JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ANOMALY DETECTION IN NETWORK TRAFFIC SIMULATION

Lakshmi Priyanka Vallabhajosyula
Final Year B.Tech student (CSE)
Department of Computing Technologies
SRM Institure of Science and Technology
Chennai, India

Dr. Pretty Diana Cyril
Assistant Professor
Department of Computing Technologies
SRM Institure of Science and
Chennai, India

Abstract:

The overview of traffic anomaly detection analysis provided by Traffic Anomaly Detection enables the security features of multimedia services. The author's method is based on the examination of time aggregating neighbouring traffic periods. MLtechniques and Python are frequently used for anomaly identification in traffic simulation. To begin with, the data from the traffic simulation must be gathered and prepared. This could entail acquiring information on numerous traffic metrics, such as vehicle speed, density, and flow, and then cleaning and translating the information into a format that is appropriate for ML algorithms.

Key Findings: Python, Tcp/Ip, CNN, KNN.

1. Introduction

The practice of spotting unexpected or anomalous behaviour in traffic flow patterns or traffic-related events is known as anomaly detection in the context of traffic simulation. Anomaly detection in traffic simulation can be used to spot unusual traffic behaviour, such as collisions, traffic jams, or unexpected traffic flows, that may have an impact on the functioning of the traffic system as a whole. The researcher for this task uses the Python programming language for anomaly detection in traffic simulation. Anomaly detection in traffic simulation aims to enhance overall traffic efficiency and safety by enabling real-time traffic management, early warnings of possible traffic issues, and anomaly detection. Anomaly detection has applications in transportation engineering, urban planning, and traffic management systems.

2. Background Study

In an emerging area of research, anomaly detection in traffic simulation utilising Convolutional Neural Networks (CNN) and Transmission Control Protocol/Internet Protocol (TCP/IP) with Python integrates computer vision, deep learning, and network programming approaches. CNNs are used to detect anomalies in real-time during traffic

simulations by first training a deep learning model using traffic flow data or video footage of traffic. The benefit of employing CNNs is their capacity to recognise anomalies that can be challenging to spot using conventional statistical techniques. A popular network protocol that enables data transfer between computers linked to the internet is TCP/IP. TCP/IP can be used in traffic simulation to send data between the many sensors, cameras, and control centres that make up a traffic management system. Anomaly detection using TCP/IP involves developing software that can monitor network traffic for abnormal patterns, such as unusual data packet sizes or frequency of data transmissions. Python, TCP/IP, and CNNs provide a potent toolkit for anomaly identification in traffic simulation. Applying anomaly detection methods to traffic simulation presents some additional difficulties that must be overcome. The accuracy of anomaly detection models, for instance, may be impacted by problems with data quality, such as missing or noisy data. Additionally, establishing and maintaining real-time traffic management systems can be challenging and necessitate careful thought regarding elements like system scalability, dependability, and security. By utilising these methods, real-time traffic management systems that can quickly and effectively detect anomalies can be created. Additionally, the collaboration of scholars and practitioners, as well as the expansion of prior work in the subject, is facilitated by the usage of open-source tools and frameworks.

3. Literature review

Over the past few decades, a significant amount of work has been devoted to modelling traffic events and automatic incident identification. Early in the 1970s, occupancy measurements on the upstream and downstream at fixed road sections were used to establish the first method of traffic incident detection [1]. This kind of statistical technique finds large deviations between observed data and traffic characteristics predicted by prior probability or by locating outliers using the standard normal deviation principle [2]. On the other hand, a different kind of statistical methods, such as the well-known McMaster algorithm [3], is based on identifying a large pattern by employing a fundamental flow-speed-occupancy diagram. The individual objectives and characteristics of the application determine the methodology to be used for anomaly identification in traffic simulation using Python and ML techniques. The best strategy for identifying a variety of abnormalities in traffic simulation may involve combining supervised and unsupervised learning techniques with statistical techniques. The effectiveness of the aforementioned algorithms, meanwhile, depends on how accurately the criteria used to identify traffic are chosen.

According to Pei, (2022), CNN is a unique type of deep feed-forward neural network that stands out among all deep learning methods. utilised frequently to categorise photographs. Because of its translation invariance properties and shared-weights architecture, it has advantages in supervised learning [4]. These include the ability to set a range of multilayer perceptrons that are intended to require less preprocessing. It is composed of three tiers of layers, an input layer, a hidden layer, and an output layer, just like practically every other neural network. Convolutional layers, pooling layers, fully connected layers, and normalisation layers are often included in the numerous hidden layers that make up a CNN. This is the main architectural distinction between them.

