Critical Events Detection Based on Alert Logs in Smart Grid

Wenmin Li[1], Ning Dong[2], Haoliang Zhao[1,*], Jianlin Jiao[2], Hao Xv[2], Bo Li[3] and Minghui Gao[3]
[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian District, Beijing 100876, China
[2]State Grid Beijing Electric Power Company, Xicheng District, Beijing 100031, China
[3]NARI Group Corporation, Nanjing 211106, China
[*]Corresponding author

***Abstract*—Alert logs in the smart grid come from a variety of security devices and hosts. There are a large number of false alerts and low-threat alerts in the massive logs, which not only make the real threat difficult to be discovered, but also increase the difficulty of analysis. Therefore, based on the fact that there are anomalies in the process of the outbreak of security events, the concept of critical events and the critical event detection model in smart grid based on statistical analysis are proposed. The statistical analysis method is combined with the security event logs and the critical events detection algorithm is designed. The result from real data in the smart grid shows that the model can effectively detect critical events with an accuracy rate of 98%.**

Keywords*—smart grid; anomaly detection; critical event; log; alert

I. Introduction

With the informationizion upgrade of power companies and the promotion of the construction of energy Internet, power generation enterprises are accelerating the integration of the Internet [1]. The increase in production efficiency has also caused new security issues: the internal network of the smart grid is increasingly connected to the Internet, and the threats and attacks on the smart grid are becoming increasingly serious [2,3]. In order to ensure the security of the smart grid, the SGMS devices will record abnormal data (such as abnormal access data, a large number of DHCP service access), abnormal behavior (such as port scan, illegal access) and abnormal state (such as CPU, memory usage exceeds threshold) as a security event in the form of an alert log. The logs are aggregated and uploaded to the smart grid control center. For example, the log "there is a port scan event that occurs N times from X day to Y day". We can analyze these logs to detect attacks on smart grid. Different from abnormal changes in physical values such as voltage and current [4,5], the alert log data is aggregated generally. They are independent of each other usually and cannot directly represent specific threats and attacks so that it is necessary to adopt an effective security event analysis [6,7] techniques to reveal them.

Due to a mass of security events in the smart grid and possible false positives [6], it is often necessary to detect some critical objects and analyze them to reveal the threats and attacks on the smart grid. The SGMS devices generate a large number of security logs every day, which reflect the security status of the smart grid [7]. However, because the alert threshold of the log is low, many security events with extremely low threat level are reported. As a result, the number of alert logs is extremely large and the probability of false positives is high [7]. On the contrary, security events that deserve to be focused on are hard to find, which poses a huge security risk to the smart grid. Under normal circumstances, the number of the same type of alert will fluctuate within a certain range. Once the number of alerts suddenly exceeds the normal range, that is, the number of alerts surges or exceeds the threshold slowly. It means that an abnormality has occurred, and this is the critical object to be detected that we call a critical event.

Anomaly detection technology is an effective means of the security event analysis. It can dig out events with substantial threats from a mass of security events. At present, the main anomaly detection methods include cluster-based anomaly detection, distance-based anomaly detection and statistical-based anomaly detection. Cluster-based anomaly detection usually uses clustering algorithms, but the number of cluster partitions and the initial centroid are difficult to determine [8]. At the same time, the clustering algorithm divides the data into clusters. In order to obtain accurate results, manual analysis is still needed. The time complexity of distance-based anomaly detection is high [9]. Distance-based anomaly detection requires professional knowledge to set reasonable parameters, but the expertise of smart grid is not easy to obtain, which limits the use of this method. Statistical-based anomaly detection must know the mathematical distribution characteristics of the data in advance, and then use the mathematical model to detect. Otherwise, all the abnormal points are found to be uncertain [10]. The distribution characteristics of the aggregated smart grid alert log are still unclear and need further study.

