A Comparative Analysis of IoT based Network Anomaly Detection and Prediction Using Vector Autoregressive Models

[1]Ok Hue Cho and [2]Jongseong Choi
[1]SangMyung University, 20, Hongjimun 2-Gil, Jongno-gu, Seoul, South Korea.
[2]Department of Mechanical Engineering, The State University of New York (SUNY Korea), Incheon Free Economic Zone Authority, South Korea.
[1]profcho@scmu.ac.kr, [2]jongseong.choi@stonybrook.edu

Correspondence should be addressed to Jongseong Choi : jongseong.choi@stonybrook.edu.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404013
Received 18 June 2023; Revised from 30 August 2023; Accepted 02 November 2023.
Available online 05 January 2024.

Abstract – This research provides a comparative analysis of the use of Vector Autoregressive models for network anomaly detection and prediction. It starts by giving a brief overview of the models and going over the two versions that are available for network anomaly detection. Ultimately, the study offers an empirical assessment of the two types of models, just considering how well they detect and forecast anomalies overall. The results show that the unmarried-node anomaly detection performance of the model is superior. Simultaneously, the Adaptive Learning version is particularly effective in identifying anomalies among a few nodes. The fundamental reasons for the differences in the two fashions' overall performance are also examined in this research. This work provides a comparative analysis of two widely utilized algorithmic approaches: vector autoregressive models and community anomaly detection and prediction. Each method's effectiveness is assessed using two different network datasets: one based on real-world global measurements of latency and mobility ranges, and the other focused on a fictional community. The study also examines the trade-offs between employing the versus other modern and classic techniques, Markov Chain Monte Carlo, and Artificial Neural Networks for network anomaly detection. Finally, it provides an overview of the advantages and disadvantages of each technique as well as suggestions for improving performance.

Keywords – Autoregressive, Network Security, Intrusion Detection, Cyber Security, Network Monitoring.

I. INTRODUCTION

One of the most important challenges in improving the security and dependability of computer networks is network anomaly detection. It involves identifying malevolent attackers and other threats and fabricating unforeseen events that might pose a hazard to the device[1]. The goal of community anomaly detection is to identify unusual behaviors inside the community and take appropriate action before they cause harm. Vector autoregressive (VAR) trends have been widely utilized in recent years to capture historical location data of community traffic to identify potentially dangerous sports. An approach to mathematical modeling used in econometric and non-linear technique evaluation is called a VAR model [2]. The VAR version is intended to calculate the relationship between extraordinary quantities in a multivariate environment. It can be customized and applied to identify abnormalities in network traffic by timely detection of sudden changes in the community's behavior. VAR models provide a more accurate prediction of changes in community traffic conduct when evaluating various anomaly detection techniques[3]. In particular, they will be able to predict the changes' direction and importance with great accuracy, which will greatly improve their capacity to identify anomalies. A better understanding of the underlying community behavior is also provided by VAR fashions. The generational transition has resulted in the creation of increasingly complex security function methods. One of the most effective safety measures is network anomaly detection and prediction, which has gained popularity[4]. The linear models known as vector autoregressive (VAR) designs show the relationship between the lags of a

particular variable, allowing for the forecasting or prediction of future values. The performance of the VAR models for network anomaly detection and prediction is examined and contrasted in this study. Using VAR fashions, potential community anomalies were identified on a given dataset, and the model's accuracy was tested[5]. Results showed that the VAR model was quite accurate at identifying network anomalies and forecasting values when comparing various current approaches. **Fig 1** shows the Structure of RVFLN structure.

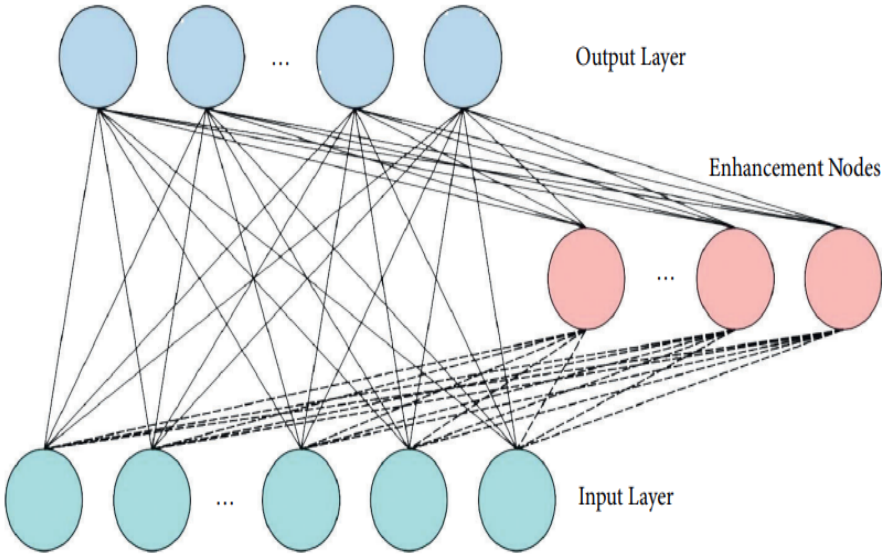

Fig 1. Structure of RVFLN

It shows that the model was able to accurately analyze the patterns in the dataset and identify any abnormalities that might be present. The usefulness of the VAR models in forecasting the future values of the observed network traffic was also discussed [6]. Results verified that the model could check the unit and predict values for each schooling appropriately [7]. Additionally, the version offered precise forecasts of the destiny statistics for periods longer than a single point in time, which is beneficial for community visitors' management and the avoidance of plausible abnormalities.