4. Methodology

In traffic simulation, identifying anomalous events or behaviours that depart from the predicted patterns is a vital task known as anomaly detection [6]. Anomaly detection in traffic simulation can be done using a variety of approaches. One strategy is to find anomalies based on trends in the data using statistical techniques, such as time-series analysis. Another strategy is to study the typical patterns of traffic behaviour using machine learning techniques like neural networks or decision trees, and then spot abnormalities based on deviations from those patterns. A threshold-based detection is a typical approach for anomaly detection in traffic simulation. This entails establishing thresholds for certain characteristics, such as vehicle speed or density, and marking any data that deviate from these limits as anomalies. Even though this method is straightforward and simple to use, it might not be able to pick up intricate or minute irregularities. To identify anomalies as sites that fall outside of these clusters, another strategy is to utilise clustering techniques to group comparable traffic patterns together. Anomalies that are difficult to detect with threshold-based methods may be more easily recognised using this technique.

Using Python and machine learning algorithms, anomaly detection in traffic simulation can be accomplished in a number of ways [7]. To categorise traffic patterns as typical or abnormal, one frequent technique is to utilise supervised learning algorithms, such as decision trees or support vector machines. In order to determine whether new traffic patterns are anomalous or not, a model that has been trained on a labelled dataset of typical traffic patterns must be used [5]. This approach may be useful for identifying clearly defined anomalies, but it might not be appropriate for identifying subtle or complex anomalies. The researcher has to analyse various methods for better findings. For the purpose of developing machine learning methods for anomaly identification in traffic simulation, Python libraries like sci-kit-learn, Pandas, and NumPy can be employed. A variety of supervised and unsupervised learning methods are available with Scikit-learn, and data manipulation and preparation are possible using Pandas and NumPy [8].

5. Outcome Results

```
[81] plt.plot(history.history['loss'], label='loss')
     plt.plot(history.history['val_loss'], label='val_loss')
     plt.xlabel('Epoch')
     plt.ylabel('SCCE Loss')
     plt.legend()
     plt.grid(True)
```

