Mathematical Representation of Netflow Analysis Decision Making Based on Production Logic

Alimdzhan Babadzhanov[1], Inomjon Yarashov[2], Maruf Juraev[3],
Alisher Otakhonov[4], Adilbay Kudaybergenov[5], Rustam Utemuratov[6]
Engineering Federation of Uzbekistan, Tashkent, Uzbekistan[1]
University of World Economy and Diplomacy, Tashkent, Uzbekistan[2]
Tashkent International University of Education, Department of Information Technology, Tashkent, Uzbekistan[2]
Jizzakh Branch of the National University of Uzbekistan named after Mirzo Ulugbek, Jizzakh, Uzbekistan[3]
Fergana State University, Fergana, Uzbekistan[4]
National University of Uzbekistan named after Mirzo Ulugbek, Tashkent, Uzbekistan[5, 6]
Karakalpak State University, Nukus, Uzbekistan[6]

Abstract—**In the sphere of NetFlow traffic analysis, accurate detection of anomalous behavior in real time remains a critical challenge. This study will present a mathematical representation for decision making in NetFlow analysis, production implementation logic for automated expert knowledge. A specialized software system will be developed for collecting and processing NetFlow traffic events in real time cyberspace. NetFlow data will be accumulated in PCAP (Packet Capture) format and converted to .csv using a three-step algorithmic sequence: packet reading, key feature extraction, and output formatting. A total of 89 different packet features will be identified, grouped into 10 categories, including flow statistics, payload features, inter-arrival time, and TCP flags. Based on these features, 110 frequently occurring network processes will be synthesized and identified into seven sets, such as classic cyber threats, application-level cyber threats, anomalies, and normal netflows. Each event will be formally expressed using Boolean production rules in the IF...THEN format, linking subsets of feature vectors (F) to certain network events (A). These production rules form the knowledge base of the expert system, which allows for the efficient identification of cyber threats such as DDoS attacks, port scanning, spoofing, and covert channels. The architecture will ensure systematic analysis for the early detection and identification of netflow anomalies, facilitating to the solidity and protection of complex information systems. The suggested decision-making representation based on production logic will ensure scalable and explainable synthesis and/or analysis, opening the way to the creation of intelligent structures for identifying cyber threats in cyberspace. The accuracy of the results obtained was checked using a confusion matrix.**

Keywords—Production rules; packet features; PCAP; network events; netflow; productiosn logic; confusion matrix

I. Introduction

Real-time monitoring, algorithmic analysis and synthesis of NetFlow traffic in cyberspace is relevant in ensuring cybersecurity and information security [1], [2]. Among the modern technologies of NetFlow Traffic analysis and monitoring, NetFlow monitoring research is known to be fundamental in identifying abnormal states and dealing with cyber threats. It demonstrates the need for flexible formalization in cyberspace [3] and automation to investigate the robustness of decision tree construction and acceptance phenomena in NetFlow traffic analysis and synthesis. Classical statistical techniques and heuristic algorithms [4] are useful in adapting to the increasing algorithmic complexity of contemporary Peer-to-Peer networks in a formalized format. This research work clearly demonstrates the fundamental validity of the decision tree construction in NetFlow monitoring analysis based on production logic [7]. Production logic [5], [6] provides a structural approach to formalizing IF-THEN rules viewed in the form of expert system knowledge, building intelligent systems for identifying abnormalities in real-time [8].

The proposed methodology starts with real-time capture and preprocessing of NetFlow monitoring traffic and research, which is initially stored in PCAP format, and then converted to .csv format using a three-step algorithm: reading the packet capture, extracting and formatting the package capture features. A comprehensive set of 89 package capture features, divided into 10 logical groups (e.g. payload statistics, flow rate, arrival time, TCP flags, etc.) is extracted from the obtained digital data. These package capture features are the fundamental basis for constructing a set of feature vectors F. Based on the obtained package capture features, the research presents 110 common netflow monitoring scenarios (set A is shown), including normal flows, Distributed Denial-of-Service (DDoS) cyberattacks, port scanning in netflow monitoring, remote access attempts, digital data spoofing, and cyber-threats in the environment of covert digital channels. These statuses are based on production logic rules [9], [10], [11], which are combined in a unique combination of features, forming the foundation of the knowledge base of the expert system. These rules simplify the adoption of digital and automated decision trees, supporting the reactive mitigation of abnormal and cyber threats.

The proposed structure with the support of production logic fundamentals provides formalization and algebraic logical representation of decision-making phenomena, localization, realization and implementation suitable support. Optimality [12] and reliability correctness mean that localization is adaptive in systems operating in complex cyberspace. The research is aimed at bridging the gap between real-time NetFlow monitoring research and mental analysis [13], Boolean rule-based cyber threat analysis and synthesis [14], which are improved by fusion of secure, flexible and mental cyber defense infrastructures [15], [16].

Unlike production-based Netflow analysis techniques, network flow analysis using rules allows you to automate the process of identifying and classifying anomalies observed in Netflow flows. The knowledge base built on rules is easy to modify and new rules can be added.

As part of the experimental results, a network flow monitoring system was created in the laboratory environment at the "Academician Vosil Kabulov Digital Technologies and Cybersecurity" Scientific Innovation Center. As a result of the monitoring, more than 10,000 packets were recorded. The recorded packets were saved in the PCAP file format. The packet data in the PCAP file format was formed as a stream based on the developed algorithm and the packet features were extracted. The result of analyzing the PCAP file was 64 flows. Of the flows created for the research work, 45 were marked as normal flows, and the remaining 19 flows were recorded as flows reflecting network events. The network flow created based on the developed production rules was analyzed and the results obtained were checked using a confusion matrix.

II. Related Works

The literature review for the research work is presented in Table I. This is because analyzing the literature review in tabular form is useful in showing the advantages and disadvantages of each literature studied.

Maruf et al. [17] used statistical analysis techniques to analyze other network flows and verified the validity of the statistical analysis technique using a combined markov chain.

Kabulov et al. [18] implemented an algorithmic analysis of threats in information security using functional tables.

Navruzov et al. [19] analyzed the most common network threat, DDOS attack, in network flows. Their approach is based on packet properties.

Batista et al. [20] developed network rules based on the OpenFlow protocol. Based on these rules, they analyzed the relationship between data coming from IoT devices.

Yarashov et al. [21] presented a computerized system for modeling behavior in a production system. This approach is a clear example of working on the basis of production rules in computer systems.

