ODFM: Abnormal Traffic Detection Based on Optimization of Data Feature and Mining

Xianzong Wu
College of Intelligence and Computing, Tianjin University, Tianjin, China

***Abstract*—The booming of computer networks and software applications has led to an explosive growth in the potential damage caused by network attacks. Efficient detection of abnormal traffic in networks is appealing for facilely mastering the traffic tracking and locating for network usage at low resource cost. High quality abnormal traffic detection of Internet becomes particularly relevant during the automated services of multiple application situations. This paper proposes a novel abnormal traffic detection algorithm called ODFM based on the optimization of data feature and mining. Specially, we develop a feature selection strategy to reduce the feature analysis dimension, and set a peer-to-peer (P2P) traffic identification module to filter and mine the related service traffic to reduce the amount of data detection and facilitate the abnormal traffic detection. Experimental results demonstrate that the proposed algorithm greatly improves the detection accuracy, which verifies its effectiveness and competitiveness in the general tasks of abnormal network traffic detection.**

***Keywords*—Abnormal traffic; detection; data mining; feature dimension optimization; network security**

I. Introduction

With the increasing complexity and volume of network traffic, the need for effective anomaly traffic detection algorithms is essential to ensure the security and reliability of networks. Anomaly traffic, which deviates significantly from normal patterns, can indicate potential cyber threats, network performance issues, or abnormal user behavior [1]. Anomaly traffic detection involves the identification of abnormal patterns or behaviors within network traffic that deviate significantly from the expected norm. The study has drawn great attention in recent decades due to the strong security demands on the network communications. However, traffic detection is deemed to be a developing and challenging issue since it needs to deal with various difficulties coming from imbalanced data, increasing network traffic volume, evolving and sophisticated attacks, as well as dynamic and variable network environments [2-5]. Various techniques have been developed to detect and analyze such anomalies, ranging from statistical-based methods [6-8], time series analysis [9-11], machine-learning based methods [12-14], deep-learning based methods [15-17], ensemble methods [18-20], flow-based analysis [21-23], hybrid methods [24, 25], to unsupervised clustering methods [26, 27].

Statistical based methods analyze network traffic data statistically to model normal behavior and detect anomalies [8]. These methods include mean-based models, standard deviation-based models, moving average models, and Gaussian distribution modeling. Time series analysis techniques model network traffic as a time-ordered series of data points, capturing temporal dependencies and patterns. Techniques such as Autoregressive Integrated Moving Average (ARIMA) [28], Hidden Markov Models (HMM) [29], and Wavelet Transform [30] is commonly used for time series analysis in anomaly detection. On the other hand, machine learning algorithms leverage historical network traffic data to distinguish between normal and anomalous patterns. Supervised learning algorithms like Support Vector Machines (SVM) [31], Random Forests [32], and Neural Networks classify traffic based on labeled datasets. Unsupervised learning algorithms, such as clustering and density estimation, detect anomalies without prior labeling. Deep learning algorithms, such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), automatically learn hierarchical and temporal patterns from raw network traffic data, excelling in detecting complex and evolving anomalies. However, they require significant computational resources and large labeled datasets. Ensemble methods combine multiple anomaly detection models to improve overall detection performance. Techniques like bagging, boosting, and stacking reduce false positives and enhance accuracy. Flow-based analysis aggregates network traffic into flows, representing communication between specific source and destination IP addresses. Flow-based anomaly detection uses flow features such as traffic volume, duration, or byte count to identify deviations from normal flow behavior. Besides, hybrid methods integrate multiple detection techniques, such as statistical, machine learning, and rule-based methods, to enhance accuracy. These approaches leverage the strengths of each technique to effectively identify a wide range of anomalies. Unsupervised clustering methods group network traffic instances into clusters, identifying anomalies as instances that deviate or form separate clusters. Clustering algorithms like k-means [33], DBSCAN [34], and self-organizing maps (SOM) [35] are commonly employed for anomaly traffic detection.