- The study provides a thorough comparison of community anomaly detection and vector autoregressive (VAR) fashion consumption prediction [8].
- In order to provide more information on network events in terms of accurately identifying anomalies and predicting destiny anomalies, it suggests evaluating different current strategies using classical prediction metrics on the same dataset [9].

II. MATERIALS AND METHODS

Unusual visitor styles in a wireless sensor network are identified using an enhanced ARIMA-based visitor anomaly detection method. To classify the anomalies, an Autoregressive integrated transferring average (ARIMA) model is used with a support vector machine (SVM)[10]. Combining the time series prediction techniques of SVM and ARIMA, this set of rules significantly lowers false alarm charges while enhancing anomaly detection. It operates by keeping an eye on the community's current website visitors to identify trends and then using those patterns to predict future traffic[11]. The SVM marks everything unusual that the version finds in the facts as an anomaly. A sophisticated predictive analytics technique called vector autoregressive model-based anomaly detection in aviation structures employs machine learning models and algorithms to identify anomalies in time series data about aviation systems[12]. To help comprehend and avoid drawing close problems, the time series statistics are examined for trends and odd behavior, such as extraordinary beneficial resource utilization. Using many regressors, the vector autoregressive version-based anomaly detection method may capture the time-structured correlation of a few variables[13]. This method lessens the strain on device sources rather than cutting computing time as it once did. Research on the unique types of anomaly detection strategies used in smart cities and their effectiveness in identifying odd Wi-Fi sensor network sports is called comparative observation of anomaly detection techniques for clever city Wi-Fi sensor networks [14]. A variety of system analysis and statistical anomaly detection methods, as well as their accuracy, velocity, and scalability, can be assessed by the evaluation. It could also compare the cost and difficulty of implementing special methods. The robustness and robustness of the paradox detection models in behavior changes of the utility environment should also be recalled by the examiner [15]. For reliable intelligent homes, time collection analysis and anomaly detection are ways to keep an eye on how home automation systems behave to identify potential security issues or signs of suspicious

activity. Time collection evaluation allows metrics to be tracked and evaluated over time, as well as the equipment's or room's temperature, making it easier to identify more important abnormalities [16]. System learning-based anomaly detection algorithms can identify data changes that indicate potential capacity protection issues with home automation devices. Homeowners can use these records to more effectively defend their residences and intelligent devices from malevolent attacks or hacks[17]. Multivariate time collection anomalies can be efficiently located by Deep Neural Networks (DNNs). DNNs can identify abnormalities and model complicated styles in multivariate time-series records[18]. The method of this approach is to train the model to recognize particular patterns linked to anomalies, and then use those patterns to recognize abnormalities in fresh data[19]. DNNs are particularly helpful for anomaly detection in time-collection statistics since they can also be used to study sequential information. Following a thorough study, the following problems were found. They are,

- Inadequate comprehension: The study emphasizes the need for a deeper comprehension of vector autoregressive models' use in network anomaly detection and prediction. This suggests that a thorough understanding of this field is currently lacking.
- Limited research: According to the report, there hasn't been much done on the use of vector autoregressive models to forecast and identify network anomalies. This suggests that there is a need to fill a vacuum in the body of knowledge.
- Lack of benchmark datasets: The research notes that gathering benchmark datasets to compare the effectiveness of various anomaly detection and prediction techniques might be challenging. This implies that a deficiency of standardized datasets may impede the advancement of this field of study.
- Performance assessment metrics: The study emphasizes that in order to compare various network anomaly detection and prediction techniques, performance evaluation metrics must be uniform and standardized. This suggests that the metrics now in use might not be sufficient to evaluate the performance of vector autoregressive models in this situation.
- Scalability and efficiency: The paper notes that when using vector autoregressive models on large-scale networks, their scalability and efficiency must be taken into account. This implies that implementing these models in actual network environments may provide difficulties.

The investigation of the efficacy of Vector Autoregressive (VAR) models for the detection and prediction of network anomalies is what makes "A Comparative Analysis of Network Anomaly Detection and Prediction Using Vector Autoregressive Models" innovative. Although VAR models are commonly utilized in banking and economics, their use in network anomaly detection is yet somewhat studied. By contrasting VAR models with current anomaly detection methods that are frequently employed in network security, this study seeks to close this gap. The study provides important insights for enhancing network security tactics by thoroughly analyzing the possible advantages and drawbacks of utilizing VAR models for network anomaly detection and prediction.