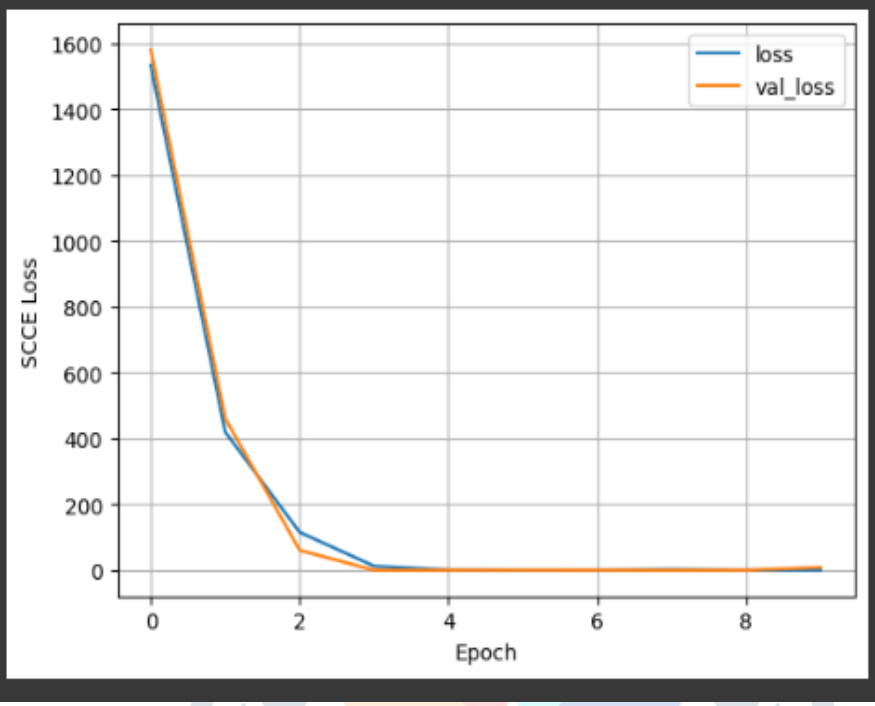

Figure 1: Plot of SCCE Loss vs Epoch

A common way to track the progress of a machine learning model during training is by plotting the Sparse Categorical Cross-Entropy (SCCE) loss against the number of epochs. The SCCE loss is a commonly used loss function in multi-class classification tasks, where each data point belongs to only one class. The plot helps to visualize how the model's performance improves over time as it learns to better predict the correct class labels for the given inputs. By monitoring the SCCE loss curve during training, one can determine whether the model is converging or overfitting to the training data.

```
[82] plt.plot(history.history['accuracy'], label='accuracy')
     plt.plot(history.history['val_accuracy'], label='val_accuracy')
     plt.xlabel('Epoch')
     plt.ylabel('Accuracy')
     plt.legend()
     plt.grid(True)
```

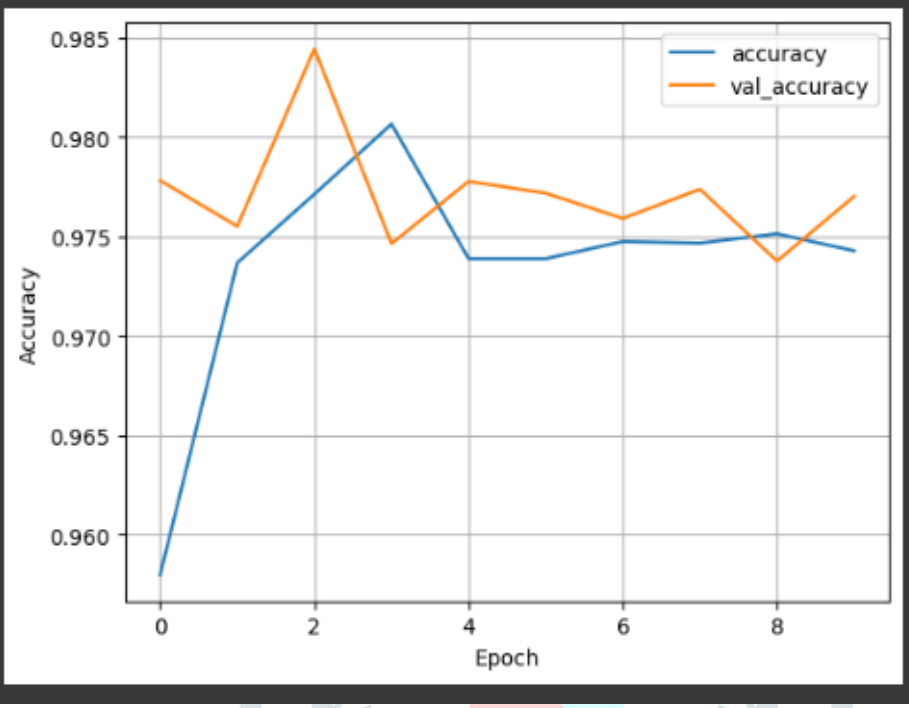

Figure 2: Plot of Accuracy vs Epoch

A common way to visualize a deep learning or machine learning model's progress during training is by plotting the Accuracy versus Epoch. This plot shows how the model's accuracy improves over successive epochs of training. An epoch represents one complete pass of the model over the entire training dataset. Typically, the x-axis of the plot displays the number of epochs while the y-axis shows the accuracy. By monitoring the accuracy versus epoch plot, one can gain insight into the performance of the model and determine if there are any trends or patterns that could indicate potential issues, such as overfitting or underfitting to the training data.