These mentioned methods have been faced with various difficulties and challenges. For instance, many continually develop sophisticated techniques to evade detection, employing tactics like traffic encryption, obfuscation, and mimicry. This poses a challenge for anomaly detection systems to identify and classify these evolving attacks accurately. Network traffic

often exhibits an imbalanced distribution, where normal traffic significantly outweighs anomalous traffic. Imbalanced data can lead to biased models that exhibit a higher false positive rate or overlook certain types of anomalies. Handling imbalanced data and addressing the bias towards the majority class is critical for effective anomaly detection. Network environments are dynamic, constantly evolving with changing user behavior, new applications, and network configurations. Anomaly detection algorithms need to adapt and remain capable of detecting novel or emerging anomalies. This requires continuous monitoring, model updates, and the ability to adapt to evolving network dynamics. Besides, real-time anomaly detection places stringent requirements on processing time and scalability. Efficient and scalable algorithms are necessary to handle large-scale network traffic, detect anomalies within strict time constraints, and ensure real-time analysis. Moreover, deep learning and complex machine learning models, while powerful in detecting anomalies, often lack interpretability and explain ability. Understanding the rationale behind anomaly detection outcomes is essential for effective response and decision-making by analysts. Ensuring transparent and interpretable anomaly detection models is a challenging task [36]. These challenges are actively being explored and addressed through ongoing research and development efforts. Advancing anomaly traffic detection techniques requires focusing on handling evolving attacks, addressing imbalanced and limited labeled data, enhancing adaptability to dynamic environments, reducing false positives, and improving interpretability. By addressing these challenges, the future advancement of anomaly detection systems in network traffic analysis can be achieved.

Developing appropriate anomaly detection method need to consider several factors, including the specific network environment, the characteristics of the anomalies, computational resources, and available data. Ongoing research and studies are actively exploring novel algorithms, advancements in machine learning techniques, and the integration of domain knowledge to enhance the accuracy and efficiency of anomaly detection in network traffic. These efforts aim to improve the detection capabilities in diverse network environments and adapt to evolving threat landscapes. In this paper, we propose a new approach called ODFM to anomaly traffic detection that jointly optimizes the data volume and feature dimension to achieve high-accuracy detection results. The main contributions of the paper include:

1) The paper propose a novel method of combining feature dimension reduction preprocessing with data mining optimization to general tasks of anomaly traffic detection, which takes into account the P2P traffic identification and the related-service traffic filter.

2) We introduce a feature selection strategy to achieve the feature dimension reduction, which can automatically locate key traffic information, speed up the anomaly detection and then propel the data optimization and engine extraction. It effectively addresses the issue of feature vector construction of data mining model.

3) An elaborate system design is conducted to verify the effectiveness of the proposed ODFM algorithm. Experimental evaluations have demonstrated its efficacy and great potential under various traffic anomaly detection conditions.

II. The Proposed ODFM Method

A. Foundation Statement

The proposed ODFM method is designed to address the issue of anomaly traffic detection. To achieve the goal of ODFM, we develop an ODFM based anomaly network traffic detection system that can effectively identify and classify abnormal patterns within network traffic data. The system aims to enhance network security, mitigate potential threats, and optimize network performance. The system mainly involves data mining mechanism to complete the network anomaly traffic detection.

Data mining mechanism aims to facilitate advanced anomaly detection and prevention in network traffic through the application of association mechanisms. When employing association mechanisms, it becomes feasible to effectively detect patterns and hidden knowledge between diverse and interconnected data items. By combining various data attribute values, these mechanisms can predict attribute values for a certain class of data, which provides significant advantages in acquiring and utilizing patterns from massive datasets. Within computer information systems, the utilization of association mechanisms enables precise analysis of network anomalies, fault information, and user network data. The formation of fault factor sets and the integration and analysis of related information categories enhance the scrutiny of traffic data and management processes. Moreover, potential rules between different network information and data are derived, thereby lessening the likelihood of network risks and achieving efficient early warning and handling of abnormal network traffic.