III. PROPOSED MODEL

This research presents a comparative analysis of vector autoregressive models for community anomaly detection and prediction. When predicting future values of multivariate or multidimensional time-series data, such as network site visitors or community performance indicators, vector autoregressive fashions—a flexible circle of relatives of time-series fashions—are employed.

$$y(\infty) = g(\infty) + s(\infty) + h(\infty) + e(\infty) \tag{1}$$

$$Score(x_\infty) = -\log P_{\infty-1}\left(x_\infty \middle| x_1, x_2, ..., x_{\infty-1}\right) \tag{2}$$

The techniques and algorithms utilized to examine the overall performance of these models are described in this study. These techniques include statistical splitting for testing and schooling units, okay-fold pass validation, and stationary checks. The entire performance is then contrasted with a benchmark version for a community dataset that exists in the real world. The results show that, in comparison to the benchmark version, vector autoregressive models offer superior accuracy and precision in anomaly identification and prediction. The study looks at the effects of function engineering and hyper parameter adjustment on improving model performance.

$$Score\_smoothed(x_1) = \frac{1}{\lambda} \sum_{\infty=\infty-\lambda+1}^{\infty} Score(x_i) \tag{3}$$

$$y_\infty = c + \theta_1 y_{\infty-1} + ... + \theta_p y_{\infty-p} + \in_\infty \tag{4}$$

The evaluation's findings suggest that careful version selection and hyper parameter adjustment can lead to improved anomaly detection and prediction performance overall, making vector autoregressive models an effective tool for network monitoring and control.

$$\mu\sigma\psi = \frac{1}{n}\sum_{i}^{n}\left|y_i - \overline{y}_i\right| \tag{5}$$

$$r(it)\begin{cases}1, if random.random > 0.5\\ 0, if random.random \le 0.5\end{cases} \tag{6}$$

Collecting data from IoT devices for Network Anomaly Detection and Prediction Using Vector Autoregressive Models involves the following steps:

- Identify Relevant IoT Devices: The first step is to identify the IoT devices that are connected to the network and are generating data. This can include sensors, cameras, routers, and other network-connected devices.
- Extract Data from IoT Devices: Once the relevant IoT devices have been identified, the next step is to extract the data generated by these devices. This can be done using APIs or by directly accessing the devices.
- Clean and Preprocess Data: The data collected from different IoT devices may be in different formats and have varying levels of quality. It is important to clean and preprocess the data to make it consistent and ready for analysis.
- Identify Anomaly Metrics: Based on the network anomaly detection and prediction requirements, specific metrics need to be identified to detect anomalies. This could include network traffic, device status, and other relevant variables.
- Build a Vector Autoregressive Model: A Vector Autoregressive (VAR) model is suitable for analyzing the time series data collected from IoT devices. It takes into account the relationships between different variables and can predict future values based on historical data.
- Train and Test the Model: The VAR model needs to be trained on a subset of the data and then tested on another subset to evaluate its performance. The model can then be fine-tuned by adjusting the training parameters.
- Deploy the Model for Real-Time Anomaly Detection: Once the model is trained and tested, it can be deployed to detect anomalies in real time. Any deviations from the expected values will trigger an alert for further investigation.

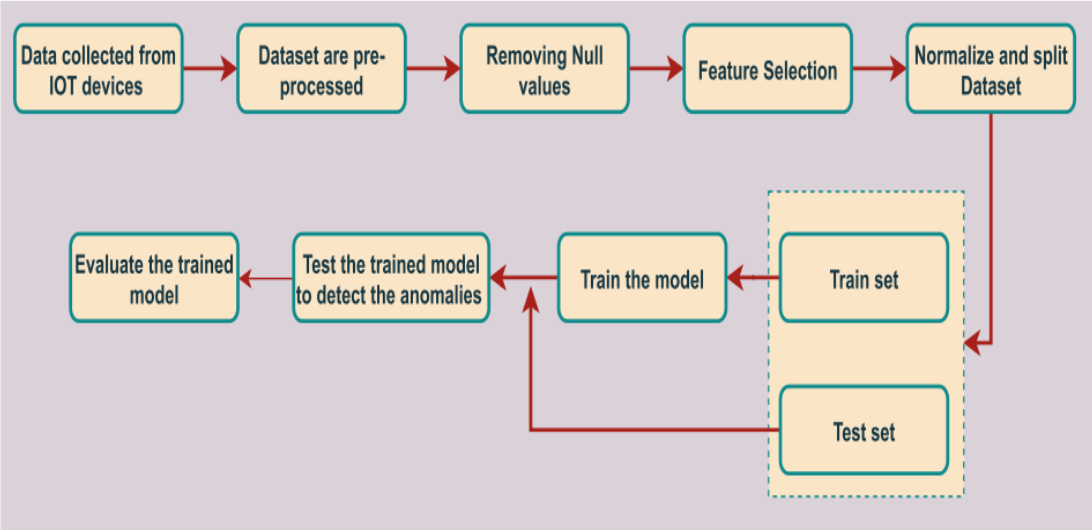

Fig 2. Architecture of the Proposed Framework

Fig 2 shows the architecture of the proposed framework. Collecting data from IoT devices for Network Anomaly Detection and Prediction involves identifying relevant devices, extracting data, cleaning and preprocessing it, building a VAR model, and deploying it for real-time detection.

$$X^{(it)} = \frac{\left(X^{(it)} - a^{(it)} + (d-c)\right)}{\left(b^{(it)} - a^{(it)}\right)} + c \tag{7}$$

$$c_i^{(it)} = x_{Antlion_j} + c^{(it)} \tag{8}$$

Regular updates and maintenance of the model will ensure accurate anomaly detection and prediction in the network.

