network (DNN) to detect brute-force attacks on FTP and SSH protocols, with an accuracy of up to 99.9%. [6] shows that DNN is able to recognize SYN Flood DDoS attacks on IoT networks with 99.36% accuracy. In addition,[7] it shows that in many cases deep learning models are able to produce better accuracy and adaptability than traditional machine learning methods in intrusion detection systems. This supports the use of artificial neural network (ANN) algorithms as an effective approach to strengthen security systems and automatically recognize complex attack patterns.

Based on these studies, one of the main problems faced by banking security systems is the difficulty of detecting *brute-force* attacks quickly and accurately, as the attacker's login patterns often resemble normal user activity. In addition, conventional security systems are still static and have not been able to adapt to the ever-evolving variety of attack patterns. To address these problems, this study offers the use of *Artificial Neural Network (ANN)* as a relevant approach to detect *brute-force* attacks on networks in the banking sector. ANN has the ability

to study network activity patterns and distinguish between normal activity and attack activity automatically. With the application of this method, the banking system is expected to be more responsive in recognizing threats and improving the security of customer data.

This research aims to develop a brute-force attack detection system on computer networks by utilizing Artificial Neural Network (ANN) technology that can be applied in banking security systems. Through this research, the researcher hopes to expand his insight and experience in the application of ANN technology in the field of network security. For readers, this study is expected to be able to provide a deeper understanding of the mechanism of brute-force attacks and their detection methods using artificial intelligence. Meanwhile, for the public and the banking industry, the results of this research are expected to be a reference in the development of a more intelligent, efficient, and adaptive security system against various cyber threats that continue to develop.

The scope of this research is focused on the development and testing of brute-force attack detection systems on computer networks using artificial neural networks (ANN). The dataset used is taken from Kaggle and is the result of extracting network traffic capture files with the pcap extension that has been converted to csv format using the cicflowmeter tool. This research is limited to designing a web-based application that is able to receive pcap file input from the user, process the file on the back-end side with the help of a cicflowmeter, and classify the results into four classes, namely: Normal, Brute-force FTP, Brute-force SSH, and Web Attack Brute-force. The classification process is carried out after the data is processed and the features required by the ANN model are mapped, so that the developed system focuses on automating the process of extraction, processing, and detection of brute-force attacks based on network traffic data available in the form of pcap files. The final result of the system is in the form of the output of the classification of the four classes which is displayed to the user through the web application interface.

2. Research Methods

This research uses a software development method that considers the complexity and dynamics in the creation of a brute-force attack detection system based on the Artificial Neural Network (ANN). Therefore, the Agile kanban development methodology was chosen because it is able to provide a responsive, adaptive, and efficient process in dealing with changes during software development. The Agile Kanban method focuses on visualizing workflows and completing tasks continuously (continuous delivery).[8] explained that *the Agile* method was developed to improve collaboration and flexibility in software development. Therefore, the application of Agile Kanban in this study helps the process of designing and testing an ANN-based *brute-force* attack detection system to be more efficient and adaptable to changing needs. The Agile Kanban stages for machine learning-based applications generally include a series of visual, measurable, and adaptive steps focused on workflow management and continuous improvement. Each of these stages can be adjusted to the needs of machine learning application development, from the initial stage to the deployment of the model into the application. The stages are as follows:

1. Planning
   At this stage, all activities to be carried out are recorded and planned. Includes identifying problems, determining research objectives, and the need for Artificial Neural Network datasets and models.
2. Development
   This stage contains the activities that are being worked on. It includes the design of the Artificial Neural Network architecture, *the data preprocessing* process, the model training (*training*), and the creation of the system interface.
3. Testing
   Once development is complete, testing is carried out to ensure the model and system are working properly. The results were tested using test data and evaluation metrics (accuracy, precision, recall, confusion matrix).
4. Reviews
   This stage evaluates the results of the work to find out if it is in accordance with the target. If not, improvements are made to the model or system.
5. Deployment
   The final stage where the system is ready to use. Trained Artificial Neural Network models are implemented into the detection system.