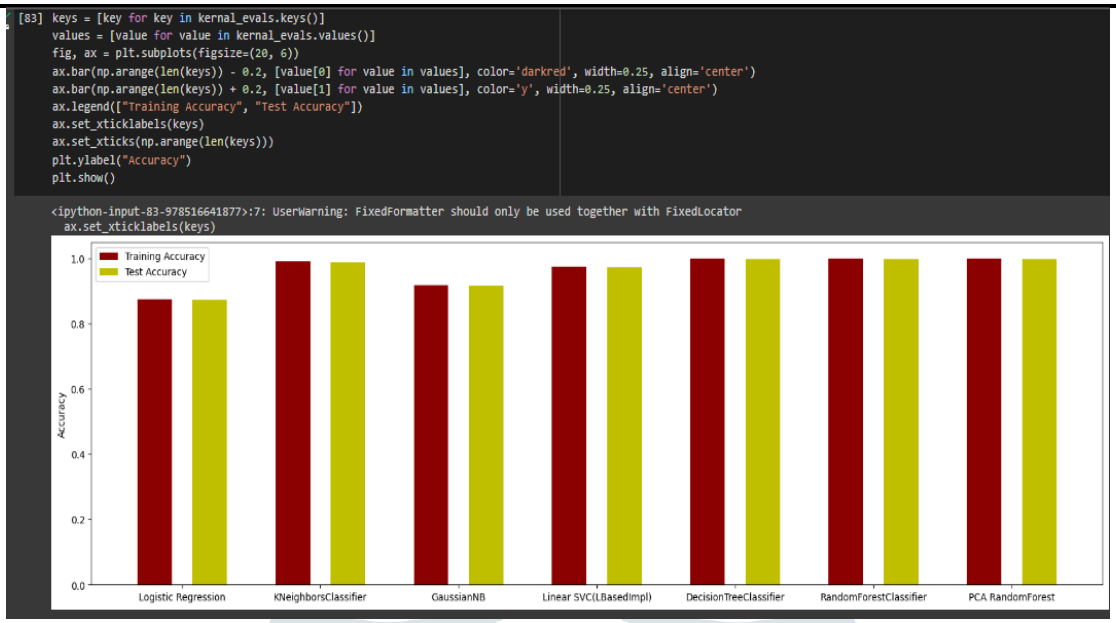

Figure 3: Bar Plot regarding accuracy

In an effort to compare the training accuracy and test accuracy of a machine learning model, a researcher has created a bar plot. This type of visualization, also referred to as a bar chart or bar graph, is a graphical representation of data that utilizes rectangular bars to display the distribution or frequency of information across different categories or groups. Bar plots are commonly used to compare and contrast data across multiple categories or groups, making them a popular choice for data visualization tasks. By creating a bar plot that compares the training accuracy and test accuracy of a model, the researcher can easily assess the model's performance and identify potential issues such as overfitting or underfitting. The bar plot provides a clear and concise way to visualize the proportional differences between the two types of accuracy and offers valuable insights into the model's overall performance.

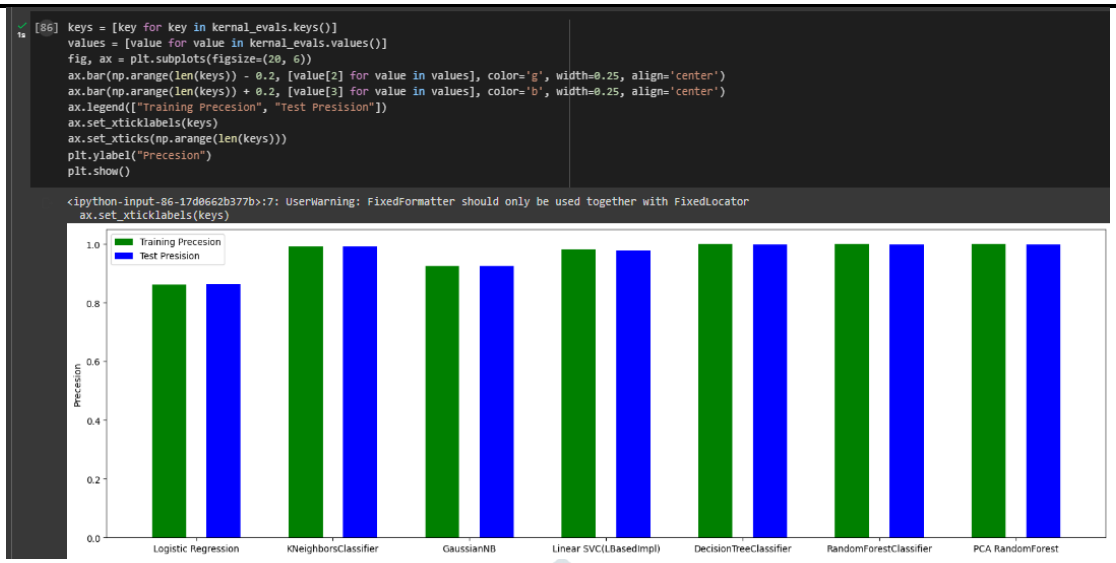

Figure 4: Bar plot regarding precision

A bar plot is a graphical tool that is commonly used to compare and contrast the variability or precision of data points across multiple categories or groups. Precision, in statistical terms, is a measure of the accuracy and consistency of a measurement or estimate, and is closely related to measurement error. Bar plots can be used to visualize precision in several ways, such as by displaying the range, standard deviation, or confidence intervals of data points. By utilizing these visual cues, bar plots can effectively illustrate the accuracy and variability of data in a clear and concise manner. This makes them a valuable tool for analyzing and interpreting data in a variety of fields, including scientific research, finance, and marketing.