TABLE I. Comparative Analysis of the Suitability of Related Works to the Research Work

№	Source name	Main content	Relevance to the topic
1	Sikos, L. F. Paper [7]	Analysis and synthesis of netflow packets using the Wireshark software tool; support semi-automated decision-making using honeypot technology	Necessary methods for formalizing knowledge bases from NetFlow flows and representing them in production rule systems
2	Sikos, L. F. Paper [22]	Summary of packet capture analysis for Netflow monitoring	Mathematical expression in cyberthreat identification in NetFlow monitoring system
3	Laptiev et al. Paper [23]	Anomalous review of Netflow traffic and monitoring	Production rule is an automatic realization mechanism using conditions for creation
4	Boateng & Rahim Paper [13]	Netflow monitoring using statistical analysis and forecasting (For example Regression models)	Mathematical representation of decision support from NetFlow flow
5	Carlet, C. Paper [24]	Solving problem of cyber security based on boolean function	Mathematical realization of the production rule formalization
6	Yao et al. Paper [25]	Boolean functions to dynamically influence factors and order behaviors	Rules for netflow data through algorithmic classification and mathematical representation
7	Carlet, C. Paper [26]	Distribution properties and characteristics of Boolean functions	Algorithmization and modeling of decision-making events in NetFlow
8	Kellerer et al. Paper [27]	Simulating cyberattacks on Siemens S7 protocol in a virtual environment	Formalization and simulation of real cyberthreat events in the context of NetFlow
9	Ţălu, M. Paper [28]	A key factor in netflow in Kubernetes is the provisioning of DDoS mitigation technologies.	NetFlow monitoring supports DDoS cyberattack analysis and production rule-based response defense mechanisms
10	Bittencourt & Marengoni Paper [29]	Regulatory systems and the instruments that connect their elements	Implementing decision tree rules for NetFlow synthesis with production logic support
11	Rahmatullah et al. Paper [30]	Effectively organize and evaluate the logically connected architecture of a LoRa-MQTT network for IoT	Support for decision rule models in protocol operation in NetFlow research
12	Li & Chen Paper [31]	Detection and analysis of anomalous events using LSTM and wavelet	Supporting AI mathematical approaches to analyzing NetFlow stream data
13	Jing et al. Paper [32]	IoT systems and their connecting elements apply heuristic algorithms to input data	Mechanism for working with features when building production rules
14	Tahboush et al. Paper [33]	Fusion algorithmic technology in the implementation and testing of brute-force cyber-attacks	Mathematical methods in the implementation of Production logic rules based on binary code
15	Guo et al. Paper [34]	Production logic knowledge graphs automatically convert events into theoretical knowledge	Implementation of production rules and conceptualization of the decision-making system

III. The Proposed Methodology

A. Separating Packet Features

Experts use several techniques to analyze netflow traffic. One of the most effective techniques is real-time netflow traffic analysis. In the research work, [35] a real-time program was created to analyze netflow traffic. In laboratory conditions, a netflow traffic was created and the traffic was recorded. The netflow traffic must be recorded in PCAP format and converted to csv format for analysis. For this, an algorithm was developed to convert the recorded data to the required format [28]. The algorithm consists of 3 steps:

1) Reading from PCAP format: When reading packet data, the file must be in PCAP format. Packet feature are used to format packet data as a stream (see Algorithm 1).

Algorithm 1 Reading from PCAP format

```
function nextPacket( )
  set packetInfo ← None
  while pcapPtr < pcapLen do
    call totGen.nextId( ) ▶ Generate a new ID for the packet
    read timeHigh ← unpack 4 bytes from pcapData at offset pcapPtr+4
    read timeLow ← unpack 4 bytes from pcapData at offset pcapPtr +8
    compute timeStamp ← 1,000,000 × timeHigh + timeLow
    read capLen ← unpack 4 bytes from pcapData at offset pcapPtr +8+4
    read capLen ← unpack again to ensure correct length value
    advance pcapPtr ← pcapPtr + 16 ▶ Move pointer to packet body
    extract packetData ← pcapData[pcapPtr : pcapPtr + capLen]
    advance pcapPtr ← pcapPtr + capLen ▶ Move pointer to next packet
    if isIpv4TCP(packetData) then
      set packetInfo ← getIpv4Info(packetData, timeStamp)
      break
    else if isIpv6TCP(packetData) then
      set packetInfo ← getIpv6Info(packetData, timeStamp)
      break
    end if
  end while
  return packetinfo
end function
```

2) Separate packet features: The stream is important in extracting packet features. Because it is important that the data of one stream is not mixed with the data of another stream. Therefore, after extracting the data of one stream, it moves on to the data of the next stream (see Algorithm 2).

Algorithm 2 Separate packet features

```
function init(generator, srcIP, dstIP, srcPort, dstPort, protocol, timeStamp, headBytes, payloadBytes, flags, TCPWindow, payload)
  id ← generator.nextId()▶ Create a unique ID for the packet
  srcIP ← srcIP ▶ Setting the source IP address
  dstIP ← dstIP ▶ Setting the destination IP address
  srcPort ← srcPort ▶ Setting the source port
  dstPort ← dstPort Setting the destination port
  protocol ← protocol ▶ Protocol setup
  timeStamp ← timeStamp ▶ Set timestamp
  headBytes ← headBytes ▶ Setting the packet header length
  payloadBytes ← payloadBytes ▶Load length setting
                    .   .   .
  return (flags >> 5) & 1 ▶ Check the URG flag
end function
function hasFlagECE()
  return (flags >> 6) & 1 ▶ Check the ECE flag
end function
function hasFlagCWR()
  return (flags >> 7) & 1 ▶ Check the CWR flag
end function
```

3) Save in csv format: After separating the flow features by packet, they are saved in csv format.

Algorithm 3 Save in csv format

```
function addpacket(packet: basicpacketinfo)
  if packet is none then
    return
  end if
  set currentts ← packet.gettimestamp() ▶ get packet timestamp
  set pktfwdflowid ← packet.getfwdflowid() ▶ get forward flow id
  set pktbwdflowid ← packet.getbwdflowid() ▶ get backward flow id
  if pktfwdflowid in currentflows or pktbwdflowid in currentflows then
    if pktfwdflowid in currentflows then
      set flowid ← pktfwdflowid
                    .   .   .
function dumppayloadtocsv()
  open "payload.csv" for writing as csvfile
  set writer ← csv.writer(csvfile) ▶ initialize csv writer
  for each flow in finishedflows do
    set flow ← flowtype(flow) ▶ resolve the actual flow
    set output ← flow.generateflowfeatures() ▶ extract features
    writer.writerow(output) ▶ write features to csv
  end for
end function
```

Fig. 1 shows the process from capture to conversion to the desired format. Production rules are used to analyze the packet features stored in csv format.

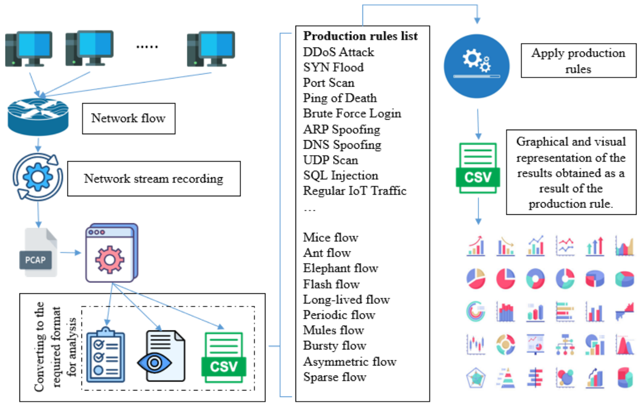

Fig. 1. Netflow recording and production rule-based analysis.