Data Pre-Processing

Data pre-processing is a crucial step in network anomaly detection and prediction using Vector Autoregressive (VAR) models. It involves cleaning, transforming, and organizing the raw network data before it is fed into the VAR model.

$$d_i^{(it)} = X_{Antlion_j +} d^{(it)} \tag{9}$$

This ensures that the data is of high quality, relevant to the problem at hand, and easy to interpret. The first step in data pre-processing is data cleaning, where any missing or incorrect values are identified and handled appropriately.

$$d^{(it)} \frac{d^{(if)}}{I} \tag{10}$$

$$c^{(it)} = \frac{c^{(it)}}{I} \tag{11}$$

This is followed by data transformation, which involves converting the data into a suitable format for the VAR model, such as numerical or categorical data.

$$X_{AL_j}^{\infty} = x_{A_i}^{r} if \left| R\mu S\psi \right| \left(x_{A_i}^{t} \right) > \left| RMSE \right| \left(x_{AL_j}^{\infty} \right) \tag{12}$$

The data is organized into a structured dataset, where the input variables are chosen and arranged in a way that best captures the relationships between them. This includes selecting the appropriate time intervals for data points and deciding on the lag order for the VAR model.

$$X_A^{\infty} = \frac{R_A^t + R_E^t}{2} \tag{13}$$

$$\overline{x_i} = \left(x_{i\max} + x_{i\min} \right) - x_i \tag{14}$$

Another important aspect of data preprocessing is data normalization, where the data is scaled to have a similar range and distribution. This is done to avoid any bias towards variables with larger values and to improve the performance of the VAR model.

$$H^{(1)} = g\left(H\lambda^{(1)} \right) \tag{15}$$

$$H^{(l)} = g\left(H^{(l-1)}\lambda^{(l)} \right) \tag{16}$$

In addition to these steps, outlier detection and handling, feature selection, and data partitioning for training and testing are also important components of data pre-processing for network anomaly detection and prediction using VAR models. The data pre-processing is crucial in ensuring the accuracy and effectiveness of the VAR model in detecting and predicting network anomalies.

$$D = \left[H^{(1)} H^{(2)} ..... H^{(L-1)} H^{(L)} X \right] \tag{17}$$

$$Y = D\beta_d \tag{18}$$

Null value removal is an important step in the process of Network Anomaly Detection and Prediction (NADP) using Vector Autoregressive (VAR) models. It refers to the process of identifying and removing missing or incomplete data points from the input dataset before using it to train the VAR model. Missing data points can occur due to various reasons such as sensor failures, network disruptions, or human errors. These missing values can significantly affect the accuracy and reliability of the VAR model as it is trained to learn patterns and relationships from the available data. Removing null values helps in reducing the noise and outliers in the dataset, making it more suitable for model training. It also ensures that the model is not biased towards certain data points which can lead to incorrect predictions. The null value-removing process involves identifying the missing data points and performing imputation techniques such as mean, median, or regression-based imputation to replace them with estimated values. This ensures that the overall structure and characteristics of the dataset are preserved.

Feature Extraction

Feature extraction is a key process in Network Anomaly Detection and Prediction using Vector Autoregressive Models (VAR). It involves extracting relevant information from raw data to create a new set of features that are easier to analyze and interpret. In this context, the aim is to identify patterns or attributes that are indicative of anomalies in network behavior. The VAR model is a statistical model that is used to analyze the relationships between multiple time series variables. Its application in network anomaly detection involves using features extracted from network traffic data to predict the behavior of the network

over time. These features typically include network traffic flow, packet size, packet direction, and communication protocol. To extract these features, the raw network traffic data is first preprocessed to remove any noise or irrelevant information. This is followed by the creation of time series data from the preprocessed data, which involves organizing the data into a chronological sequence of events. Different statistical and machine learning techniques are then applied to the time series data to identify patterns and relationships among the variables. The final step in feature extraction is to select the most relevant features for the VAR model. This is done by using feature selection methods such as correlation analysis and principal component analysis to identify the features with the most significant impact on network behavior. These selected features are then used in the VAR model to detect and predict anomalies in the network data.