The primary objective of this anomaly network traffic detection system is to reduce the data volume and feature dimensions processed by the anomaly detection module, thereby enhancing the detection accuracy. This system focuses specifically on detecting malicious scanning and denial of service (DoS)/ distributed denial of service (DDoS) anomalies under the transmission control protocol, making it particularly applicable in transmission control protocol environments with a certain scale of hosts. Traditional anomaly traffic detection systems typically involve dealing with massive input data and implementing high-dimensional feature models for training. In contrast, the design of this system develops a feature selection strategy to reduce the dimensionality of feature analysis, and includes a P2P traffic identification module to filter relevant business traffic, thus reducing the data processing load and improving the accuracy of the detection process.

B. The Workflow of ODFM System

Fig. 1 depicts the detailed workflow of the proposed ODFM system architecture, which is designed with two primary phases, off-line training and online classification. In the off-line training phase, real-time network traffic and the corresponding known traffic labels are processed to create labeled training data. Through the application of hybrid feature extraction techniques in data preprocessing, an offline traffic

classification model is developed. During the online classification stage, the system performs real-time network traffic collection and an off-line data file analysis within the traffic collection and analysis module. All essential field information related to the traffic is stored in the database for further analysis. In the data preprocessing module, the feature selection algorithm and feature extraction engine are utilized. The former is employed in an offline state for comprehensive analysis, while the latter conducts secondary statistical analysis, reorganization, and calculation of the original traffic data from the database. This facilitates the construction of feature vectors for subsequent data mining models.

The data preprocessing module encompasses some vital components such as data storage and access, key table structural design, storage design, and trigger design. These elements ensure an efficient handling of the data during processing. Within the abnormal traffic classification detection module, the primary objective is to identify and filter P2P traffic, construct a weak classifier, and enhance its performance. By following this system design, it is possible to improve the accuracy and effectiveness of online traffic classification, enabling efficient detection and analysis of abnormal network traffic.

C. The Architecture Design

To achieve comprehensive maturity and facilitate modular construction in developing the proposed ODFM system for general anomaly network traffic detection, it is imperative to design and establish a corresponding software architectural framework. Maturity, in this context, refers to universally recognized technology that guarantees superior system performance, ultimately enhancing the system's long-term viability. Modular construction involves methodically dividing software functionality into autonomous modules, thereby streamlining software maintenance and facilitating seamless upgrades.

The ODFM system is meticulously constructed upon the robust Model-View-Controller (MVC) design pattern, adopting a multi-layered approach. The design ensures an efficient segregation of crucial elements, encompassing business logic, interface display, and data models. As a result, the ODFM system achieves parallel operations, which can bolster its computational capabilities, and thereby optimizing the overall performance of network anomaly traffic detection.

The software technology implemented in the proposed system predominantly encompasses three distinctive layers. Fig. 2 illustrates the three-layer technical architecture. Firstly, Presentation Layer unveils an intuitive and user-friendly interface for seamless system operation and discernible result display. It adeptly handles the input data, promptly forwarding it to the Business Processing Layer or, in turn, receiving pertinent data communicated from the Business Parallel Processing Layer. Secondly, Business Processing Layer encompasses specific business logic, which is further partitioned into three sub-layers, including data acquisition, business applications and external interfaces. Notably, the external interfaces facilitate seamless data exchange and robust sharing capabilities between the ODFM system and other compatible systems, fostering optimal data application. Web Services has been judiciously chosen as the technology of choice for shaping these external interfaces. Lastly, Data Layer assumes the responsibility for efficient storage and meticulous management of indispensable network traffic information. It reliably provides indispensable data that fuels the construction of intricately designed classification models within the system. By harnessing the power of this meticulously crafted software technology architecture, the ODFM system confidently attains exemplary levels of maturity, modularity, and operational efficiency. With this robust framework in place, the system flawlessly accomplishes reliable network traffic analysis and impeccable management.

