Analytics-Based Security System Performance Analysis

Kristine Wau, Wana Yumini and Dedy Hartama

May 21, 2024

ANALYTICS-BASED SECURITY SYSTEM PERFORMANCE ANALYSIS

Kristine Wau[1],Wanayumini[2],Dedy Hartama[3]

[1,2]Magister of Computer Science, Potensi Utama University
JL. KL. Yos Sudarso Km. 6.5 No. 3-A, Medan, Indonesia
[1]kristinewau1@gmail.com
[2]wanayumni@gmail.com
[3]dedyhartama@amiktunasbangsa.ac.id

Abstract— Information is a very important asset in making decisions. Diversity of data is a challenge in itself in network management, monitoring and security. Security analytics is a combination of tools used to identify, protect against, and troubleshoot security events that threaten IT systems using real-time and historical data. In this research, a big data analytical approach was used to process network traffic data. by implementing the Naive Bayes and KNN algorithms, comparing the performances between the two algorithms to produce information with the best level of accuracy. The Naïve Bayes algorithm is an algorithm used for statistical classification which can be used to predict the probability of membership of a class,while the KNN algorithm is a supervised learning algorithm which is used to classify new objects based on nearby objects. The aim is to find out attack patterns on network traffic. In this research, the dataset used is network traffic data. Spark was chosen as the big data analytics framework, with Python programming as the language used. the use of big data analytics in the performance of normal data network security systems or as indicated in the training process and network traffic classification.

Keywords: Classification; Security System; Network Traffic; Naive Bayes algorithm; KNN algorithm.

I. INTRODUCTION

Information security is a very valuable asset for organizations because it is one of the strategic assets for creating business value. Therefore, protecting information security is an absolute problem that requires serious thought at all levels of the organization. Information security includes policies, procedures, processes and activities. aimed at protecting information, the increasing importance of information and data requires a security procedure to safeguard information [1]. An analytical-based security system is an approach to computer security that uses data analysis techniques to detect security attacks and threats. Security analytics involves collecting security data from various sources such as system logs, network data, and other information. This data is then analyzed using algorithms and analytical techniques to identify suspicious patterns and behavior [2]. Monitoring and detecting attacks based on traffic data on computer networks is a complex task. Some of the problems faced involve large traffic volumes, transmission system speed,service developments, developing types of attacks,as well as various data sources and methods for obtaining system data security [3].In research carried out in the analytical academic field by predicting student performance with datasets obtained from public datasets, where big data analysis is operated using Apache Spark Next, the data grouping process uses the k-mean clustering algorithm, part of the machine learning algorithm [4].In the context of network security, the use of a big data framework is focused on the Volume,Veracity and Variety characteristics of big data related to network traffic and attacks[5].Thus security measures can be taken more quickly to protect the system and improve attack classification through training and classification trials with the aim of improving attack classification using various algorithms and machine learning techniques [6].In this research, efforts were made to optimize the potential of the Big Data Analytics framework in analyzing network security systems. In research by comparing the performance of accuracy,recall,precision and f1-score levels of two classification methods using Naïve Bayes and K-Nearest Neighbor.analytics-based network traffic data.The dataset used in this research uses network traffic from network traffic data. This dataset is a comprehensive collection of network activity data for studying network infrastructure and traffic.

II. RESEARCH METHODS

In this stage, research was carried out to compare the performance levels of accuracy, recall, precision and f1-score of two classification methods using Naïve Bayes and K-Nearest Neighbor analytically based network traffic data. Figure II.1 shows a general research methodology flow diagram.