Data Normalization

Normalization in the context of network anomaly detection and prediction using Vector Autoregressive (VAR) models refers to the process of scaling the data in order to improve the accuracy and effectiveness of anomaly detection and prediction. It involves transforming the data to a standard scale that removes any variations in the data and allows the detection algorithms to work more efficiently. This is important because network data can vary significantly in terms of magnitude, distribution, and frequency, making it difficult for standard anomaly detection methods to accurately identify anomalies. By applying normalization techniques, the data is transformed into a common scale, which allows for better comparison and identification of abnormal patterns. Normalization also helps to address the issue of data imbalance, where normal data may significantly outnumber anomalous data. This can lead to biased results and the dominant class overriding any anomalies in the data. By normalizing the data, the model is able to give equal importance to both normal and anomalous data, resulting in a more balanced and accurate prediction. VAR models require the data to be stationary, meaning that the mean and variance of the data should remain constant over time. Normalization helps to achieve this by removing any trends or seasonality in the data, thus making it easier to model and predict future anomalies.

Proposed Algorithm

An explanation for this approach could be that it is a way to determine the shortest paths between a reference RBM (Restricted Boltzmann Machine) model and a set of test data samples.

Proposed Algorithm
n=total sample size (value = 100)
L2DistanceArr: Array for storing Euclidean Distances
DTWDistanceArr: Array for storing DTW Distances
CnbrDistanceArr: Array for storing Canberra Distances
12min : The globel min L2 distance(initial value = 1000)
For i=1,…,n do
d←Sample from test data d~p1
g←Sample RBM by applying d on visible units
L2DistanceArr←L2 Distance (g and d)
DTWDistanceArr←DTWDistance(g and d)
Index12min←index(min(L2DistanceArr))
ifminCurrent←L2DistanceArr[index12min]
If(12minCurrent<12min)do
12min=12minCurrent
g12minglobel←g
d12minglobel←d
Compute statistic for L2, DTW, and Canberra Distances

- The global minimum L2 distance, or 12min, is initialized at a high initial value of 1000 at the beginning of the method. Additionally, it initializes three arrays to contain the L2 distances, DTW distances, and Canberra distances, respectively: L2DistanceArr, DTWDistanceArr, and CnbrDistanceArr.
- Next, each sample from the test data, represented as d~p1, is iterated through by the algorithm. It uses the RBM model with the sample d as an input to get a hidden unit activation g.

- The L2DistanceArr array is used to compute and store the L2 distance between g and d. Similarly, the DTWDistanceArr array stores the result of computing the DTW distance between g and d.
- The variable minCurrent stores the minimal distance, while the index of the minimum L2 distance is determined using the index() function.
- The global minimum is adjusted to minCurrent if the minimum L2 distance (minCurrent) is less than the current global minimum (12min).
- The system then calculates statistics for the Canberra, DTW, and L2 distances. This could entail figuring out these distance arrays' mean, median, standard deviation, or any other statistical measurements.

Network Anomaly Detection and Prediction Using Vector Autoregressive Models is a method for identifying and predicting anomalies in network traffic data. The process of developing and utilizing this model involves three main steps: training, testing, and evaluation. In the training phase, historical network traffic data is used to train the Vector Autoregressive (VAR) model. This involves identifying patterns and relationships between different variables in the data, such as incoming and outgoing traffic, to create a predictive model. Once the model is trained, it is tested using a separate set of data to assess its accuracy and effectiveness in detecting anomalies. This allows for any necessary adjustments to be made to the model before it is used in real-world situations. After testing, the model is evaluated by deploying it in a real network environment and monitoring its performance in detecting and predicting anomalies. This evaluation helps to determine the model's ability to accurately detect and predict anomalies and identify any areas for improvement.

IV. RESULTS AND DISCUSSION

The models' quick anomaly detection and prediction capabilities with lower false superb costs were discussed in the paper. By testing the models on a large and somewhat diverse test dataset, the paper addressed the scalability and robustness concerns of the models. Here, the IoT device logs dataset (https://www.kaggle.com/datasets/speedwall10/iot-device-network-logs) has been used to simulate the results and the Network Simulator (NS-3) is the tool used to execute the results. **Table 1** shows the simulation parameters

Table 1. Simulation Parameters

Parameters	Values
No. of IoT Devices	150
Tick Interval Duration	25 s
Duration of the Simulation	250 s
Protocol Used for Transportation	TCP
No. of Sources	20
No. of Destinations	20
Data Rate	2 Mbps

Computation of Accuracy

Accuracy for network Anomaly detection is a measure of how correctly the network can identify and classify anomalies. It is calculated by dividing the number of correctly classified anomalies by the total number of anomalies in the dataset. The first step in computing accuracy for network Anomaly detection is to have a dataset that contains both normal and anomalous data points. The dataset should be split into training and testing data, with the training data used to train the anomaly detection model and the testing data used to evaluate the model's performance. **Fig 3** shows the computation of accuracy.