D. Implementation of Modules in ODFM

The ODFM design implementation adopts the Java programming language. For the proper functioning of the windows packet capture (Winpcap) in Java, it relies on utilizing Winpcap at the data link layer control level. Winpcap provides the required network underlying resource access interface for Java, offering extensive capabilities to interact with the network resources. This implementation ensures the advantage of maintaining system independence during the operational process.

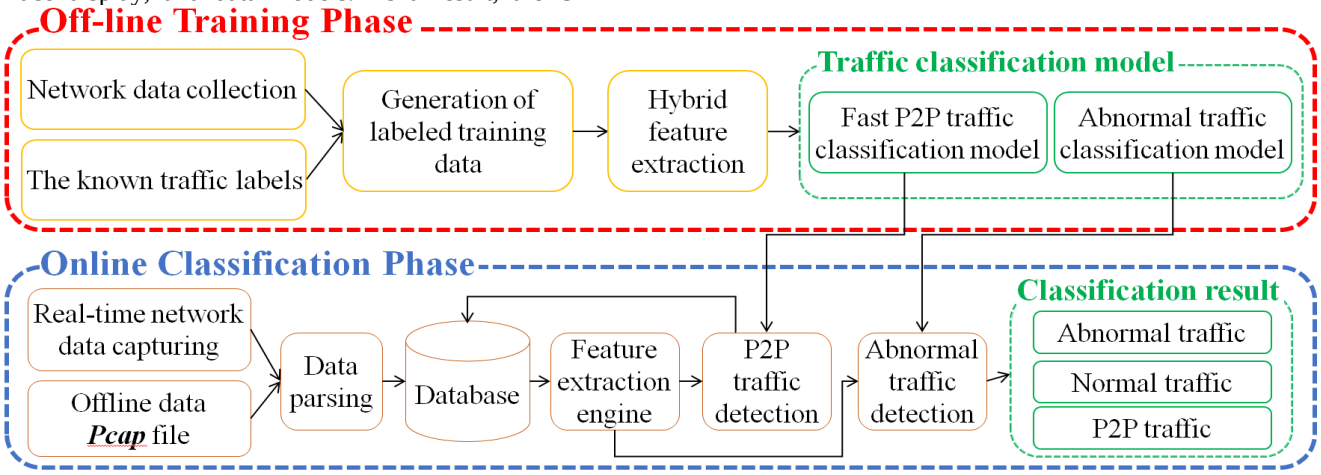

Fig. 1. The detailed flow of the proposed ODFM, which contains an off-line training phase and an online classification phase.

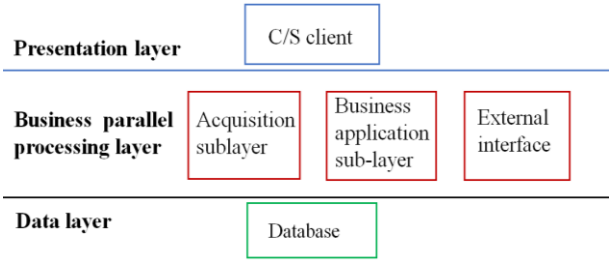

Fig. 2. The three-layer technical architecture of ODFM system.

In the module of traffic collection and parsing, Winpcap serves as a network data capturing framework, which offers excellent application performance in practical scenarios. It consists of filters, wpcap.dll, and packet.dll components. During the data capturing, the JpcapCaptor class provided by Jpcap enables the capture of network traffic. Typically, an instance object of this class is suited for specific network adapter devices, allowing various specified operations to be carried out. Firstly, network device selection is performed using the JpcapCaptor.getDeviceList() function of Java, which returns an array of NetworkInterface objects, representing the available network devices. Opening the network interfaces is done in a static mode through the openDevice() method, which requires four parameters, including enabling promiscuous mode. Secondly, packet capture is conducted using the processPacket() and loopPacket() callback methods within the JpcapCaptor class. The former is commonly used and supports non-blocking and timeout policies, while the latter does not provide these features. Thirdly, filtering rules can be set using Jpcap to achieve the filtration of unwanted packets. For instance, using an IP filter expression would only retain IP packets. Applying such filter rules not only reduces the system's data processing load but also improves application performance significantly. During the system operation, once network traffic capture is realized, the data packets need to be analyzed and processed. By extracting key header fields and conducting comprehensive data and feature extraction, the groundwork is laid for future anomaly traffic detection.