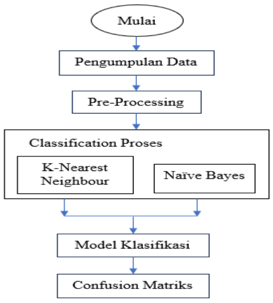


Figure II.1 Research Method Steps

A. Big data analytics

A. Big Data Analytics

Big data analysis involves identifying trends, patterns and correlations in large amounts of unstructured data, with the aim of supporting data-based analysis and decision-making processes. This technique utilizes modern technology to apply various methods, such as clustering and regression, to data sets. The big data analytics process includes the following steps:

1. Structured and Unstructured Data Collection: Data is obtained from various sources, such as network traffic logs, network analysis package applications, and network monitoring applications. Next, the data is saved in a data storage format.
2. Data Pre-Processing: After the data is collected and stored, pre-processing needs to be carried out to filter the data according to needs. This process involves changing data from an unstructured format to a structured format, according to a big data framework.
3. Data Cleaning: Data cleaning refers to the process of identifying data that is inaccurate, incomplete, or unreasonable. After that, the data is modified or deleted to improve data quality. A general framework for data cleansing includes five steps:
   - Define and determine the types of errors.
   - Search for and identify examples of errors
   - Correct errors.
   - Document examples of errors and types of errors.
   - Modify data entry procedures to reduce future errors.

B. Big Data Analytics Framework (Spark)

Apache Spark is a hybrid data processing engine that is powerful, scalable, and supports rapid distribution. As the most active open-source project for Big Data, Spark was developed at UC Berkeley in 2009. Spark provides APIs in Scala, Java,Python, and R.

To process data on a large scale with efficiency, Spark is designed to process large data simultaneously and quickly. Therefore, the Spark architecture is implemented in cluster mode, not just on one machine. The results of processes run by Spark are not saved to disk, but are stored in memory. This all-in-memory capability is a high-performance computing technique for advanced analytics, making Spark 100 times faster than Hadoop.

Additionally,Spark has an ecosystem of libraries that can be used for machine learning, as well as interactive queries that have important implications for productivity.The progressive development of Spark has complemented its ecosystem,providing full support for various libraries. Support for libraries in Spark can be seen in the image below [8].

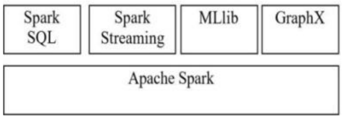


Figure II.2 Apache Spark Ecosystem

C. Data Collection

The data used as primary data in this research is internet traffic usage data taken from internet network data. Meanwhile, secondary data in this research comes from literature studies related to the research title. The data that will be obtained is student performance data as part of academic analytical in the application of big data analytics. In this data preprocessing process, internet traffic usage data is taken using Wireshark software. The data taken is then saved in.csv file format. After obtaining internet traffic usage data, the next step is the preprocessing process with Weka software.

D. Datasets

At this research stage, the dataset is prepared, then the data is divided into two, namely training data and testing data with a percentage of 60% for training data and 40% for testing data. Training data is used to form patterns or models, while testing data is used to test models that have been built. The model used is classification with the Naïve Bayes classifier and K-Nearest Neighbor algorithms which are then evaluated on the results of the classification performance in writing the accuracy level

E. .Pre-Processing

In the preparation stage for data processing, check the data, ensuring that there are no records, empty attributes or the format is appropriate for Python to process it. In the preprocessing process, data on network traffic usage is taken using Wireshark software, the data taken is then saved in file format.csv. After Once the network traffic usage data is obtained, the next step is the preprocessing process with Weka software. After the network traffic data is obtained, the pre-processing stage is carried out for training and testing data for classification using the analytical-based Naïve Bayes and K-Nearest Neighbor algorithms, which are used as a supporting library for machine learning. used the Python library sklearn for data mining.

F. Process Classification

The classification phase consists of two processes, namely training and testing predicted labels. These tasks are implemented using the Python library Sklearn for data mining, data analysis in machine learning using the K-Nearst Neiqboard and Naïve Bayes methods. The training and classification process of network data is carried out by first classifying the data into attack classes or not (attack data or normal data). Classification classes are then used as additional data features when classifying data into attack categories

G. Classification Model

After the data has gone through the classification process from the two models, a data testing process is carried out using the Naive Bayes and K-Nearest Neighbor algorithms. In the final stage of the modeling stage, the model is tested with unseen data. The magical data used at this stage is the result of a test set of split data (40%). Testing is carried out to assess how the model represents the data and how well it will perform in the future, then from the classification of each method model the accuracy level results are analyzed.