Focusing on these issues, by analyzing the characteristics of smart grid data, a smart grid critical event detection model based on statistical analysis is proposed, including log preprocessing and critical event detection. The critical event detection consists of a mutation point based detection and threshold based detection. The model first preprocesses the logs, outputs a sequence of security events for each type of alert. And then the detection method based on the mutation point is used to detect critical events in which the number of alerts surges, or the number surges first and continues to fluctuate for a period of time and then sharply decreases. The detection method based on threshold is used to extract critical events throughout the sequence in which the number of alerts exceeds a certain threshold. The main contributions of this paper are as follows:

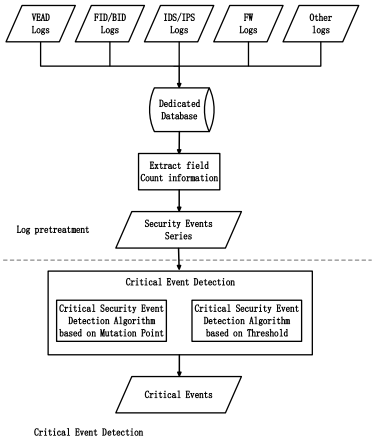


FIGURE I. MODEL FRAMEWORK

- Analyzed the fluctuation characteristics of the smart grid logs, and extracted three types of data: “surge data”, “flat top data” and “slow increase data”.
- The critical event detection algorithm is proposed, which can detect events that surge suddenly, exceed the threshold or show the “flat top data” feature. The algorithm covers multiple forms of abnormal data and reduces the possibility of missing outliers.
- Combined the real smart grid environment data, the algorithm core parameters are determined and the accuracy is calculated. The results show that indicators (e.g. accuracy rate) are above 96%.

II. MODEL FRAMEWORK

Usually there is a process of increasing of the number of visits or requests in cyber attack from the successful invasion to a large-scale outbreak. For example, anomaly detection system (ADS) indicates noticed activities that differ significantly from the recognized normal usage profiles as anomalies [11,12], which can be obtained through traffic or logs. The security devices in the SGMS monitor the system status and generate logs at all times, based on which this paper constructed a smart grid critical event detection model to discover the security events characterized by a surge in the number of alerts. The model includes the log preprocessing and the critical event detection.

The part of log preprocessing mainly collects and extracts logs of various security devices to form a sequence of security events. And the part of critical event detection consists of a critical event detection algorithm based on the mutation point and a critical event detection algorithm based on the threshold, which detects the critical events of different characteristics. The specific process of the model is shown in figure 1.

A. Critical Event Detection Based on Mutation Point

Taking the sequence of security events of an alert as input, it focuses on events with a surge in the number of alerts, events with a sharp decrease in the number of alerts and the flat-top events in which the small range continues to fluctuate and then sharply decreases. The critical events of this alert type is output.

TABLE I. SIGNIFICANCE OF SYMBOLS

	Positive Class	**Negative Class**
Detected	TP (true positives)	FP (false positives)
Undetected	FN (false negatives)	TN (true negatives)

B. Critical Event Detection Based on Threshold

Taking the sequence of security events of an alert as input, it focuses on the events that slowly increase until exceeding a certain threshold rather than sudden increasing. The critical events of this alert type is output.

C. Model Evaluation Index

For a given data set, the positive class represents the sample that should be detected, and the negative class represents the sample that should not be detected. The detected positive class is denoted as TP, the detected negative class is denoted as FP. The undetected positive class is denoted as FN, the undetected negative class is denoted as TN, as shown in Table 1.

Accuracy (ACC): The ratio of the number of samples correctly classified by the algorithm to the total number of samples. ACC = (TP + TN) / (TP + FP + FN + TN). Precision (P): The ratio of the number of correctly detected samples to the total number of samples detected. P = TP / (TP + FP). Recall (R): The ratio of the number of correctly detected samples to the number of samples that should be detected. R = TP / (TP + FN). Comprehensive Evaluation Index (F1-Measure): The harmonic mean of the precision and the recall, usually F1 = 2 * P * R / (P + R).

III. MODEL IMPLEMENTATION

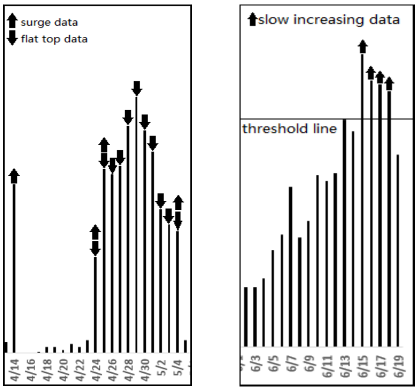
A. Log Preprocessing

The log preprocessing stage was responsible for extracting and counting log information from the database, which would generate a sequence of security events. Firstly, the required fields that constitute the security event were extracted from each valid log, and then the number of alerts was counted to form a sequence of security events, which was finally input into the critical event detection algorithm. The log preprocessing process is illustrated below.