In the module of data storage, traffic storage mainly lays the data foundation for the data mining model. In the construction of the ODFM system, the open-source MySQL database server is selected as the database management system. To enhance data storage efficiency and application performance, the c3p0 connection pool technology is also utilized. This involves submitting the opening and closing of database connections to the connection pool, which significantly improves the efficiency of data access at the application level. For the data parsing, the c3p0-config.xml is configured with the necessary database parameters. The inclusion of jar packages such as c3p0-0.9.1.2.jar, mysql-connector-java-5.08-bin.jar (the MySQL database driver package), and commons-dbutils-1.4.jar (JDBC encapsulation library) enables simplified development using the dbutils framework. This framework streamlines development by encapsulating result sets, managing resource releases, and facilitating database transactions. The database storage module design primarily focuses on table, trigger, and stored procedure implementations to fulfill the requirements of business needs. Trigger design plays a pivotal role in connecting the P2P traffic identification module with the abnormal traffic detection module. It enables the effective filtering of P2P traffic within transmission control protocol (TCP), thereby enhancing the accuracy of abnormal traffic data processing and reducing false positives. This is accomplished through a trigger that automatically deletes P2P traffic in the tcp_table based on IP addresses when a result is inserted into the p2p_result_table, thereby facilitating data filtering. These implementations are critical in meeting the functional requirements of the ODFM system while optimizing database storage efficiency and supporting accurate traffic analysis.

In the module of feature extraction engine, effective feature extraction provides corresponding feature samples for the traffic classification model. Fig. 3 provides a complete transition diagram of a TCP connection state. During the implementation of this module, statistical analysis of traffic information is conducted, and certain TCP sessions are maintained. For instance, non-three-way handshake RST packets are eliminated. The module's core revolves around the utilization of different network connection states and the sequence of state transformations. This allows for the description of traffic information and enables the incorporation of state statistics mechanisms for analyzing packets in various states. In a TCP connection, if a TCP flow is initiated by the SYN initiator, the first forward starting point is determined. All directions are indicated as forward, denoted by "+", while the direction of packets transmitted by the counterpart is represented by "-". In a complete TCP connection process, both nodes involved in the connection can initiate the connection, resulting in a symmetrical transformation graph. However, statistical analysis is typically performed based on one of the states. A complete TCP connection usually comprises three stages: the three-way handshake, data exchange, and connection release. Generally, only the relevant states are iterated in a self-looping manner. The application of this module allows for a comprehensive analysis of TCP traffic, ensuring accurate tracking of connection states and statistical analysis of corresponding packets. By considering the unique characteristics of each state, the module enables effective traffic analysis and facilitates the identification of potential anomalies or irregularities within TCP connections. During the operation of data feature extraction, this study focuses on analyzing packets that fall within {SYN, SYN/ACK, ACK, RST/ACK, RST, FIN, FIN/ACK, Data} categories. Any malformed packets that do not adhere to the RFC specification are promptly flagged for immediate alarm. In cases where the TCP connection is normal or RST packets are detected, the stream recording is halted, and the packet information is recorded and the connection is cleared. To effectively maintain the same flow information, the TCP packet sequence number is leveraged. Suppose the sequence number is denoted as S, the load size of the current packet is represented as L, and the sequence number of the subsequent TCP packet is $S+L$. This approach allows for accurate tracking and management of the flow information. On the other hand, the feature extraction of P2P traffic identification is relatively straightforward. It involves that extracting the payload length information from each UDP stream and realizing packet statistics using a List set. By focusing on these specific packet categories and applying appropriate techniques for maintaining flow information and

P2P traffic feature extraction, the ODFM streamlines the traffic analysis process and ensures the identification of abnormal or irregular packets while adhering to standardized protocols.