B. Classify Packet Features

In the research work, the 11 most important packet information was used to extract packet features. srcIP, dstIP, srcPort, dstPort, protocol, timestamp, headBytes, payloadBytes, flags, windowSize and payload are the main information for forming the flow data [17]. Based on this information, 89 different features were extracted. Table II lists the extracted features [36].

In reality, the number of package features allocated could be more than 89, but for the purposes of this research, 89 features were considered sufficient. The package features were divided into 10 groups:

- Identification features;
- Flow statistics;
- Payload statistics;
- Header statistics;
- Rates and Ratios (flow rate and direction ratio);
- Inter-Arrival Time (Inter-Arrival Time);
- Flags (TCP flags);
- Initial Window Size;
- Subflow statistics;
- Bulk statistics.

Based on the extracted features, netflow analysis is performed.

TABLE II. PACKET FEATURES

№	Feature name	Marking
1	Flow ID	f_1
2	Src IP	f_2
3	Src Port	f_3
4	Dst IP	f_4
5	Dst Port	f_5
6	Protocol	f_6
7	Flow Pkt Num	f_7
8	Flow Pld Byte Sum	f_8
9	Flow Pld Byte Max	f_9
10	Flow Pld Byte Min	f_{10}
11	Flow Pld Byte Mean	f_{11}
12	Flow Pld Byte Std	f_{12}
13	Fwd Pkt Num	f_{13}
14	Fwd Pld Byte Sum	f_{14}
15	Fwd Pld Byte Max	f_{15}
16	Fwd Pld Byte Min	f_{16}
17	Fwd Pld Byte Mean	f_{17}
18	Fwd Pld Byte Std	f_{18}
19	Bwd Pkt Num	f_{19}
20	Bwd Pld Byte Sum	f_{20}
21	Bwd Pld Byte Max	f_{21}
22	Bwd Pld Byte Min	f_{22}
23	Bwd Pld Byte Mean	f_{23}
24	Bwd Pld Byte Std	f_{24}
25	Fwd Head Byte Max	f_{25}
26	Fwd Head Byte Min	f_{26}
27	Fwd Head Byte Mean	f_{27}
28	Fwd Head Byte Std	f_{28}
29	Bwd Head Byte Max	f_{29}
30	Bwd Head Byte Min	f_{30}
31	Bwd Head Byte Mean	f_{31}
32	Bwd Head Byte Std	f_{32}
33	Flow Duration(ms)	f_{33}
34	Flow Pkts/s	f_{34}
35	Flow Pld Bytes/ms	f_{35}
36	Fwd Pkts/s	f_{36}
37	Fwd Pld Bytes/ms	f_{37}
38	Bwd Pkts/s	f_{38}
39	Bwd Pld Bytes/ms	f_{39}
40	Pkts Ratio	f_{40}
41	Bytes Ratio	f_{41}
42	Flow IAT Max	f_{42}
43	Flow IAT Min	f_{43}
44	Flow IAT Mean	f_{44}
45	Flow IAT Std	f_{45}
46	Fwd IAT Max	f_{46}
47	Fwd IAT Min	f_{47}
48	Fwd IAT Mean	f_{48}
49	Fwd IAT Std	f_{49}
50	Bwd IAT Max	f_{50}
51	Bwd IAT Min	f_{51}
52	Bwd IAT Mean	f_{52}
53	Bwd IAT Std	f_{53}
54	FIN Count	f_{54}
55	SYN Count	f_{55}
56	RST Count	f_{56}
57	PSH Count	f_{57}

58	ACK Count	f_{58}
59	URG Count	f_{59}
60	ECE Count	f_{60}
61	CWR Count	f_{61}
62	Fwd PSH Count	f_{62}
63	Bwd PSH Count	f_{63}
64	Fwd URG Count	f_{64}
65	Bwd URG Count	f_{65}
66	Fwd Init Win Bytes	f_{66}
67	Bwd Init Win Bytes	f_{67}
68	Fwd Pkts With Pld	f_{68}
69	Bwd Pkts With Pld	f_{69}
70	Sub Flow Fwd Pkts	f_{70}
71	Sub Flow Fwd Bytes	f_{71}
72	Sub Flow Bwd Pkts	f_{72}
73	Sub Flow Bwd Bytes	f_{73}
74	Flow Act Sum	f_{74}
75	Flow Act Max	f_{75}
76	Flow Act Min	f_{76}
77	Flow Act Mean	f_{77}
78	Flow Act Std	f_{78}
79	Flow Idle Sum	f_{79}
80	Flow Idle Max	f_{80}
81	Flow Idle Min	f_{81}
82	Flow Idle Mean	f_{82}
83	Flow Idle Std	f_{83}
84	Fwd Avg Pkts/Bulk	f_{84}
85	Fwd Avg Bytes/Bulk	f_{85}
86	Fwd Avg Bulk/s	f_{86}
87	Bwd Avg Pkts/Bulk	f_{87}
88	Bwd Avg Bytes/Bulk	f_{88}
89	Bwd Avg Bulk/s	f_{89}

A list of 89 features was introduced, defining them as a set F.

$$F = \{f_1, f_2, \dots, f_{89}\} \quad (1)$$

The recorded netflow consisted of 64 packet streams. In fact, the number of recorded packets was more, but when combined into a stream, it amounted to 64. After extracting the data recorded from the netflow into packet features, [37] it is checked based on the production rules created to check the network status. The results can be displayed visually and graphically to make the analysis process understandable.

C. Analysis of Network Events

By recording the netflow, packet features were extracted. From these characteristics, the most common and common situations that occur in the network were identified. Fig. 2 shows the situations that occur in the network, divided into seven groups:

- Classic threats;
- Application/Protocol threats;
- Transport - Level threats;
- Hidden/Anomalous traffic;
- Network Anomalous;
- Normal and beneficial flows;
- Flow type.

Table III lists the 110 most common network conditions based on seven groups. This means that the 110 most common network conditions are identified by the characteristics of the extracted packets. The first factor in forming the list of selected threats was the most common, although there may be more such conditions [32]. The second factor in forming the list in Table III was those that can be identified by the characteristics of the packets. Early detection of conditions in the network is always important, which prevents network performance from being disrupted. The characteristics of the extracted packets are used to identify the selected conditions.