The original log format in table 2 generated by the security devices is:

TABLE II. ORIGINAL LOG FORMAT

Field	Description
<0>	The alert level in this paper is divided into <0><1><2>, and the severity is reduced in turn.
2006-03-12 20:12:23	It is a warning time.
fw01	The device name is "Firewall 01".
FW 0	The device type is "firewall", where "0" in "FW 0" indicates the log subtype.
System EXCEPTION	The system is abnormal and is an important alert.



(a) (b)

FIGURE II. DATA FEATURE

<warning level> <space> warning time <space> device name <space> device type <space> content description. For example, "<0> 2006-03-12 20:12:23 fw01 FW 0 System EXCEPTION".

Combined the log information of multiple devices to perform aggregation to obtain an alert with comprehensive information. For example, "<0> <2018-06-19 03:23:36> <"Device name"> <Illegal login: 1.2.1.62 Illegal login using user itmg 1.1.1.52> <2018-06-19 03:23:36> <1> <Illegal login> <1.2.1.62> <1.1.1.52>", these are WARNINGLEVEL, WARNINGTIME, DEVICENAME, CONTENT, WARNINGSTARTTIME, TIMES, WARNINGTYPE, SOURCEIP and DSTIP. The WARNINGTYPE can be used to obtain the alert type of the security event. The WARNINGSTARTTIME and WARNINGTIME can be used to determine the time when the security event occurs and ends. The TIMES can be used to calculate the number of security events in the set time interval T. Therefore, the sequence of security events $[E, t_1, n_1], [E, t_2, n_2], \cdots, [E, t_n, n_n]$ is available in units of triples $[E, t, n]$ with time span τ. E_i represents an alert type like illegal login. T is the start time of the set time interval T, and n is the number of times that the alert occurred from the start time to the end time of the time T.

B. Critical Event Detection

The critical event detection stage included a multiplication point-based and threshold-based critical event detection algorithm that detected critical events from a sequence of security events. The algorithm assumed that the security event sequence period length is τ, that is, the duration from t_1 to t_n in the sequence of security events $[E, t_1, n_1], [E, t_2, n_2], \cdots, [E, t_n, n_n]$. Each unit duration of the security event is T.

1) Analysis of Data Characteristics

Through the statistics of 8 million logs in a certain area of China Smart Grid for several consecutive days, we found that the smart grid alert data presents two data characteristics.

As shown in figure 2 (a), in the sequence of security events, the number of alerts in a certain period of time T suddenly increased by several times and then sharply decreased, which indicated that the alert occurred too frequently during this period of time T compared to other times, which was very likely that the system was under attack and needed to be monitored. Another situation was that the number of alerts in a certain period of time T suddenly increased sharply, but did not decrease immediately. It fluctuated within a certain range, and then decreased after a period of time. This data was like a "flat top". We called it flat top data and we must monitor it.

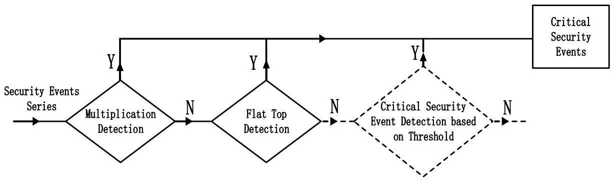


FIGURE III. FLOW CHART OF CRITICAL EVENT DETECTION ALGORITHM BASED ON MUTATION POINT

As shown in figure 2 (b), the number of adjacent security events in the sequence of security events did not surge suddenly but slowly increased during several unit time T. In such cases, when the number of alerts reached a certain value, which indicated that the danger has reached the level that had to be paid attention to, and it needed to be monitored.

2) Critical event detection based on mutation point

Some alerts in the sequence of security events would increase at some time by several times or present the status of flat-top data. At this time, attack behaviors or incorrect operations might occur in the system. In order to extract such anomaly data, a critical event detection algorithm based on the mutation point was designed for the above two data features as shown in figure 3, and determined the value of the relevant parameters on the data of τ = 103 days and T = 1 day.