3. Results and discussion

3.1. Planning

Before the network-traffic-dataset-for-known-unknown-attacks *dataset* can be used for machine learning algorithm analysis and training, this dataset needs to be cleaned. After cleaning, the author analyzed the *network-traffic-dataset-for-known-unknown-attacks* dataset by classifying traffic into four classes, namely *Normal, Brute-force FTP, Brute-force SSH, and Web Attack Brute-force*. The table below shows the number and percentage of traffic on *the network-traffic-dataset-for-known-unknown-attacks* dataset by attack type according to the attack label:

Table 1: Target Class Percentage

Label	Quantity	Percentage %
Web Attack - Brute Force	1420	35.5
Brute-force FTP	865	21.62
NORMAL	861	21.52
Brute-force SSH	854	21.35

This study used *a network-traffic-dataset-for-known-unknown-attacks* dataset with a ratio of 80:20, which is 80% of the dataset for training and the remaining 20% for testing.

3.2. Development

In this study, *the Activity Diagram* was compiled at the development stage in the Agile Kanban method, which is when the system design process is carried out. This diagram serves to describe the flow of system activities starting from uploading pcap files, extracting pcap files into csv form on the backend side, classifying with selected features, until the system displays the classification results to web pages. Here's the logical flow of the program:

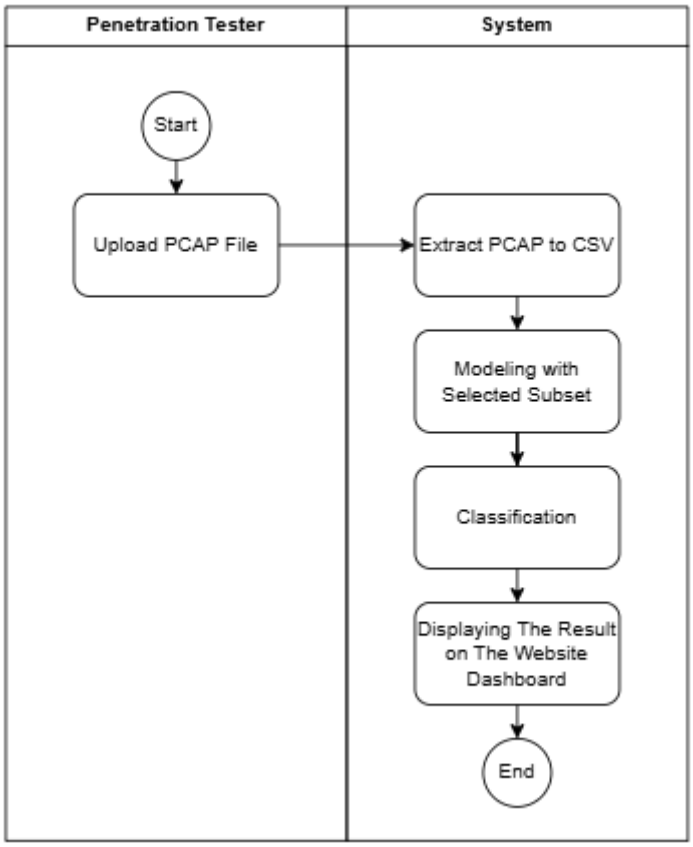

Fig. 1: Activity Diagram

This study uses the Feature Selection Method in the form of *Wrapper Methods. Feature selection is an important part of optimizing the performance of the classifier*[9]. The wrapper method evaluates subsets based on the performance of classification models such as Naïve Bayes (NB), Support Vector Machine (SVM), and Artificial Neural Network (ANN). Subset creation is done in a similar way to the search strategy-dependent filter method, and evaluation is repeated for each subset. The wrapper method is usually slower than the filter method to find a good subset. Practically, we can combine any search technique and modeling algorithm to use as a wrapper[10]. The steps taken by the author include the following:

The initial selection of a subset of features can begin by considering all available features or a specific subset, such as destination ports, stream duration, number of packets, packet speed, and the average size of the packets that are candidate features. Furthermore, the feature search method is used to find the best feature subsets by evaluating each subset using a classifier based on performance metrics such as accuracy. This process is repeated iteratively until a subset of optimal features is found that results in maximum performance or insignificant changes. Finally, a subset of the best features obtained were used in the training and testing of the final model, which in this study resulted in an ANN model with an accuracy of about 95% and was able to classify each attack class well.