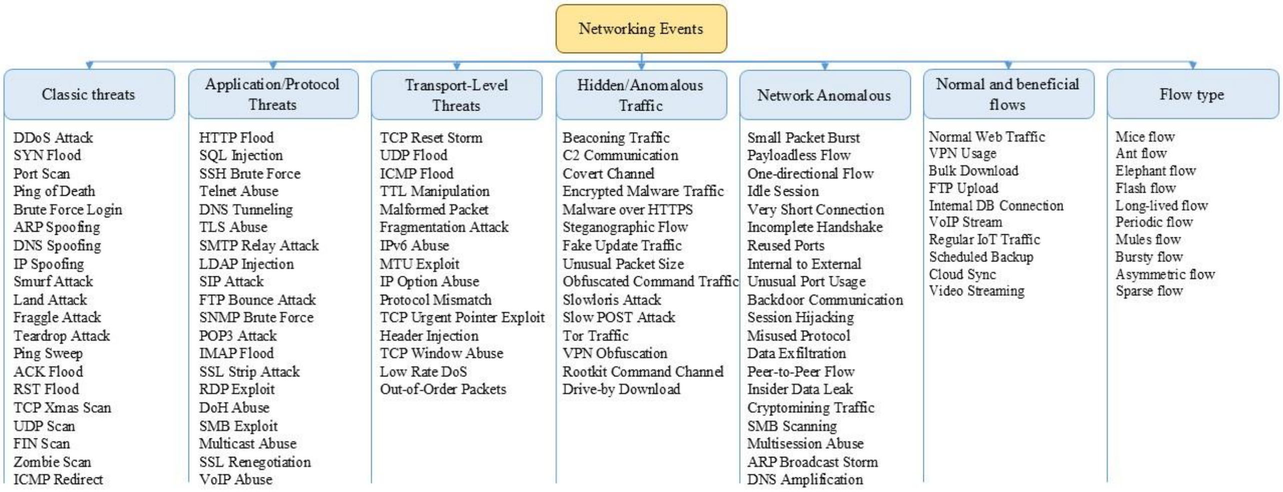

Fig. 2. Grouping of events in the network.

There are various techniques for detecting threats based on package properties. These include mathematical analysis, statistical analysis, visualization, production logic, fuzzy logic, etc [38]. The research used the production logic technique. Production rules serve as a knowledge base for expert systems [39].

TABLE III. THE MOST COMMON NETWORK EVENTS

№	Status name	Marking
1	DDoS Attack	a_1
2	SYN Flood	a_2
3	Port Scan	a_3
4	Ping of Death	a_4
5	Brute Force Login	a_5
6	ARP Spoofing	a_6
7	DNS Spoofing	a_7
8	IP Spoofing	a_8
9	Smurf Attack	a_9
10	Land Attack	a_{10}
11	Fraggle Attack	a_{11}
12	Teardrop Attack	a_{12}
13	Ping Sweep	a_{13}
14	ACK Flood	a_{14}
15	RST Flood	a_{15}
16	TCP Xmas Scan	a_{16}
17	UDP Scan	a_{17}
18	FIN Scan	a_{18}
19	Zombie Scan	a_{19}
20	ICMP Redirect	a_{20}
21	HTTP Flood	a_{21}
22	SQL Injection	a_{22}
23	SSH Brute Force	a_{23}
24	Telnet Abuse	a_{24}
25	DNS Tunneling	a_{25}
26	TLS Abuse	a_{26}
27	SMTP Relay Attack	a_{27}
28	LDAP Injection	a_{28}
29	SIP Attack	a_{29}
30	FTP Bounce Attack	a_{30}
31	SNMP Brute Force	a_{31}
32	POP3 Attack	a_{32}
33	IMAP Flood	a_{33}
34	SSL Strip Attack	a_{34}
35	RDP Exploit	a_{35}
36	DoH Abuse	a_{36}
37	SMB Exploit	a_{37}
38	Multicast Abuse	a_{38}
39	SSL Renegotiation	a_{39}
40	VoIP Abuse	a_{40}
41	TCP Reset Storm	a_{41}
42	UDP Flood	a_{42}
43	ICMP Flood	a_{43}
44	TTL Manipulation	a_{44}
45	Malformed Packet	a_{45}
46	Fragmentation Attack	a_{46}
47	IPv6 Abuse	a_{47}
48	MTU Exploit	a_{48}
49	IP Option Abuse	a_{49}
50	Protocol Mismatch	a_{50}
51	TCP Urgent Pointer Exploit	a_{51}
52	Header Injection	a_{52}
53	TCP Window Abuse	a_{53}
54	Low Rate DoS	a_{54}
55	Out-of-Order Packets	a_{55}
56	Beaconing Traffic	a_{56}
57	C2 Communication	a_{57}
58	Covert Channel	a_{58}
59	Encrypted Malware Traffic	a_{59}
60	Malware over HTTPS	a_{60}
61	Steganographic Flow	a_{61}
62	Fake Update Traffic	a_{62}
63	Unusual Packet Size	a_{63}
64	Obfuscated Command Traffic	a_{64}
65	Slowloris Attack	a_{65}
66	Slow POST Attack	a_{66}
67	Tor Traffic	a_{67}
68	VPN Obfuscation	a_{68}
69	Rootkit Command Channel	a_{69}
70	Drive-by Download	a_{70}
71	Small Packet Burst	a_{71}
72	Payloadless Flow	a_{72}
73	One-directional Flow	a_{73}
74	Idle Session	a_{74}
75	Very Short Connection	a_{75}
76	Incomplete Handshake	a_{76}
77	Reused Ports	a_{77}
78	Internal to External	a_{78}
79	Unusual Port Usage	a_{79}
80	Backdoor Communication	a_{80}
81	Session Hijacking	a_{81}
82	Misused Protocol	a_{82}
83	Data Exfiltration	a_{83}
84	Peer-to-Peer Flow	a_{84}
85	Insider Data Leak	a_{85}
86	Cryptomining Traffic	a_{86}
87	SMB Scanning	a_{87}
88	Multisession Abuse	a_{88}
89	ARP Broadcast Storm	a_{89}
90	DNS Amplification	a_{90}
91	Normal Web Traffic	a_{91}
92	VPN Usage	a_{92}

93	Bulk Download	a_{93}
94	FTP Upload	a_{94}
95	Internal DB Connection	a_{95}
96	VoIP Stream	a_{96}
97	Regular IoT Traffic	a_{97}
98	Scheduled Backup	a_{98}
99	Cloud Sync	a_{99}
100	Video Streaming	a_{100}
101	Mice flow	a_{101}
102	Ant flow	a_{102}
103	Elephant flow	a_{103}
104	Flash flow	a_{104}
105	Long-lived flow	a_{105}
106	Periodic flow	a_{106}
107	Mules flow	a_{107}
108	Bursty flow	a_{108}
109	Asymmetric flow	a_{109}
110	Sparse flow	a_{110}

The network events listed in Table III have been designated as set A.

$$A = \{a_1, a_2, \dots, a_{110}\} \qquad (2)$$

The situations recorded in the network status collection are the most frequently detected by network specialists. Network professionals will certainly try to avoid [40] these situations when monitoring network activity and use various software tools for monitoring.

IV. Experimental Results and Discussions

To achieve the goal, it is necessary to automate decision-making using an expert system to analyze netflow traffic and identify faults in it. Based on the netflow traffic, production rules are created for expert systems using the recorded packet features and network events.

A. Formalizing the Mathematical Representation of Production Rules

Production rules are considered as a knowledge base for expert systems. The packet features given in set F are used to identify the 110 network events given in set A. As a result, 110 production rules were created.