a) Multiplication point detection: This case applied to two consecutive time T security events $[E, t_i, n_i]$ *,* $[E, t_{i+1}, n_{i+1}]$*,* $n_i * n_{i+1} \neq 0$*. If* n_{i+1} *exceeded or equaled* α *of* n_i *(* α *is a multiplication parameter), then the latter multiplication point* $[E, t_{i+1}, n_{i+1}]$ *was considered to be a critical event that needed to be monitored; In particular, the security events* $[E, t_{i-1}, n_{i-1}]$ *and* $[E, t_{i+1}, n_{i+1}]$ *of two adjacent time Ts of an event* $[E, t_i, n_i]$*,* $n_{i-1} * n_{i+1} = 0$ *and* $n_i \neq 0$*, regardless of whether* n_i *was large enough,* $[E, t_i, n_i]$ *was considered to be a critical event and needed to be monitored.*

Took the DDoS event as an example, detected and analyzed the multiplication parameter α with 1.5, 2.0, and 3.0. As shown in figure 4, the total number of days of DDoS sequence was τ =103 days. Manual review for the positive class of critical events was (TP + FN) = 15.

In table 3, when the multiplication parameter was α = 1.5, 2, 3 samples were detected. The positive class TP = 15 were detected, and the negative class FP = 8 were detected. So the undetected positive class was FN = 15 – 15 = 0, the undetected negative class was TN = 103 – 23 – 0 = 80. The same was true for α = 2.0 and α = 3.0. The detection results were as follows.

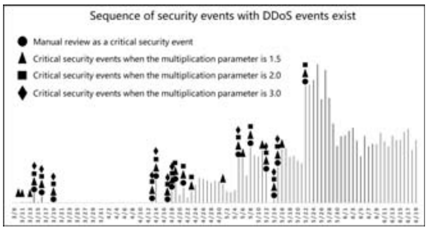


FIGURE IV. MULTIPLICATION DETECTION RESULTS FOR DDOS EVENTS EXIST

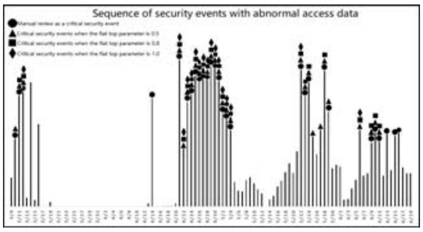


FIGURE V. FLAT-TOP DETECTION RESULTS FOR ANOMALY ACCESS DATA

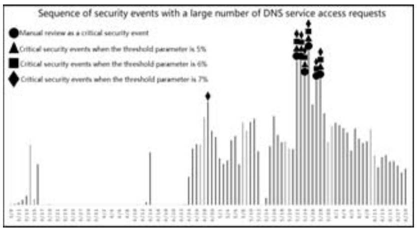


FIGURE VI. THRESHOLD DETECTION RESULTS FOR A LARGE NUMBER OF DNS SERVICE ACCESS REQUESTS

TABLE III. DIFFERENT MULTIPLICATION POINT PARAMETER TRAINING RESULTS

	α = 1.5	α = 2.0	α = 3.0
ACC	92.2%	99.0%	96.1%
P	65.2%	93.8%	100.0%
R	100.0%	100.0%	73.3%
F1	78.8%	96.9%	84.4%

TABLE IV. DIFFERENT FLAT TOP PARAMETERS TRAINING RESULTS

	γ = 0.5	γ = 0.8	γ = 1.0
ACC	92.2%	94.2%	87.4%
P	85.7%	95.8%	94.1%
R	85.7%	82.1%	57.1%
F1	86.0%	88.4%	71.0%

TABLE V. DIFFERENT THRESHOLD PARAMETER TRAINING RESULTS

	θ = 5%	θ = 6%	θ = 7%
ACC	99.0%	100.0%	99.0%
P	100.0%	100.0%	85.7%
R	83.3%	100.0%	100.0%
F1	90.7%	100.0%	92.5%

In summary, when the multiplication parameter was α = 2.0, the ACC and F1 were the highest.

b) Flat top data detection: Observing the entire sequence of security events, it was found that there was such a data feature that a certain security event $[E, t_i, n_i]$ was judged as a multiplication point. After that, the number of subsequent security events did not fall down exponentially, but fluctuated gently. After a few time, there would be a double reduction point $[E, t_j, n_j]$ to fall multiple times, just like a trapezoidal “flat top”, we called the security event between the multiplication point $[E, t_i, n_i]$ and the double reduction point $[E, t_j, n_j]$ “flat top data”. For “flat top data” $[E, t_k, n_k]$, $i < k < j$, if n_k exceeded or equaled the larger of the multiples of the multiplication point and the double reduction point $\max(n_i, n_j)$ γ times (γ is the flat top parameter), it was considered that the $[E, t_k, n_k]$ was a critical event and needed to be monitored.