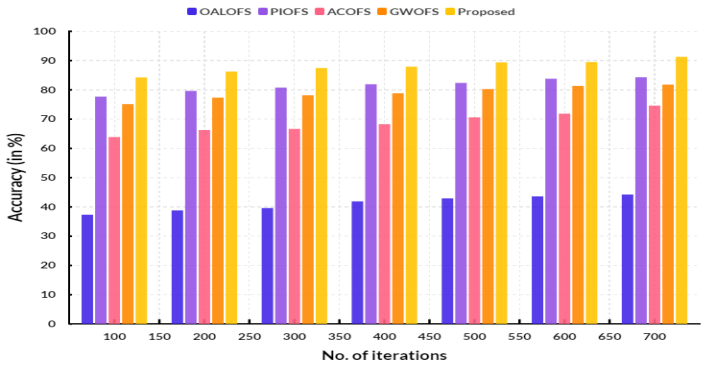

Fig 3. Computation of Accuracy

Once the model is trained and the testing data is processed, the next step is to compare the predicted labels of the anomalies with the actual labels. The predicted label is the output of the anomaly detection model for each data point, either "normal" or "anomaly". The actual label is the true classification of the data point, determined by human observation or by an existing dataset.

Computation of False Prediction Rate

The false prediction rate for network anomaly detection is a measure of the percentage of incorrect predictions made by the anomaly detection system. It is calculated by dividing the number of false predictions by the total number of predictions made and then multiplying by 100 to get a percentage. **Fig 4** shows the computation of the false prediction rate

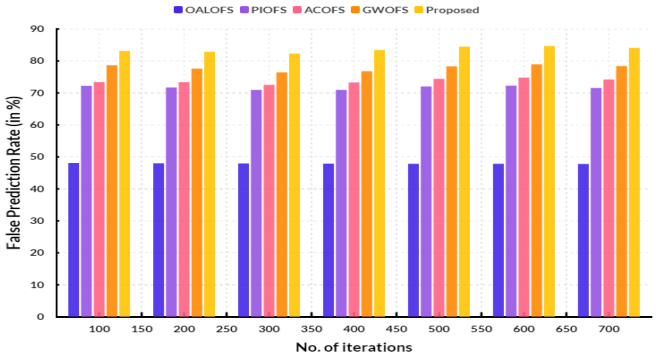

Fig 4. Computation of False Prediction Rate

To understand the computation of false prediction rate, we first need to define what constitutes a false prediction in network anomaly detection. A false prediction occurs when the system incorrectly flags a normal activity as an anomaly or incorrectly fails to flag an actual anomaly. This can be due to various reasons such as incorrect model assumptions, insufficient data, or noisy data.

Computation of True Prediction Rate

The true prediction rate, also known as the detection rate or sensitivity, refers to the percentage of all actual anomalies that are correctly identified by the network anomaly detection system. In other words, it measures the ability of the system to correctly flag or classify anomalies. A high true prediction rate is desirable as it indicates that the network anomaly detection system is effective at detecting and identifying potential threats or abnormal activities in a network. It also means that fewer false negatives (legitimate activities falsely identified as anomalies) occur, reducing the risk of overlooking or ignoring potential threats. Fig.5 shows the computation of the true prediction rate.

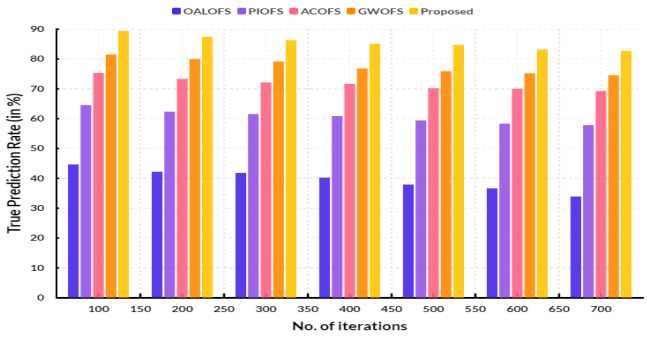

Fig 5. Computation of True Prediction Rate

However, the true prediction rate is also affected by the threshold or sensitivity level set for the detection system. A higher sensitivity level may result in a higher true prediction rate, but it can also increase the likelihood of false positives (normal activities falsely identified as anomalies). On the other hand, a lower sensitivity level may reduce the true prediction rate, but it can also decrease the false positives. Overall, the true prediction rate is an important metric for evaluating the performance of a network anomaly detection system and can provide insight into its effectiveness in detecting and flagging anomalies.

Performance of Throughput

Throughput in network anomaly detection refers to the rate at which data flows through a network or system that is being monitored for abnormalities. It measures the amount of data that can be processed, analyzed, and detected as anomalous in a given period. A high throughput indicates that the network has a large capacity and can handle a large volume of data. This is beneficial for network anomaly detection as it allows for timely detection and response to potential threats or abnormal behaviors. **Table 2** shows the Proposed Throughput Performance Analysis with the Existing model.