Formula 3 is used to create production rules.

$$IF \dots, THEN \dots \qquad (3)$$

To put it into mathematical representation, the production rules were written in the form of Boolean rules.

$$p_1: f_{34} \wedge f_{35} \wedge f_{45} \wedge f_{55} \wedge f_{58} \wedge f_{68} \rightarrow a_1 \mid f_{34} > 5000, f_{35} > 10, f_{45} < 1, f_{55} > 100, f_{58} < 10, f_{68} > 50$$

$$p_2: f_{36} \wedge f_{45} \wedge f_{55} \wedge f_{58} \wedge f_{68} \rightarrow a_2 \mid f_{36} > 1000, f_{45} < 1, f_{55} > 100, f_{58} < 5, f_{68} > 30$$

$$p_3: f_7 \wedge f_{33} \wedge f_{55} \rightarrow a_3 \mid f_7 < 5, f_{33} < 200, f_{55} > 20$$

$$p_4: f_6 \wedge f_8 \wedge f_{27} \wedge f_{45} \rightarrow a_4 \mid f_6 = ICMP, f_8 > 10000, f_{27} > 1500, f_{45} > 2$$

$$p_5: f_{36} \wedge f_{55} \wedge f_{58} \rightarrow a_5 \mid f_{36} > 500, f_{55} > 50, f_{85} > 50$$

$$p_6: f_6 \wedge f_7 \wedge f_8 \wedge f_{42} \wedge f_{45} \rightarrow a_6 \mid f_6 = ARP, f_7 = 1, f_8 < 100, f_{42} > 5, f_{45} < 1$$

$$p_7: f_6 \wedge f_7 \wedge f_8 \wedge f_{27} \wedge f_{42} \wedge f_{74} \rightarrow a_7 \mid f_6 = UDP, f_7 < 3, f_8 < 300, f_{27} > 100, f_{42} > 3, f_{74} = 0$$

$$p_8: f_7 \wedge f_8 \wedge f_{42} \wedge f_{55} \wedge f_{58} \wedge f_{74} \rightarrow a_8 \mid f_7 > 50, f_8 > 5000, f_{42} > 20, f_{55} > 50, f_{58} = 0, f_{74} = 0$$

$$p_9: f_6 \wedge f_7 \wedge f_8 \wedge f_{42} \wedge f_{45} \rightarrow a_9 \mid f_6 = ICMP, f_7 > 100, f_8 > 10000, f_{42} > 10, f_{45} < 1$$

$$p_{10}: f_2 \wedge f_3 \wedge f_6 \wedge f_{55} \rightarrow a_{10} \mid f_2 = f_4, f_3 = f_5, f_6 = TCP, f_{55} > 0$$

$$p_{11}: f_6 \wedge f_7 \wedge f_8 \wedge f_{42} \rightarrow a_{11} \mid f_6 = UDP, f_7 > 100, f_8 > 10000, f_{42} > 10$$

$$p_{12}: f_6 \wedge f_8 \wedge f_{12} \wedge f_{27} \rightarrow a_{12} \mid f_6 = TCP, f_8 < 400, f_{12} > 50, f_{27} < 20$$

$$p_{13}: f_6 \wedge f_7 \wedge f_{33} \rightarrow a_{13} \mid f_6 = ICMP, f_7 < 10, f_{33} < 500$$

$$p_{14}: f_{36} \wedge f_{55} \wedge f_{58} \rightarrow a_{14} \mid f_{36} > 500, f_{55} < 10, f_{58} > 50$$

$$p_{15}: f_{36} \wedge f_{56} \wedge f_{58} \rightarrow a_{15} \mid f_{36} > 200, f_{56} > 100, f_{58} < 5$$

$$p_{16}: f_6 \wedge f_{55} \wedge f_{57} \wedge f_{58} \wedge f_{59} \rightarrow a_{16} \mid f_6 = TCP, f_{55} > 0, f_{57} > 0, f_{58} = 0, f_{59} > 0$$

$$p_{17}: f_6 \wedge f_7 \wedge f_8 \wedge f_{36} \rightarrow a_{17} \mid f_6 = UDP, f_7 > 50, f_8 > 5000, f_{36} > 200$$

$$p_{18}: f_6 \wedge f_{36} \wedge f_{55} \wedge f_{56} \wedge f_{58} \wedge f_{59} \rightarrow a_{18} \mid f_6 = TCP, f_{36} > 100, f_{55} = 0, f_{56} = 0, f_{58} = 0, f_{59} = 0$$

$$p_{19}: f_6 \wedge f_7 \wedge f_{33} \wedge f_{58} \rightarrow a_{19} \mid f_6 = TCP, f_7 = 1, f_{33} < 100, f_{58} = 0$$

$$p_{20}: f_6 \wedge f_8 \wedge f_{33} \wedge f_{42} \rightarrow a_{20} \mid f_6 = ICMP, f_8 < 200, f_{33} < 300, f_{42} > 5$$

$$p_{21}: f_5 \wedge f_6 \wedge f_{35} \wedge f_{36} \wedge f_{45} \rightarrow a_{21} \mid f_5 = 80, f_6 = TCP, f_{35} > 10, f_{36} > 1000, f_{45} < 1$$

$$p_{22}: f_5 \wedge f_8 \wedge f_{45} \wedge f_{58} \wedge f_{68} \rightarrow a_{22} \mid f_5 = 80, f_8 > 5000, f_{45} < 1, f_{58} > 50, f_{68} > 50$$

$$p_{23}: f_5 \wedge f_{36} \wedge f_{55} \wedge f_{58} \rightarrow a_{23} \mid f_5 = 22, f_{36} > 500, f_{55} > 50, f_{58} < 5$$

$$p_{24}: f_5 \wedge f_{36} \wedge f_{55} \wedge f_{58} \rightarrow a_{24} \mid f_5 = 23, f_{36} > 200, f_{55} > 30, f_{58} < 10$$

$$p_{25}: f_5 \wedge f_6 \wedge f_8 \wedge f_{27} \rightarrow a_{25} \mid f_5 = 53, f_6 = UDP, f_8 > 2000, f_{27} < 50$$

$$p_{26}: f_5 \wedge f_6 \wedge f_{36} \wedge f_{75} \rightarrow a_{26} \mid f_5 = 443, f_6 = TCP, f_{36} > 300, f_{75} > 1$$

$$p_{27}: f_5 \wedge f_{36} \wedge f_{45} \wedge f_{68} \rightarrow a_{27} \mid f_5 = 25, f_{36} > 100, f_{45} < 2, f_{68} > 20$$

$p_{28}: f_5 \wedge f_8 \wedge f_{36} \wedge f_{58} \rightarrow a_{28} \mid f_5 = 389, f_8 > 3000, f_{36} > 200, f_{58} < 5$

$p_{29}: f_5 \wedge f_7 \wedge f_{36} \rightarrow a_{29} \mid f_5 = 5060, f_7 > 50, f_{36} > 300$

$p_{30}: f_5 \wedge f_7 \wedge f_8 \wedge f_{36} \rightarrow a_{30} \mid f_5 = 21, f_7 > 30, f_8 > 10000, f_{36} > 200$