H. Confusion Matrix

At this stage we will test whether the performance of the prediction results is effective so that it can be used as a model recommendation for use. one way to describe the performance of a classification model is the number of instances classified correctly and incorrectly. These values are represented in a matrix which is a tabulated visualization of the performance of the supervised learning algorithm. The rows represent the number of instances in the actual class while the columns represent the number of instances in the predictive class. The confusion matrix testing method can produce calculations with 4 outputs (Marchaletal., 2014).

1. Accuracy: Is the percentage of data that is classified correctly to the total amount of data. Accuracy is calculated by the formula A = (TP + TN) / (TP + FP + FN + TN).
2. Precision (P): This is the percentage ratio of the amount of true positive (TP) data divided by the total amount of classified true positive (TP) and false positive (FP) data. Precision is calculated by the formula P = (TP / (TP + FP)) x 100%.
3. Recall(R): Defined as the ratio of % of the number of true positive (TP) data divided by the number of true positive (TP) and false negative (FN) classified records. R=TP/((TP+FN))x100%.
4. F-Measure (F): Defined as the harmonic average of Precision (P) and Recall(R) and represents the balance between them. F(2.P.R)/((P+R))x100%.

III. RESULTS AND DISCUSSION

This chapter explains a series of trials and performance evaluations of research carried out starting from data collection in the form of network activity logs, security attack data or datasets related to analytical-based security systems.

A. Data analysis

At the data analysis stage, in the form of data collection and labeling of data obtained from the UWF-ZeekData22 network traffic dataset, the data is processed into student performance data as part of academic analytical in the application of big data analytical.

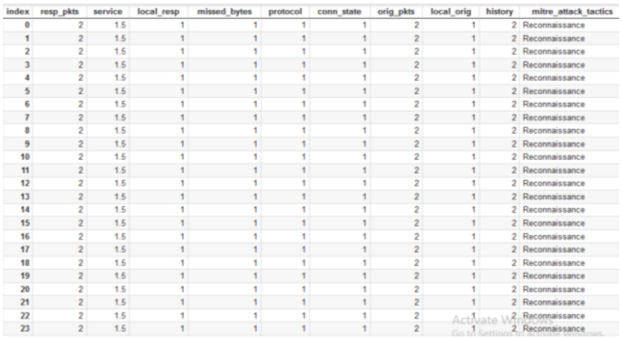
Table I Classification Of Security Systems

No	Name	Description
1	Reconnaissance	Reconnaissance Attack patterns can include a series of actions or behaviors that are typical of specific attacks such as DDoS attacks,brute force attacks, SQL injection attacks and others.
2	None	This pattern includes a traffic activity that is common and expected in a normally operating network.

A. Pre-processing

The preparation stage for data processing is to check the data, ensuring that there are no records, empty attributes or that the format is appropriate. In this data preprocessing process, network traffic usage data is taken using Wireshark software, the data taken is then saved in .csv file format. After obtaining network traffic usage data, the next step is the preprocessing process with Weka software. The steps in data pre-processing are:

1. Download and install WEKA software
2. After installation, go to application explorer.
3. On the preprocess menu, select the Open File tab.
4. Select file.csv



index	resp_pkts	service	local_resp	missed_bytes	protocol	conn_state	orig_pkts	local_orig	history	mitre_attack_tactics
0	2	1.5	1	1	1	1	2	1	2	Reconnaissance
1	2	1.5	1	1	1	1	2	1	2	Reconnaissance
2	2	1.5	1	1	1	1	2	1	2	Reconnaissance
3	2	1.5	1	1	1	1	2	1	2	Reconnaissance
4	2	1.5	1	1	1	1	2	1	2	Reconnaissance
5	2	1.5	1	1	1	1	2	1	2	Reconnaissance
6	2	1.5	1	1	1	1	2	1	2	Reconnaissance
7	2	1.5	1	1	1	1	2	1	2	Reconnaissance
8	2	1.5	1	1	1	1	2	1	2	Reconnaissance
9	2	1.5	1	1	1	1	2	1	2	Reconnaissance
10	2	1.5	1	1	1	1	2	1	2	Reconnaissance
11	2	1.5	1	1	1	1	2	1	2	Reconnaissance
12	2	1.5	1	1	1	1	2	1	2	Reconnaissance
13	2	1.5	1	1	1	1	2	1	2	Reconnaissance
14	2	1.5	1	1	1	1	2	1	2	Reconnaissance
15	2	1.5	1	1	1	1	2	1	2	Reconnaissance
16	2	1.5	1	1	1	1	2	1	2	Reconnaissance
17	2	1.5	1	1	1	1	2	1	2	Reconnaissance
18	2	1.5	1	1	1	1	2	1	2	Reconnaissance
19	2	1.5	1	1	1	1	2	1	2	Reconnaissance
20	2	1.5	1	1	1	1	2	1	2	Reconnaissance
21	2	1.5	1	1	1	1	2	1	2	Reconnaissance
22	2	1.5	1	1	1	1	2	1	2	Reconnaissance
23	2	1.5	1	1	1	1	2	1	2	Reconnaissance