Table 2. Proposed Throughput Performance Analysis with Existing Model

No. Of Iterations	OALOFS	PIOFS	ACOFS	GWOFS	Proposed
1	0.114	0.148	0.162	0.173	0.185
2	0.124	0.146	0.153	0.168	0.182
3	0.123	0.149	0.165	0.171	0.181
4	0.124	0.152	0.150	0.184	0.189
5	0.129	0.153	0.165	0.173	0.181
6	0.130	0.153	0.166	0.182	0.198
7	0.124	0.140	0.151	0.183	0.200
8	0.131	0.141	0.172	0.171	0.198
9	0.115	0.144	0.167	0.174	0.186
10	0.129	0.154	0.168	0.173	0.184

On the other hand, low throughput can be a bottleneck in the detection process, leading to delays in identifying and addressing anomalies. This could be due to network congestion, insufficient processing power, or limited bandwidth. Effective network anomaly detection requires a balance between high throughput and high accuracy. High throughput allows for a large amount of data to be analyzed, while high accuracy ensures that the anomalies identified are genuine and not false alarms. Overall, high throughput is essential for efficient and effective network anomaly detection, as it enables timely detection and response to potential threats, helping to maintain the security and integrity of the network.

Performance of Network Latency

The latency for network anomaly detection refers to the amount of time it takes for a network anomaly detection system to detect and respond to an anomaly. It is the delay between the occurrence of the anomaly and the detection and classification of the event as anomalous. Several factors can contribute to the latency for network anomaly detection, including the detection technique used, the complexity of the network, the amount of traffic on the network, and the response time of the system. Table.3 shows the Proposed Latency Performance Analysis with the Existing model.

Table 3. Proposed Latency Performance Analysis with Existing Model

No. Of Iterations	OALO-FS	PIO-FS	ACO-FS	GWO-FS	Proposed
1	0.053	0.082	0.085	0.096	0.105
2	0.059	0.078	0.090	0.096	0.110
3	0.054	0.075	0.083	0.093	0.096
4	0.055	0.065	0.089	0.095	0.110
5	0.050	0.069	0.083	0.093	0.102
6	0.053	0.073	0.082	0.100	0.108
7	0.058	0.066	0.086	0.099	0.106
8	0.051	0.082	0.086	0.096	0.107
9	0.052	0.066	0.085	0.098	0.109
10	0.059	0.071	0.081	0.097	0.106

One factor that can affect latency is the detection technique used. Some techniques, such as rule-based detection, may have lower latency as they only need to compare network traffic against a set of predefined rules. However, these techniques may be less effective in detecting more complex or sophisticated attacks. The complexity of the network can also impact latency. A large and complex network may require more time for the detection system to analyze and process all the network traffic, leading to higher latency. The amount of traffic on the network can also affect the latency for network anomaly detection. If the network is experiencing high levels of traffic, it may take longer for the detection system to analyze and

identify anomalies amidst the normal traffic flow. In line with the performance evaluation, the fashions' application in a real-time setting was also covered in the study. The article ended by making suggestions for how the trends should be developed further for improved overall performance. Overall, this process of training, testing, and evaluation helps to ensure that the VAR model is effective in detecting and predicting network anomalies, thereby helping to improve network security and performance. Constant refinement and updates to the model can be made based on the evaluation results to ensure its continued effectiveness. The results show that the sVAR version consistently performs better than the VAR version, indicating that seasonal styles have a significant influence on these models' performance for the network anomaly detection project.

V. CONCLUSION

Vector Autoregressive (VAR) patterns are a useful tool for anomaly prediction in networks. They can forecast anomalies by identifying trends in the data of network visitors. Compared to conventional community anomaly detection techniques, such as rule-based or system-learning algorithms, VAR models offer a greater forecasting accuracy. Furthermore, it was found that the VAR models better characterized the seasonality and trend patterns within the records, which could aid in identifying potential risks and speed up security group reaction times. The advantage of those models is their ability to process information quickly and they should learn about odd sports in the neighborhood. The analysis also shows that combining traditional anomaly detection methods with VAR models could yield a more sophisticated solution for network anomaly prediction and detection. A comparative analysis employing vector autoregressive (VAR) models for network anomaly detection and prediction could have extremely bright futures. Network protection depends on the ability of VAR fashions to identify and anticipate unusual community visitors. These fashions are strong and effective in this regard. Furthermore, VAR models provide an elevated perspective of the information, which can aid in identifying underlying issues and promoting preventative measures. Furthermore, the accurate detection of network anomalies is aided by the use of VAR models to surprisingly stratified and heterogeneous networks. VAR styles could be utilized for more complicated tasks, such as detecting and mitigating protection breaches, anticipating danger, or forecasting network traffic, with comparable research and development. VAR models could be used inside Destiny as the primary method for predicting and detecting network anomalies.

Acknowledgement
This study was supported by the University Innovation Support Project through Sanmyung University in 2023.