$p_{31}: f_5 \wedge f_8 \wedge f_{36} \wedge f_{55} \rightarrow a_{31} \mid f_5 = 161, f_8 < 500, f_{36} > 500, f_{55} > 55$

$p_{32}: f_5 \wedge f_{35} \wedge f_{45} \wedge f_{68} \rightarrow a_{32} \mid f_5 = 110, f_{35} > 5, f_{45} < 1, f_{68} > 30$

$p_{33}: f_5 \wedge f_8 \wedge f_{36} \wedge f_{58} \rightarrow a_{33} \mid f_5 = 143, f_8 > 8000, f_{36} > 300, f_{58} < 5$

$p_{34}: f_5 \wedge f_6 \wedge f_{58} \wedge f_{66} \rightarrow a_{34} \mid f_5 = 443, f_6 = TCP, f_{58} > 30, f_{66} < 100$

$p_{35}: f_5 \wedge f_6 \wedge f_{58} \wedge f_{66} \rightarrow a_{35} \mid f_5 = 443, f_6 = TCP, f_{58} > 30, f_{66} < 100$

$p_{36}: f_5 \wedge f_8 \wedge f_{36} \wedge f_{58} \rightarrow a_{36} \mid f_5 = 3389, f_8 > 10000, f_{36} > 500, f_{58} < 5$

$p_{37}: f_5 \wedge f_8 \wedge f_{36} \wedge f_{75} \rightarrow a_{37} \mid f_5 = 443, f_8 > 5000, f_{36} > 200, f_{75} > 1$

$p_{38}: f_4 \wedge f_6 \wedge f_7 \wedge f_{35} \rightarrow a_{38} \mid f_4 = Multicast, f_6 = UDP, f_7 > 100, f_{35} > 5$

$p_{39}: f_5 \wedge f_6 \wedge f_8 \wedge f_{75} \rightarrow a_{39} \mid f_5 = 443, f_6 = TCP, f_8 > 5000, f_{75} > 2$

$p_{40}: f_5 \wedge f_7 \wedge f_{36} \wedge f_{68} \rightarrow a_{40} \mid f_5 = 5060, f_7 > 100, f_{36} > 300, f_{68} > 80$

$p_{41}: f_{36} \wedge f_{56} \wedge f_{58} \wedge f_{74} \rightarrow a_{41} \mid f_{36} > 500, f_{56} > 50, f_{58} < 5, f_{74} = 0$

$p_{42}: f_6 \wedge f_7 \wedge f_8 \wedge f_{42} \wedge f_{45} \rightarrow a_{42} \mid f_6 = UDP, f_7 > 100, f_8 > 8000, f_{42} < 5, f_{45} < 2$

$p_{43}: f_6 \wedge f_7 \wedge f_8 \wedge f_{45} \rightarrow a_{43} \mid f_6 = ICMP, f_7 > 100, f_8 > 10000, f_{45} < 2$

$p_{44}: f_5 \wedge f_6 \wedge f_{27} \wedge f_{42} \rightarrow a_{44} \mid f_5 \neq 0, f_6 = IP, f_{27} > 100, f_{42} < 1$

$p_{45}: f_9 \wedge f_{10} \wedge f_{12} \wedge f_{24} \rightarrow a_{45} \mid f_9 < 20, f_{10} = 0, f_{12} > 50, f_{24} > 30$

$p_{46}: f_5 \wedge f_{12} \wedge f_{18} \wedge f_{33} \rightarrow a_{46} \mid f_5 \neq 0, f_{12} > 70, f_{18} > 30, f_{33} < 200$

$p_{47}: f_6 \wedge f_8 \wedge f_{45} \wedge f_{58} \rightarrow a_{47} \mid f_6 = IPv6, f_8 > 4000, f_{45} > 2, f_{58} < 5$

$p_{48}: f_8 \wedge f_9 \wedge f_{10} \wedge f_{33} \rightarrow a_{48} \mid f_8 > 1400, f_9 > 1300, f_{10} > 10, f_{33} < 300$

$p_{49}: f_6 \wedge f_{25} \wedge f_{27} \wedge f_{74} \rightarrow a_{49} \mid f_6 = IP, f_{25} > 100, f_{27} < 20, f_{74} = 0$

$p_{50}: f_5 \wedge f_6 \wedge f_{42} \wedge f_{58} \rightarrow a_{50} \mid f_5 \neq 25\ f_6 = UDP, f_{42} > 3, f_{58} < 5$

$p_{51}: f_{33} \wedge f_{59} \wedge f_{64} \wedge f_{66} \rightarrow a_{51} \mid f_{33} < 200, f_{59} > 0, f_{64} > 0, f_{66} < 100$

$p_{52}: f_6 \wedge f_{25} \wedge f_{33} \wedge f_{58} \rightarrow a_{52} \mid f_6 = TCP, f_{25} > 100, f_{33} < 300, f_{58} < 10$

$p_{53}: f_{27} \wedge f_{33} \wedge f_{66} \rightarrow a_{53} \mid f_{27} > 1000, f_{33} < 300, f_{66} < 100$

$p_{54}: f_7 \wedge f_{34} \wedge f_{74} \rightarrow a_{54} \mid f_7 > 50, f_{34} < 2, f_{74} > 1000$

$p_{55}: f_{42} \wedge f_{49} \rightarrow a_{55} \mid f_{42} < f_{43}, f_{49} > 50$

$p_{56}: f_{33} \wedge f_{44} \wedge f_{77} \rightarrow a_{56} \mid f_{33} > 60000, f_{44} < 5, f_{77} < 2$

$p_{57}: f_5 \wedge f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{57} \mid f_5 \neq 80, f_7 < 10, f_8 < 1000, f_{33} > 30000$

$p_{58}: f_5 \wedge f_8 \wedge f_{48} \wedge f_{74} \rightarrow a_{58} \mid f_5 \neq 80, f_8 < 500, f_{48} < 3, f_{74} > 1000$

$p_{59}: f_5 \wedge f_6 \wedge f_8 \wedge f_{68} \rightarrow a_{59} \mid f_5 = 443, f_6 = TCP, f_8 > 5000, f_{68} > 20$

$p_{60}: f_5 \wedge f_{68} \wedge f_{74} \rightarrow a_{60} \mid f_5 = 443, f_{68} > 30, f_{74} > 5000$

$p_{61}: f_8 \wedge f_9 \wedge f_{45} \wedge f_{74} \rightarrow a_{61} \mid f_8 < 800, f_9 < 100, f_{45} > 20 f_{74} > 1000$

$p_{62}: f_5 \wedge f_8 \wedge f_{74} \wedge f_{77} \rightarrow a_{62} \mid f_5 = 80, f_8 < 500, f_{74} < 200, f_{77} < 2$

$p_{63}: f_9 \wedge f_{10} \wedge f_{12} \rightarrow a_{63} \mid f_9 > 1400, f_{10} < 50, f_{12} > 200$