Took the abnormal access data as an example, the flat top data was detected and analyzed by the flat top parameter γ of 0.5, 0.8, and 1.0. As shown in figure 5, the total number of days of abnormal access data sequence was τ = 103 days. Manual review for the positive class of critical events was (TP + FN) = 28.

In table 4, when the flat top parameter was γ = 0.5, 28 samples were detected. The positive class TP = 24 were detected, and the negative class FP = 4 were detected. So the undetected positive class was FN = 28 – 24 = 4, the undetected negative class was TN = 103 – 28 – 4 = 71. The same was true for γ = 0.8 and γ = 1.0. The detection results were as follows. In summary, when the flat top parameter was γ = 0.8, the ACC and F1 were the highest.

3) Critical event detection based on threshold

Although the number of alerts in the sequence of security events did not show explosive growth, it might reach a very high value in the case of slow growth. In order to solve such problems, a critical event detection algorithm based on threshold was designed as shown in figure 7, and determined the value of the relevant parameters on the data of τ = 103 days and T = 1 day.

Threshold detection: The algorithm first sorted the number of alerts of each security event from high to low, forming a sequence of sequential security events. And then made the data at the front θ position as a threshold (θ is a threshold parameter, which can take 5% of the entire sequence, 6%, etc.). Compared with the number of alerts for each security event in the sequence, if the number of alerts exceeded or equaled the threshold, the security event was considered to be a critical event and needed to be monitored.

Took a large number of DNS (Domain Name Service) service access requests as an example, the threshold parameter θ was 5%, 6%, and 7% to detect and analyze critical events based on threshold. As shown in figure 6, the total number of days of DNS (Domain Name Service) service access requests sequence was τ=103 days. Manual review for the positive class of critical events was (TP + FN) = 6.

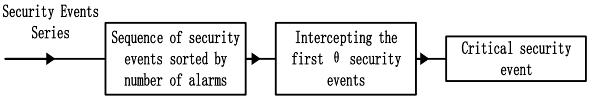

Security Events Series
Sequence of security events sorted by number of alarms
Intercepting the first θ security events
Critical security event

FIGURE VII. FLOW CHART OF CRITICAL EVENT DETECTION ALGORITHM BASED ON THRESHOLD

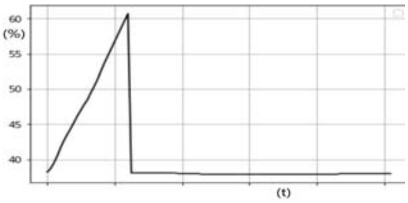


FIGURE VIII. PERCENTAGE OF HARDWARE CONSUMPTION WHILE THE PROGRAM IS RUNNING

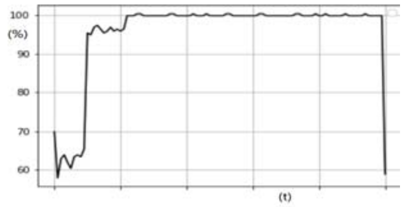


FIGURE IX. PERCENTAGE OF HARDWARE CONSUMPTION WHILE THE PROGRAM IS RUNNING

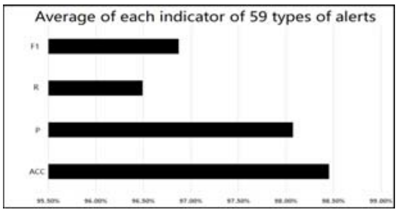


FIGURE X. TEST RESULTS FROM ANOTHER BATCH OF REAL DATA IN THE SMART GRID

In table 5, when the threshold parameter was θ = 5%, 5 samples were detected. The detected positive class was TP = 5, and the detected negative class was FP = 0. So the undetected positive class was FN = 6 – 5 = 1, the undetected negative class was TN = 103 - 5 – 1 = 97. The same was true for θ = 6% and θ = 7%. The detection results were as follows. In summary, when the threshold parameter was θ = 6%, the ACC and the F1 were the highest.

IV. GLOBAL EVALUATION

After parameter training, the multiplication parameter was determined to be α = 2.0, the flat top parameter was determined to be γ = 0.8 and the threshold parameter was determined to be θ = 6%. Conducted overall critical event detection with 59 kinds of alert in a certain area for 23 consecutive days. The security event sequence period length was τ = 23 days.