Figure III.1 Network Traffic Result Data processed in WEKA

The network traffic classification process of implementing the Naive Bayes and KNN algorithms in analytical libraries with Python programming on network traffic involves collecting network traffic data which includes normal and anomalous attack patterns. After pre-processing the Naive Bayes and KNN models are trained and tested using a cleaned dataset. This inputs data into an algorithm to recognize attack patterns and normal behavior in the form of model evaluation results based on accuracy, precision, recall and F1-score metrics. this gives an idea of how well the model works.

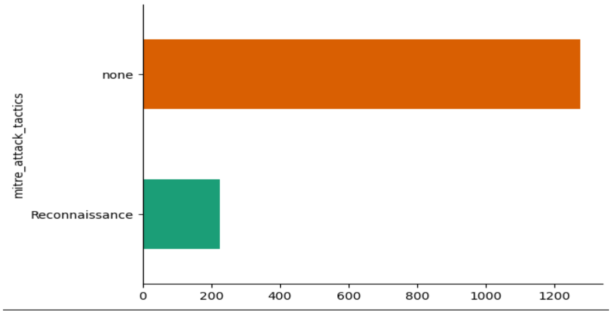
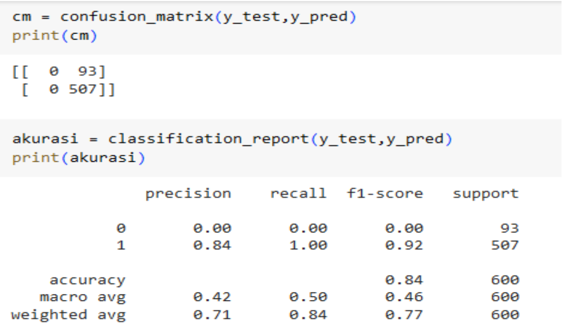


Figure III.2 Normal Attack Patterns and Anomaly Attacks

1) Naive Bayes Classification

After the data passes through data preprocessing, the data is classified using the Naïve classification method. The results of the Naïve Bayes algorithm classification are shown in Figure III.1, which is the result of 60% training data which is 900 while 40% testing data is 600 with a model accuracy level of 84%.



```
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[  0  93]
 [  0 507]]
```

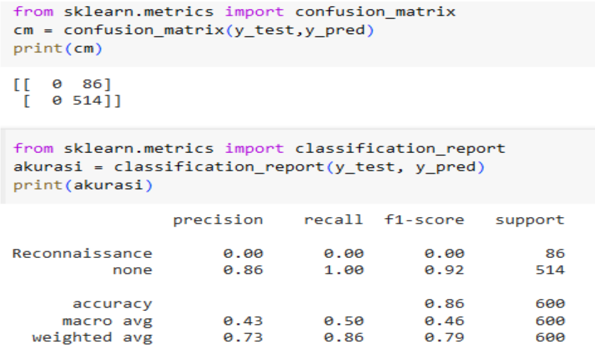
```
akurasi = classification_report(y_test,y_pred)
print(akurasi)
```

```
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        93
           1       0.84      1.00      0.92       507

    accuracy                           0.84       600
   macro avg       0.42      0.50      0.46       600
weighted avg       0.71      0.84      0.77       600
```