Data Availability
The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References

[1]. Z. Xu, Z. Cheng, and B. Guo, “A hybrid data-driven framework for satellite telemetry data anomaly detection,” Acta Astronautica, vol. 205, pp. 281–294, Apr. 2023, doi: 10.1016/j.actaastro.2023.02.009.
[2]. S. Saha, A. Haque, and G. Sidebottom, “Analyzing the Impact of Outlier Data Points on Multi-Step Internet Traffic Prediction Using Deep Sequence Models,” IEEE Transactions on Network and Service Management, vol. 20, no. 2, pp. 1345–1362, Jun. 2023, doi: 10.1109/tnsm.2023.3262406.
[3]. D. Dwivedi, P. K. Yemula, and M. Pal, “DynamoPMU: A Physics Informed Anomaly Detection, Clustering, and Prediction Method Using Nonlinear Dynamics on μ PMU Measurements,” IEEE Transactions on Instrumentation and Measurement, vol. 72, pp. 1–9, 2023, doi: 10.1109/tim.2023.3327481.
[4]. Y. Wang, Z. Yu, and L. Zhu, “Intrusion detection for high-speed railways based on unsupervised anomaly detection models,” Applied Intelligence, vol. 53, no. 7, pp. 8453–8466, Jul. 2022, doi: 10.1007/s10489-022-03911-8.
[5]. D. Borda, M. Bergagio, M. Amerio, M. C. Masoero, R. Borchiellini, and D. Papurello, “Development of Anomaly Detectors for HVAC Systems Using Machine Learning,” Processes, vol. 11, no. 2, p. 535, Feb. 2023, doi: 10.3390/pr11020535.
[6]. M. Alizadeh and J. Ma, “High-dimensional time series analysis and anomaly detection: A case study of vehicle behavior modeling and unhealthy state detection,” Advanced Engineering Informatics, vol. 57, p. 102041, Aug. 2023, doi: 10.1016/j.aei.2023.102041.
[7]. J. Yang, Z. Yue, and Y. Yuan, “Deep probabilistic graphical modeling for robust multivariate time series anomaly detection with missing data,” Reliability Engineering & System Safety, vol. 238, p. 109410, Oct. 2023, doi: 10.1016/j.ress.2023.109410.
[8]. Y.-X. Lu, X.-B. Jin, D.-J. Liu, X.-C. Zhang, and G.-G. Geng, “Anomaly Detection Using Multiscale C-LSTM for Univariate Time-Series,” Security and Communication Networks, vol. 2023, pp. 1–12, Jan. 2023, doi: 10.1155/2023/6597623.

[9]. M. Abdallah et al., “Anomaly Detection and Inter-Sensor Transfer Learning on Smart Manufacturing Datasets,” Sensors, vol. 23, no. 1, p. 486, Jan. 2023, doi: 10.3390/s23010486.
[10]. Q. He, G. Wang, H. Wang, and L. Chen, “Multivariate time-series anomaly detection via temporal convolutional and graph attention networks,” Journal of Intelligent & Fuzzy Systems, vol. 44, no. 4, pp. 5953–5962, Apr. 2023, doi: 10.3233/jifs-222554.
[11]. J. Bae, J. H. Lee and S. Kim, “PNI : Industrial Anomaly Detection using Position and Neighborhood Information,” In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, doi: 10.48550/arXiv.2211.12634.
[12]. D. Pan and S. Hamdar, “From Traffic Analysis to Real-Time Management: A Hazard-Based Modeling for Incident Durations Extracted Through Traffic Detector Data Anomaly Detection,” Transportation Research Record: Journal of the Transportation Research Board, p. 036119812311744, Jun. 2023, doi: 10.1177/03611981231174445.
[13]. A. Copiaco et al., “An innovative deep anomaly detection of building energy consumption using energy time-series images,” Engineering Applications of Artificial Intelligence, vol. 119, p. 105775, Mar. 2023, doi: 10.1016/j.engappai.2022.105775.
[14]. H. Liu and L. Li, “Anomaly Detection of High-Frequency Sensing Data in Transportation Infrastructure Monitoring System Based on Fine-Tuned Model,” IEEE Sensors Journal, vol. 23, no. 8, pp. 8630–8638, Apr. 2023, doi: 10.1109/jsen.2023.3254506.
[15]. Y. Qiao, J. Lü, T. Wang, K. Liu, B. Zhang, and H. Snoussi, “A Multi-head Attention Self-supervised Representation Model for Industrial Sensors Anomaly Detection,” IEEE Transactions on Industrial Informatics, pp. 1–10, 2023, doi: 10.1109/tii.2023.3280337.
[16]. Q. Wang and Q. Shen, “Multivariate time-series anomaly detection,” International Conference on Intelligent Systems, Communications, and Computer Networks (ISCCN 2023), Jun. 2023, doi: 10.1117/12.2679609.
[17]. M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King and S. Pan, “A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection,” 2020, arXiv preprint arXiv:2307.03759.
[18]. G. Wang et al., “Anomaly Detection for Data from Unmanned Systems via Improved Graph Neural Networks with Attention Mechanism,” Drones, vol. 7, no. 5, p. 326, May 2023, doi: 10.3390/drones7050326.
[19]. C. Ding, J. Zhao, and S. Sun, “Concept Drift Adaptation for Time Series Anomaly Detection via Transformer,” Neural Processing Letters, vol. 55, no. 3, pp. 2081–2101, Aug. 2022, doi: 10.1007/s11063-022-11015-0.