Host kernel version was Linux version 4.15.0-36-generic, and compiler version was gcc version 5.4.0 20160609. One 4-core CPU and its model was Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz with 16G memory. The running environment of program was Python 3.6.5. In the above test, the total time consumption was 12min35s, the memory usage was shown in figure 8, and the CPU usage was shown in figure 9.

In figure 8, the progressive increase in memory usage was the process of reading data from the database, and then the memory remained basically unchanged. In figure 9, the CPU usage of the algorithm fluctuated at 100%. Finally, each evaluation index was shown in figure 10. It could be seen that the average accuracy of the algorithm was 98.45%, the average precision was 98.07%, the average recall rate was 96.49%, and the average comprehensive evaluation index was 96.87%. Therefore, the model can effectively detect the critical events in the smart grid.

V. CONCLUSION

In this paper, a smart grid critical event detection model based on statistical analysis was proposed. Based on the logs of security devices like intrusion detection system, the data characteristics were analyzed, a critical event detection algorithm based on the mutation point and threshold-based were proposed.

ACKNOWLEDGMENTS

This research was funded Science and Technology Project Funding of State Grid Corporation of China (Research and Application of Network Security Situation Awareness Technology in Power Monitoring System in 2018-2019, Contract no.: 521104180028).

REFERENCES

[1] S. A. Yadav, S. R. Kumar, S. Sharma and A. Singh, "A review of possibilities and solutions of cyber attacks in smart grids," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), Noida, 2016, pp. 60-63.

[2] R. k. Kaur, L. K. Singh and B. Pandey, "Security Analysis of Smart Grids: Successes and Challenges," in IEEE Consumer Electronics Magazine, vol. 8, no. 2, pp. 10-15, March 2019.

[3] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the Smart Grid: A Comprehensive Compilation of Intrusion Detection and Prevention Systems," in IEEE Access, vol. 7, pp. 46595-46620, 2019.

[4] V. Roy, S. S. Noureen, S. B. Bayne, A. Bilbao and M. Giesselmann, "Event Detection From PMU Generated Big Data using R Programming," 2018 IEEE Conference on Technologies for Sustainability (SusTech), Long Beach, CA, USA, 2018, pp. 1-6.

[5] S. J. Matthews and A. St. Leger, "Leveraging MapReduce and Synchrophasors for Real-Time Anomaly Detection in the Smart Grid," in IEEE Transactions on Emerging Topics in Computing, vol. 7, no. 3, pp. 392-403, 1 July-Sept. 2019.

[6] J. Liu, L. Gu, G. Xu and X. Niu, "A correlation analysis method of network security events based on rough set theory," 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content, Beijing, 2012, pp. 517-520.

[7] S. Zhang, Y. Gao, M. Zhang, J. Ge and S. Wang, "The Study of Network Security Event Correlation Analysis Based on Similar Degree of the Attributes," 2013 Fourth International Conference on Digital Manufacturing & Automation, Qingdao, 2013, pp. 1565-1569.

[8] H. Li, "Research and Implementation of an Anomaly Detection Model Based on Clustering Analysis," 2010 International Symposium on Intelligence Information Processing and Trusted Computing, Huanggang, 2010, pp. 458-462.

[9] YANG Jin-wei, "Distance-based and information entropy-based outlier detection over uncertain data," Yunnan University, Yunnan, 2011, pp. 51.

[10] Chuan Zhong, Qiang Gao, Baohong Geng, Fang Yuan, Jian Geng and Wanyuan Li, "Characteristics of big data of power transmission and transformation in smart grid," Proceeding of the 11th World Congress on Intelligent Control and Automation, Shenyang, 2014, pp. 3154-3158.

[11] A.R. Jakhale, "Design of anomaly packet detection framework by data mining algorithm for network flow," 2017 International Conference on Computational Intelligence in Data Science, Chennai, 2017, pp. 1-6.

[12] R. AlShaalan, B. AsSadhan, J. Al-Muhtadi, H. Bin-Abbas, F. A. El-Samie and S. Alshebeili, "Constant false alarm rate anomaly-based approach for network intrusion detection," 2013 High Capacity Optical Networks and Emerging/Enabling Technologies, Magosa, 2013, pp. 141-145.