$p_{64}: f_6 \wedge f_8 \wedge f_{58} \wedge f_{74} \rightarrow a_{64} \mid f_6 = TCP, f_8 < 400, f_{58} < 3, f_{74} > 2000$

$p_{65}: f_{33} \wedge f_{58} \wedge f_{68} \wedge f_{69} \rightarrow a_{65} \mid f_{33} > 10000, f_{58} < 3, f_{68} = 0, f_{69} = 0$

$p_{66}: f_7 \wedge f_{58} \wedge f_{68} \rightarrow a_{66} \mid f_7 > 50, f_{58} < 3, f_{68} = 0$

$p_{67}: f_5 \wedge f_8 \wedge f_{44} \rightarrow a_{67} \mid f_5 \in \{9001; 443\}, f_8 > 2000, f_{44} > 10$

$p_{68}: f_5 \wedge f_6 \wedge f_8 \wedge f_{34} \rightarrow a_{68} \mid f_5 \in \{1194; 443\}, f_6 = TCP, f_8 > 4000, f_{34} < 10$

$p_{69}: f_6 \wedge f_8 \wedge f_{58} \wedge f_{74} \rightarrow a_{69} \mid f_6 = TCP, f_8 < 800, f_{58} < 2, f_{74} > 3000$

$p_{70}: f_5 \wedge f_8 \wedge f_{74} \wedge f_{77} \rightarrow a_{70} \mid f_5 = 80, f_8 < 2000, f_{74} < 1000, f_{77} < 1$

$p_{71}: f_7 \wedge f_9 \wedge f_{33} \rightarrow a_{71} \mid f_7 > 30, f_9 < 100, f_{33} < 1000$

$p_{72}: f_8 \wedge f_{68} \wedge f_{69} \rightarrow a_{72} \mid f_8 = 0, f_{68} = 0, f_{69} = 0$

$p_{73}: f_{13} \vee f_{19} \rightarrow a_{73} \mid f_{13} = 0, f_{19} = 0$

$p_{74}: f_8 \wedge f_{33} \wedge f_{74} \rightarrow a_{74} \mid f_8 < 500, f_{33} > 300000, f_{74} < 50$

$p_{75}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{75} \mid f_7 < 5, f_8 < 200, f_{33} < 200$

$p_{76}: f_{54} \wedge f_{55} \wedge f_{58} \rightarrow a_{76} \mid f_{54} = 0, f_{55} = 1, f_{58} = 0$

$p_{77}: f_5 \wedge f_8 \wedge f_{66} \rightarrow a_{77} \mid f_5 = f_3, f_8 > 0, f_{66} < 500$

$p_{78}: f_2 \wedge f_4 \rightarrow a_{78} \mid f_2 = 19.168.\%, f_4 \neq 192.168.\%$

$$p_{79}: f_5 \wedge f_8 \rightarrow a_{79} \mid f_5 \notin \{80,443,53,21,22,25\}, f_8 > 1000$$

$$p_{80}: f_5 \wedge f_8 \wedge f_{33} \wedge f_{36} \wedge f_{66} \rightarrow a_{80} \mid f_5 \in \{4444; 1337\}, f_8 > 0, f_{33} > 10000, f_{36} < 1, f_{66} < 1024$$

$$p_{81}: f_8 \wedge f_{33} \wedge f_{55} \wedge f_{56} \wedge f_{58} \rightarrow a_{81} \mid f_8 > 500, f_{33} < 500, f_{55} = 1, f_{56} = 1, f_{58} = 1$$

$$p_{82}: f_6 \wedge f_8 \rightarrow a_{82} \mid f_6 \notin \{1; 6; 17\}, f_8 > 1000$$

$$p_{83}: f_2 \wedge f_4 \wedge f_8 \wedge f_{36} \rightarrow a_{83} \mid f_2 = 192.168.\%, f_4 \neq 192.168.\%, f_8 > 5000, f_{36} > 5$$

$$p_{84}: f_5 \wedge f_8 \wedge f_{36} \rightarrow a_{84} \mid f_5 \in \{6881; 51413; 135\}, f_8 > 2000, f_{36} > 10$$

$$p_{85}: f_2 \wedge f_8 \wedge f_{33} \rightarrow a_{85} \mid f_2 = f_4, f_8 > 2000, f_{33} < 1000$$

$$p_{86}: f_5 \wedge f_6 \wedge f_8 \wedge f_{34} \wedge f_{66} \rightarrow a_{86} \mid f_5 = 3333, f_6 = 6, f_8 > 5000, f_{34} > 10, f_{66} > 2000$$

$$p_{87}: f_5 \wedge f_{36} \wedge f_{68} \rightarrow a_{87} \mid f_5 = 445, f_{36} > 5, f_{68} > 10$$

$$p_{88}: f_{33} \wedge f_{70} \wedge f_{72} \rightarrow a_{88} \mid f_{33} < 300, f_{70} > 10, f_{72} > 10$$

$$p_{89}: f_6 \wedge f_{33} \wedge f_{34} \rightarrow a_{89} \mid f_6 = 1, f_{33} < 100, f_{34} > 50$$

$$p_{90}: f_5 \wedge f_8 \wedge f_{20} \wedge f_{38} \rightarrow a_{90} \mid f_5 = 53, f_8 > 512, f_{20} > 2000, f_{38} > 3$$

$$p_{91}: f_5 \wedge f_8 \wedge f_{36} \rightarrow a_{91} \mid f_5 \in \{80; 443\}, f_8 > 1000, f_{36} > 1$$

$$p_{92}: f_5 \wedge f_{36} \wedge f_{66} \rightarrow a_{92} \mid f_5 \in \{1194; 443\}, f_{36} < 1, f_{66} > 10000$$

$$p_{93}: f_5 \wedge f_8 \wedge f_{36} \rightarrow a_{93} \mid f_5 = 80, f_8 > 100000, f_{36} > 10$$

$$p_{94}: f_5 \wedge f_8 \wedge f_{13} \rightarrow a_{94} \mid f_5 = 21, f_8 > 10000, f_{13} > 10$$

$$p_{95}: f_5 \wedge f_{13} \wedge f_{14} \rightarrow a_{95} \mid f_5 = 3306, f_{13} > 5, f_{14} > 1000$$

$$p_{96}: f_5 \wedge f_{14} \wedge f_{36} \rightarrow a_{96} \mid f_5 = 5060, f_{14} < 200, f_{36} > 10$$

$$p_{97}: f_5 \wedge f_8 \wedge f_{36} \rightarrow a_{97} \mid f_5 \in \{1883; 8883\}, f_8 < 1000, f_{36} < 2$$

$$p_{98}: f_5 \wedge f_8 \wedge f_{33} \rightarrow a_{98} \mid f_5 \in \{445; 139\}, f_8 > 50000, f_{33} > 100000$$

$$p_{99}: f_5 \wedge f_8 \wedge f_{33} \wedge f_{36} \rightarrow a_{99} \mid f_5 \in \{80; 443\}, f_8 > 20000, f_{33} > 60000, f_{36} > 2$$