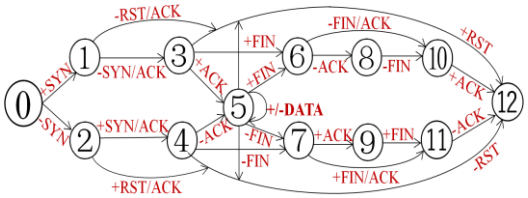

Fig. 3. The transition diagram of a TCP connection state.

In the module of abnormal traffic classification module, we analyze the relevant feature samples extracted by the feature extraction engine module in the traffic classification module to obtain classification results. This allows us to construct a decision tree classifier and apply the AdaBoost algorithm [37] in the anomaly traffic classification module. With this operation, the classifier can be transformed into a strong classifier, which further enhances the accuracy of P2P traffic identification without the need for additional boosting.

In the anomaly traffic classification process, we primarily focus on the off-line training and online classification calculations. For practical classification, we need to develop additional functionality based on the related *jar* packages of *Weka*. Therefore, understanding the underlying algorithms encapsulated in Weka is crucial. Once the data mining process is complete, the anomalous network traffic is passed to the alert module, which generates alert information. The files containing non-anomalous traffic information are then deleted, completing the entire anomaly network traffic detection process. The decision tree training model is stored in memory. By loading the training samples and applying the "*buildClassifier* (Instances instances)" method, the trained classifier model is loaded into memory. Through these procedures, we can effectively analyze and classify abnormal network traffic, which ensures accurate detection and response to potential threats.

III. Experimental Evaluation

In this section, comprehensive experiments are conducted to validate the effectiveness of our proposed ODFM method in the context of anomaly network traffic detection tasks. Stress tests are carried out on each module to analyze the system's traffic processing capabilities. The average processing speed of each module is evaluated and presented in Table I, which highlights the competitive speed of the abnormal traffic classification detection module and indicates favor consumed time when processing a mass of data packets in the modules of traffic collection, parse and store. Although the system design incorporates the AdaBoost modeling strategy [37], which may initially require a longer training time, real-time classification becomes feasible after successfully completing the model building process. The trained model can be serialized in memory, enabling efficient offline training. During real-time classification, only the feature samples need to be directly input for processing.

Furthermore, the system employs multi-threaded parallel computing to pursue efficient processing times. For instance, the processing of 500,000 network data is completed in less than 180 seconds. However, in practical applications, traffic capture and statistics are typically implemented at minute intervals. In general, the statistical interval time above 180s can be considered to meet the real-time requirements of the system. The experimental results indicate the robustness and efficiency of the proposed ODFM method for anomaly network traffic detection, indicating its suitability and effectiveness for real-time applications.

TABLE I. Results of Data Processing of each Module

Module\Parameter	Packets (ten-thousand)	Time (s)
Collection, Parse, Store	19.6	60
Feature extraction engine	51.2	60
Traffic classification detection	100	5.8

IV. Conclusion

In this paper, we propose a novel anomaly traffic detection approach called ODFM that incorporates the optimization of feature dimension reduction and data mining. Different from the traditional methods that take massive data as input and implement the complex design of high-dimensional feature sample to achieve the model training, this paper first adopts the feature selection mechanism to reduce the feature analysis dimension, and sets the P2P traffic identification module to filter the related service traffic, so as to reduce the amount of data detection and improve the detection accuracy. The motivation behind ODFM is that an optimization of approximate feature dimension and normal traffic mining can complete fast anomaly detection while ensuring high detection accuracy. Experiments indicate that the whole pipeline can produce competitive detection results, and it can be able to address various challenging anomaly traffic situations, showing its obvious efficacy in the task of network anomaly traffic detection.

References

[1] Z. Minjie and Z. Yilian, "Abnormal Traffic Detection Technology of Power IOT Terminal Based on PCA and OCSVM," 2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2023, pp. 549-553.

[2] H. Gong, C. Liu, W. Gao, L. Wang and X. Wang, "MSTP Network Data Traffic Anomaly Optimization Detection Algorithm," 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS), Chengdu, China, 2023, pp. 32-35.