5. Conclusion

After analyzing the entire task the researcher performed the entire approach and the extra curriculum activities of the traffic simulation. Python-based anomaly detection in traffic simulation is a viable method for enhancing traffic efficiency and safety. The development of real-time traffic management systems that can quickly and effectively detect anomalies is achievable by utilising machine learning techniques like CNNs with network programming languages like TCP/IP. Python offers a flexible and potent foundation for creating such systems, with a large selection of tools and frameworks for network programming, machine learning, and data processing. Additionally, Python's open-source tool and framework usage facilitates collaboration between researchers and practitioners and the expansion of prior studies in the subject.

6. Future Scope

There is great potential for utilizing machine learning (ML) techniques in the field of traffic simulation systems to predict and detect anomalies. ML-based anomaly detection in traffic simulation systems holds promise for improving the effectiveness, security, and sustainability of the transportation system. There is a need for more sophisticated deep-learning models specifically designed for the detection of anomalies in traffic simulations. Integrating different neural network architectures, such as combining Convolutional Neural Networks (CNN), Deep Neural Networks

(DNN), and Recurrent Neural Networks (RNN), could help capture various patterns and dynamics in traffic data and enhance the accuracy and resilience of the anomaly detection models. Currently, anomaly detection techniques in traffic simulation systems mostly rely on a single type of data, such as sensor data or GPS data. However, combining data from multiple sources, including social media, sensors, GPS, and simulation data, could provide a more comprehensive and holistic understanding of traffic patterns and behaviors. Future studies could focus on developing methods that efficiently integrate and utilize data from multiple sources to improve the accuracy and responsiveness of anomaly identification in traffic simulation. Real-time anomaly detection in traffic simulation is crucial for many real-world applications, such as traffic management, intelligent transportation systems, and incident detection. Future research could focus on developing real-time anomaly detection techniques that process and analyze traffic data instantly, providing decision-makers with immediate and actionable information. This may involve developing scalable, lightweight models that can efficiently handle massive amounts of data in real-time and deliver precise anomaly detection results. The development of ML-based anomaly detection techniques in traffic simulation systems has the potential to revolutionize the transportation industry, enhancing the safety and efficiency of our roads and highways.

Reference List

[1]Pei, J., Zhong, K., Jan, M.A. and Li, J., 2022. Personalized federated learning framework for network traffic anomaly detection. *Computer Networks*, *209*, p.108906.

[2]Luo, Y., Xiao, Y., Cheng, L., Peng, G. and Yao, D., 2021. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Computing Surveys (CSUR)*, *54*(5), pp.1-36.

[3]Anton, S.D.D., Sinha, S. and Schotten, H.D., 2019, September. Anomaly-based intrusion detection in industrial data with SVM and random forests. In *2019 International conference on software, telecommunications and computer networks (SoftCOM)* (pp. 1-6). IEEE.

[4]Stiawan, D., Idris, M.Y.B., Bamhdi, A.M. and Budiarto, R., 2020. CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, *8*, pp.132911-132921.

[5]Zhang, Y.M., Wang, H., Wan, H.P., Mao, J.X. and Xu, Y.C., 2021. Anomaly detection of structural health monitoring data using the maximum likelihood estimation-based Bayesian dynamic linear model. *Structural Health Monitoring*, *20*(6), pp.2936-2952.

[6]Rong, H., Teixeira, A.P. and Soares, C.G., 2020. Data mining approach to shipping route characterization and anomaly detection based on AIS data. *Ocean Engineering*, *198*, p.106936.

[7Doshi, K. and Yilmaz, Y., 2020. Fast unsupervised anomaly detection in traffic videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (pp. 624-625).

[8]Hwang, R.H., Peng, M.C., Huang, C.W., Lin, P.C. and Nguyen, V.L., 2020. An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access*, *8*, pp.30387-30399.

[9]Mokhtari, S., Abbaspour, A., Yen, K.K. and Sargolzaei, A., 2021. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics*, *10*(4), p.407.