The following are the features resulting from the feature selection experiment:

Table 2: Hasil Feature Selection

Column Name	Description	Meaning and Relevance
dst_port	*Destination Port*	The destination port number on the network connection (e.g. 80 for HTTP, 21 for FTP, 22 for SSH). Can indicate the type of service accessed. Certain ports are often the target of attacks (e.g. 22 → brute-force SSH, 21 → FTP).
flow_duration	*Flow Duration (µs)*	The total length of time of a single communication flow, from the first packet to the last. A very short flow can indicate *a scan attack*, while a long one can indicate *data transfer*.

tot_fwd_pkts	*Total Forward Packets*	The number of packets sent from the source to the destination during the communication session. A high value could indicate a large upload activity or a DoS attack from the sender's side.
tot_bwd_pkts	*Total Backward Packets*	The number of packets sent back from the destination to the source. The tot_fwd_pkts/tot_bwd_pkts ratio can be used to detect anomalies in communication (e.g. unbalanced responses).
flow_byts_s	*Flow Bytes per Second*	The data flow rate in bytes per second. Describe the data transfer rate per flow. Extreme values (very high or very low) can be an indicator of an attack.
flow_pkts_s	*Flow Packets per Second*	Packet delivery rate per second. Flow with a small but fast packet can signal a *flooding* (DoS/DDoS) attack.
pkt_size_avg	*Average Packet Size*	The average size of the packet on the communication stream. This value can help distinguish between normal communication (e.g., web browsing) and suspicious activity (e.g., brute-force attacks resulting in small but large packets).

3.3 Testing

A subset of features resulting from the previous Feature Selection stage was used in the training and testing process to evaluate model performance. The results are as follows:

```
================================================================
        Artificial Neural Network Classifier Final Testing
================================================================
Accuracy: 0.9525

Classification Report:
                           precision    recall  f1-score   support

         Brute-force FTP       0.98      0.98      0.98       173
         Brute-force SSH       0.90      0.95      0.92       171
                  NORMAL       0.95      0.93      0.94       172
Web Attack - Brute Force       0.97      0.95      0.96       284

                accuracy                           0.95       800
               macro avg       0.95      0.95      0.95       800
            weighted avg       0.95      0.95      0.95       800


Average Precision Score: 0.9181
```

Fig. 2: Classification report

From the results of the testing, it can be seen that the Artificial Neural Network model has a fairly satisfactory average precision score of 91%. Then the author performed a performance measure using confusion matrix, the following results were obtained:

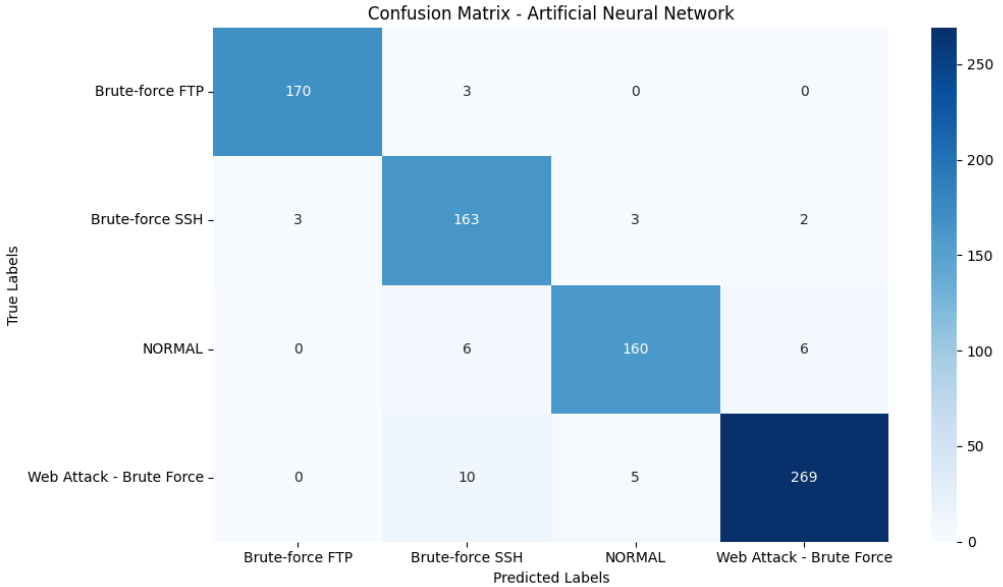

Fig. 3: Confusion Matrix

The results of the confusion matrix show that the Artificial Neural Network (ANN) model has excellent classification performance in distinguishing four types of network traffic, namely *Brute-force FTP*, *Brute-force SSH*, *NORMAL*, and *Web Attack - Brute Force*. Based on the values on the main diagonal, it can be seen that most of the data was correctly predicted by the model. The *Brute-force FTP* class obtained the best results with 170 correct predictions out of a total of 173 data, or about 98.3% accuracy. Only a small portion of the data of this class is incorrectly classified as *Brute-force SSH,* which is likely because the attack patterns of the two have similar characteristics. Meanwhile, *the Brute-force SSH* class also performed well with 163 correct predictions from 171 data (about 95.3%), although there were still some misclassifications to the *FTP*, *NORMAL,* and *Web Attack* classes. This indicates that the traffic patterns of SSH attacks sometimes have similarities to other attacks or even to normal traffic. For the *NORMAL class*, the model was able to correctly recognize 160 of the

172 data, resulting in an accuracy of approximately 93%. Some errors occur because some normal traffic is predicted to be an attack, especially *SSH* and *Web Attack - Brute Force*, which indicates that there is a normal traffic pattern that statistically resembles a light attack.

The Web Attack - Brute Force class itself also has quite high performance with 269 correct predictions from 284 data, or about 94.7% accuracy. Misclassification in this class mostly leads to *SSH* and *NORMAL,* which indicates that some attacks on the web have a connection pattern similar to SSH activity or regular traffic. Overall, this ANN model has a strong ability to recognize various network attack patterns with an overall accuracy rate in the range of 95%. The misclassification that occurs is largely due to the similarity in patterns between the types of brute-force attacks and normal activity. To improve model performance, more specific features can be added to the protocol characteristics and model parameter tuning to learn the differences between classes more optimally.

3.4. Deployment

The deployment at this stage focuses on integrating machine learning models that have been developed and stored in .pkl format (pickle/joblib) into Flask-based web applications, so that they can be accessed and used for classification by users through the web interface. Here's what the website interface looks like:

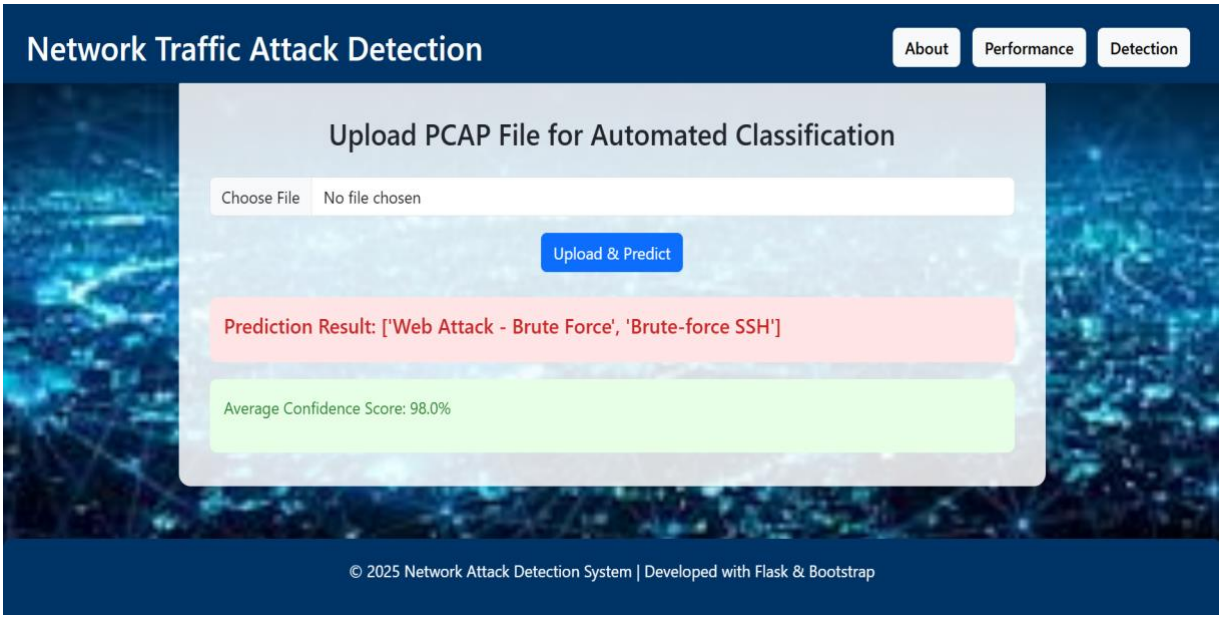

Fig. 4: User Interface

4. Conclusion

This research successfully developed a brute-force attack detection system on network computers using the Artificial Neural Network (ANN) algorithm, with a dataset consisting of 4 classes of network traffic: Normal, Brute-force FTP, Brute-force SSH, and Brute-force Web Attack. The developed system achieved an overall accuracy value of 95% and a precision of 91%, with the best performance in the detection of the Brute-force FTP class of 98.3%. This shows that ANN is already effective in identifying normal activity and attacks, and shows the ability to adapt to dynamic attack patterns, and is much better than conventional methods. However, in this study, there are still some errors in terms of grouping caused by closer normal attack and activity patterns, which can be improved and developed better in future studies. For this research, we have already implemented it in the real world and this could be a handle for the development of more defensive networks, particularly in the banking sector. For this research, it is appropriate for us to implement it in the real world and this can be a handle for the development of a more defensive network, especially in the banking sector.

As a suggestion for further research, it is recommended to explore the use of other methods other than ANN, such as Convolutional Neural Network (CNN), Support Vector Machine (SVM), or Random Forest that have the potential to improve the accuracy of brute-force attack detection. In addition, the development of hybrid methods, for example combining ANN with other machine learning methods such as ensemble learning or deep learning, could be the next step to improve the accuracy and resilience of systems from attack variations. Implementing parameter tuning and optimizing the model architecture in more depth can also improve detection performance. Finally, integration with existing network security systems will add practical value to these systems and accelerate the response to cyber threats in the banking sector.

References

[1] Ardiansyah, A. A. M. Suradi, and W. Saputra, “Strategi Keamanan Router MikroTik: Deteksi dan Mitigasi Serangan Brute Force Berbasis Scripting,” *JUKI J. Komput. dan Inform.*, vol. 7, pp. 12–19, 2025.

[2] D. Afifah, “Perlindungan Konsumen di Sektor Jasa Keuangan pada Kasus Serangan Siber Ransomware yang Menimpa Perbankan,” vol. 6, no. November, pp. 9318–9323, 2023.

[3] K. Mubarok and M. A. Romli, "Implementation of Rule Based Method in Detecting Brute Force Attacks on Owncloud Implementasi Metode Rule Based dalam Mendeteksi Serangan Brute Force pada Owncloud," vol. 5, no. January, pp. 159–167, 2025.
[4] Abdul-waliyyu Bello, Idris Wonuola, Callistus Obunadike, Anastesia Izundu, and Jacinta Izundu, "Assessing the impact of cybersecurity incidents on financial losses and user exposure in the global financial sector (2015-2024)," *Int. J. Sci. Res. Arch.*, vol. 16, no. 1, pp. 489–504, 2025, doi: 10.30574/ijsra.2025.16.1.2037.
[5] N. Alotibi and M. Alshammari, "Deep Learning-based Intrusion Detection: A Novel Approach for Identifying Brute-Force Attacks on FTP and SSH Protocol," *Int. J. Adv. Comput. Sci. Appl.*, vol. 14, no. 6, pp. 107–111, 2023, doi: 10.14569/IJACSA.2023.0140612.
[6] S. Munawarah and E. A. Winanto, "Deteksi Serangan DDoS SYN Flood Pada Jaringan Internet of Things (IoT) Menggunakan Metode Deep Neural Network (DNN) Jurnal Informatika Dan Rekayasa Komputer (JAKAKOM)," vol. 4, no. April, pp. 982–990, 2024.
[7] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos, "Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study," *Appl. Sci.*, vol. 15, no. 4, pp. 1–19, 2025, doi: 10.3390/app15041903.
[8] N. Abbas, A. M. Gravell, and G. B. Wills, "Historical roots of agile methods: Where did 'Agile thinking' come from?," *Lect. Notes Bus. Inf. Process.*, vol. 9 LNBIP, pp. 94–103, 2008, doi: 10.1007/978-3-540-68255-4_10.
[9] V. Chandani and R. S. Wahono, "Komparasi Algoritma Klasifikasi Machine Learning Dan Feature Selection pada Analisis Sentimen Review Film," *J. Intell. Syst.*, vol. 1, no. 1, pp. 55–59, 2015.
[10] N. Ansari, "A survey on feature selection techniques using evolutionary algorithms," *Iraqi J. Sci.*, vol. 62, no. 8, pp. 2796–2812, 2021, doi: 10.24996/ijs.2021.62.8.32.