[3] M. Cao, D. -n. Cheng, X. Wu and B. Wang, "Research on Auto-adaptive Traffic-Aware Abnormal Detection Method," 2009 International Symposium on Information Engineering and Electronic Commerce, Ternopil, Ukraine, 2009, pp. 445-449.

[4] G. Chen and X. Tan, "Network abnormal traffic detection method based on multi kernel KPCA-PSO-ELM," 2021 2nd International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI), Shenyang, China, 2021, pp. 183-187.

[5] H. Li, E. He, C. Kuang, X. Yang, X. Wu and Z. Jia, "An Abnormal Traffic Detection Based on Attention-Guided Bidirectional GRU," IEEE 22nd International Conference on Communication Technology (ICCT), Nanjing, China, 2022, pp. 1300-1305.

[6] G. Kaur, V. Saxena and J. P. Gupta, "Anomaly Detection in network traffic and role of wavelets," International Conference on Computer Engineering and Technology, Chengdu, China, 2010, pp. V7-46-V7-51.

[7] P. Kromkowski, S. Li, W. Zhao, B. Abraham, A. Osborne and D. E. Brown, "Evaluating Statistical Models for Network Traffic Anomaly Detection," 2019 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2019, pp. 1-6.

[8] I. C. Paschalidis and G. Smaragdakis, "A Large Deviations Approach to Statistical Traffic Anomaly Detection," Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA, 2006, pp. 1900-1905.

[9] H. Zhao et al., "A Fourier Series-Based Anomaly Extraction Approach to Access Network Traffic in Power Telecommunications," 2017 International Conference on Computer Systems, Electronics and Control (ICCSEC), Dalian, China, 2017, pp. 550-553.

[10] S. M. A. Karim, N. Ranjan and D. Shah, "A Scalable Approach to Time Series Anomaly Detection & Failure Analysis for Industrial Systems," Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2020, pp. 0678-0683.

[11] R. Sharma, N. Singh and S. Birla, "An Experimental Study for Comparing Different Method for Time Series Forecasting Prediction & Anomaly Detection," International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2021, pp. 1-4.

[12] A. Guezzaz, Y. Asimi, M. Azrour and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," in Big Data Mining and Analytics, vol. 4, no. 1, pp. 18-24, March 2021.

[13] A. Khudoyarova, M. Burlakov and M. Kupriyashin, "Using Machine Learning to Analyze Network Traffic Anomalies," IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia, 2021, pp. 2344-2348.

[14] R. Singh, N. Srivastava and A. Kumar, "Machine Learning Techniques for Anomaly Detection in Network Traffic," International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, pp. 261-266.

[15] Y. Sun, H. Ochiai and H. Esaki, "Deep Learning-Based Anomaly Detection in LAN from Raw Network Traffic Measurement," Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2021, pp. 1-5.

[16] Y. Li et al., "NIN-DSC: A Network Traffic Anomaly Detection Method Based on Deep Learning," International Conference on Signal and Image Processing (ICSIP), Suzhou, China, 2022, pp. 390-394.

[17] H. Tong, "Research on Multiple Classification Detection for Network Traffic Anomaly Based on Deep Learning," International Symposium on Computer Science and Intelligent Control (ISCSIC), Beijing, China, 2022, pp. 12-16.

[18] D. Yang and M. Hwang, "Unsupervised and Ensemble-based Anomaly Detection Method for Network Security," International Conference on Knowledge and Smart Technology (KST), Chon buri, Thailand, 2022, pp. 75-79.

[19] W. Huan, H. Lin, H. Li, Y. Zhou and Y. Wang, "Anomaly Detection Method Based on Clustering Undersampling and Ensemble Learning," IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 980-984.

[20] A. S. Varal and S. K. Wagh, "Misuse and Anomaly Intrusion Detection System using Ensemble Learning Model," 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), Bhubaneswar, India, 2018, pp. 1722-1727.