Figure III.3 Naïve Bayes Classification Results

2) K-Nearest Neighbor (KNN) Classification

In data classification using the K-Nearest Neighbor algorithm, the results of the KNN algorithm classification are shown in Figure III.2, the results of 60% training data are 900 and 40% testing data are 600, with a model accuracy level of 86%.



```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[  0  86]
 [  0 514]]
```

```
from sklearn.metrics import classification_report
akurasi = classification_report(y_test, y_pred)
print(akurasi)
```

```
                precision    recall  f1-score   support

Reconnaissance       0.00      0.00      0.00        86
          none       0.86      1.00      0.92       514

      accuracy                           0.86       600
     macro avg       0.43      0.50      0.46       600
  weighted avg       0.73      0.86      0.79       600
```

Figure III.4 K-Nearest Neighbor Classification Results

B. Analytical Approach

Big data analysis involves identifying trends, patterns, and correlations in large amounts of unstructured data to support data-driven analysis and decision-making processes. Apache Spark, as a hybrid data processing engine that is powerful, scalable, and supports fast distribution, is the most active open-source project for Big Data. In this research, analysis of the performance of security systems based on analytics as support in the machine learning analytical library for Python programming is carried out. with the implementation of the Naive Bayes and K-nearest neighbor algorithms.

CONCLUSION

The test results carried out implemented the classification algorithm using Naïve Bayes and K-Nearest Neighbor. network traffic with anomalous attack patterns and normal attacks were successfully carried out. The results obtained on the test system produced a level of accuracy from the KNN algorithm testing data that was better than Naïve Bayes at 86%. Meanwhile, the Naïve Bayes algorithm produced an accuracy level of 84% based on the precision parameter results, recall and F1-Score where KNN tends to be better.

REFERENCE

[1] Casas, P., D'Alconzo, A., Zseby, T., & Mellia, M. (2016). Big-DAMA: Big data analytics for network traffic monitoring and analysis. *LANCOMM 2016 - Proceedings of the 2016 ACM SIGCOMM Workshop on Fostering Latin-American Research in Data Communication Networks, Part of SIGCOMM 2016*, 1–3. https://doi.org/10.1145/2940116.2940117

[2] Gökdemir, A., & Çalhan, A. (2022). Deep learning and machine learning based anomaly detection in internet of things environments. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *37*(4), 1945–1956. https://doi.org/10.17341/gazimmfd.962375

[3] Kanakis, M. E., Khalili, R., & Wang, L. (2022). Machine Learning for Computer Systems and Networking: A Survey. *ACM Computing Surveys*, *55*(4). https://doi.org/10.1145/3523057

[4] Novianto, E., Herman, E., Ujianto, H., Rianto,), Informasi, M. T., Yogyakarta, U. T., Ring, J., Utara, R., Lor, J., & Yogyakarta -Indonesia, D. I. (2023). *Some rights reserved BY-NC-SA 4.0 International License KEAMANAN INFORMASI (INFORMATION SECURITY) PADA APLIKASI SISTEM INFORMASI MANAJEMEN SUMBER DAYA MANUSIA 1)*. *8*(1), 10–15. https://doi.org/10.36341/rabit.vx8i1.2966

[5] Prasetyo Nugroho, F., Wariyanto Abdullah, R., & Wulandari, S. (2019). *KEAMANAN BIG DATA DI ERA DIGITAL DI INDONESIA* (Vol. 5).

[6] Purnomo, R., Priatna, W., & Putra, T. D. (2021). Implementasi Big Data Analytical Untuk Perguruan Tinggi Menggunakan Machine Learning. *Journal of Information and Information Security (JIFORTY)*, *2*(1), 77. https://archive.ics.uci.edu

[7] Wang, L., & Jones, R. (2021). Big Data Analytics in Cyber Security: Network Traffic and Attacks. *Journal of Computer Information Systems*, *61*(5), 410–417. https://doi.org/10.1080/08874417.2019.1688731

[8] Wang, S., Balarezo, J. F., Kandeepan, S., Al-Hourani, A., Chavez, K. G., & Rubinstein, B. (2021). Machine learning in network anomaly detection: A survey. *IEEE Access*, *9*, 152379–152396. https://doi.org/10.1109/ACCESS.2021.3126834

[9] "https://spark.apache.org/."