[21] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop and M. A. Marotta, "A New Method for Flow-Based Network Intrusion Detection Using the Inverse Potts Model," IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1125-1136, June 2021.

[22] M. S. E. Sayed, N. -A. Le-Khac, M. A. Azer and A. D. Jurcut, "A Flow-Based Anomaly Detection Approach with Feature Selection Method Against DDoS Attacks in SDNs," IEEE Transactions on Cognitive Communications and Networking, vol. 8, no. 4, pp. 1862-1880, Dec. 2022.

[23] Z. Jadidi, V. Muthukkumarasamy, E. Sithirasenan and K. Singh, "Flow-based anomaly detection using semisupervised learning," 2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS), Cairns, QLD, Australia, 2015, pp. 1-5.

[24] V. Timcenko and S. Gajin, "Hybrid Machine Learning Traffic Flows Analysis for Network Attacks Detection," Telecommunications Forum (TELFOR), Belgrade, Serbia, 2022, pp. 1-8.

[25] D. Vinod and M. Prasad, "A novel hybrid automatic intrusion detection system using machine learning technique for anomalous detection based on traffic prediction," International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-7.

[26] A. G. Roselin, P. Nanda, S. Nepal and X. He, "Intelligent Anomaly Detection for Large Network Traffic with Optimized Deep Clustering (ODC) Algorithm," in IEEE Access, vol. 9, pp. 47243-47251, 2021.

[27] G. Ping, S. Feng, Y. Li and X. Ye, "Unsupervised Anomalous Traffic Detection Based on Cascading Representation and Multiple-Clustering," IEEE 8th International Conference on Computer and Communications (ICCC), Chengdu, China, 2022, pp. 2303-2307.

[28] A. H. Yaacob, I. K. T. Tan, S. F. Chien and H. K. Tan, "ARIMA Based Network Anomaly Detection," International Conference on Communication Software and Networks, Singapore, 2010, pp. 205-209.

[29] K. M. León-López, F. Mouret, H. Arguello and J. -Y. Tourneret, "Anomaly Detection and Classification in Multispectral Time Series Based on Hidden Markov Models," IEEE Transactions on Geoscience and Remote Sensing, vol. 60, pp. 1-11, 2022.

[30] V. Geppener and B. Mandrikova, "Combination of wavelet transform and Autoencoder for complex data analysis and anomaly detection," 2021 International Conference on Information Technology and Nanotechnology (ITNT), Samara, Russian Federation, 2021, pp. 1-4.

[31] S. Kumar, S. Nandi and S. Biswas, "Research and application of One-class small hypersphere support vector machine for network anomaly detection," International Conference on Communication Systems and Networks (COMSNETS 2011), Bangalore, India, 2011, pp. 1-4.

[32] D. Yao, M. Yin, J. Luo and S. Zhang, "Network Anomaly Detection Using Random Forests and Entropy of Traffic Features," International Conference on Multimedia Information Networking and Security, Nanjing, China, 2012, pp. 926-929.

[33] Z. Zhu, Y. Xie, X. Yang and W. Hu, "A fast anomaly network traffic detection method based on the constrained k-nearest neighbor," 2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2023, pp. 318-323.

[34] Z. G. Prodanoff, A. Penkunas and P. Kreidl, "Anomaly Detection in RFID Networks Using Bayesian Blocks and DBSCAN," SoutheastCon, Raleigh, NC, USA, 2020, pp. 1-7.

[35] K. Saha, M. M. Rahman Fakir and M. M. A. Hashem, "An Unsupervised Self-Organizing Map Assisted Deep Autoencoder Gaussian Mixture Model for IoT Anomaly Detection," International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 2021, pp. 1-6.

[36] M. Shen et al., "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 791-824, Firstquarter 2023.

[37] W. Li and Q. Li, "Using Naive Bayes with AdaBoost to Enhance Network Anomaly Intrusion Detection," International Conference on Intelligent Networks and Intelligent Systems, Shenyang, China, 2010, pp. 486-489.