$$p_{100}: f_5 \wedge f_8 \wedge f_{33} \wedge f_{36} \rightarrow a_{100} \mid f_5 = 443, f_8 > 500000, f_{33} > 100000, f_{36} > 5$$

$$p_{101}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{101} \mid f_7 < 5, f_8 < 100, f_{33} < 500$$

$$p_{102}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{102} \mid f_7 < 3, f_8 < 60, f_{33} < 100$$

$$p_{103}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{103} \mid f_7 > 1000, f_8 > 5000000, f_{33} > 100000$$

$$p_{104}: f_8 \wedge f_{33} \wedge f_{36} \rightarrow a_{104} \mid f_8 < 500, f_{33} < 100, f_{36} > 20$$

$$p_{105}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{105} \mid f_7 > 100, f_8 > 100000, f_{33} > 300000$$

$$p_{106}: f_7 \wedge f_{82} \wedge f_{83} \rightarrow a_{106} \mid f_7 > 10, f_{82} < 20000, f_{83} < 2000$$

$$p_{107}: f_7 \wedge f_8 \wedge f_{33} \rightarrow a_{107} \mid f_7 > 20, 10000 < f_8 < 50000, 2000 < f_{36} < 20000$$

$$p_{108}: f_8 \wedge f_{33} \wedge f_{78} \rightarrow a_{108} \mid f_8 > 500, f_{33} < 2000, f_{78} > 1000$$

$$p_{109}: f_8 \wedge f_{19} \wedge f_{40} \rightarrow a_{109} \mid f_8 > 1000, f_{19} = 0, f_{40} > 20$$

$$p_{110}: f_8 \wedge f_{33} \wedge f_{36} \rightarrow a_{110} \mid f_8 < 500, f_{33} > 100000, f_{36} < 1$$

The created rules serve as a knowledge base for expert systems. The conditions in the rules were determined taking into account the characteristics and features of the network event being analyzed. The definition of the generated production rules as a set P has been introduced.

$$P = \{p_1, p_2, \ldots, p_{110}\} \tag{4}$$

So $P: F \rightarrow A$ is expressed as a set.

B. Define Conditions for Production Rules

Not all of the extracted packet features were used in the mathematical representation of production rules. Because only the most optimal ones were selected to determine the network events. However, it is possible to express production rules using all features. However, only the optimal features were selected to eliminate redundant computations and improve the algorithm.

When determining the conditions of the production rule, the conditions were determined based on the characteristics of the network event. The 110 productions developed were derived from rule-defining network events in specifying rule conditions. That is, each rule was formed based on the characteristics of the network condition. In defining these conditions, the most desirable packet features were selected.

C. The Most Important Production Rules

From the perspective of network flow analysis, 25 production rules were selected as the most important, frequently occurring, and important to identify.

$$P = \{p_1, p_2, p_3, p_5, p_6, p_{19}, p_{22}, p_{25}, p_{26}, p_{42}, p_{43}, p_{58}, p_{60}, p_{64}, p_{65}, p_{67}, p_{69}, p_{72}, p_{80}, p_{81}, p_{84}, p_{85}, p_{86}, p_{90}, p_{103}\}$$

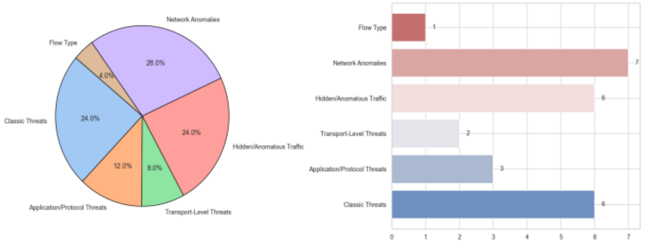

Fig. 3. The most important production rule group.

Among the selected production rules, there is no network event that falls exclusively into the group of normal and useful flows. The reason is that, firstly, the selected productions are aimed at the efficient operation of the system and the non-destructive operation of the system. Secondly, the events that fall into this group are often considered normal flows and are allowed in the network. However, they are used for analysis and filtering. Therefore, the 25 most priority production rules were formed due to their priority and frequency of occurrence in the selected production rules. Fig. 3 shows the selected production rules divided into groups.

A netflow dataset recorded in a laboratory environment was tested against 25 selected production rules.

Algorithm 4 Testing production rules

```
Load packet data from "packet_features.csv"
Initialize prediction column to empty string
Define apply_rules(row):
    if f_34 > 5000 & f_35 > 10 & f_45 < 1 & f_55 >100
    & f_58 < 10 & f_68 > 50:
        return "a1"
    else if f_36 > 1000 & f_45 < 1 & f_55 > 100 &
    f_58 < 5 & f_68 > 30:
        return "a2"
    else if f_7 < 5 & f_33 < 200 & f_55 > 20:
        return "a3"
    ...
    [continue similarly with all rule conditions up
    to a103]
    ...
    else:
        return " "
for each row in dataset:
    apply apply_rules to assign value to Prediction
    column
save updated dataset to "prediction_output.csv"
```

The results obtained from the dataset, which was checked based on the priority production rules, were stored in the “prediction_output” file (see Algorithm 4).

Algorithm 5 Production rules analysis

```
Load result data from "prediction_output.csv"
extract values from 'Prediction' column that are not null
  or "Normal"
remove duplicates and convert to list → worked_rules
define priority_rules as the list of 25 important rule
  identifiers:
  ['a1','a2','a3','a5','a6','a19','a22','a25','a26','a42','a43',
   'a58','a60','a64','a65','a67','a69','a72','a80','a81',
   'a84','a85','a86','a90','a103']
compute not_worked_rules = priority_rules-
  worked_rules
print number and list of worked_rules (sorted)
print number and list of not_worked_rules (sorted)
```

To analyze the tested dataset, we analyzed the production rules that worked and the production rules that did not work (see Algorithm 5).

D. Results Obtained Based on Production Rules

The result of the check can be seen in Fig. 4. When checking the dataset, not all of the selected top priority rules worked. Because all the given conditions must be met for all the rules to work. So, $a_1, a_2, a_{22}, a_{43}, a_{60}, a_{90}, a_{103}$ network events were detected, and the production rules $p_1, p_2, p_{22}, p_{43}, p_{60}, p_{90}, p_{103}$ worked, respectively. Out of 64 netflows, 50 normal flows and 14 corresponding to the network event were detected.

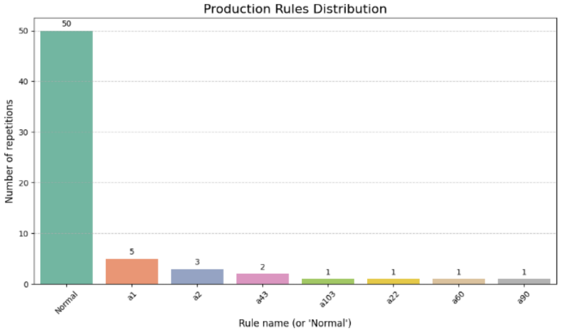

Fig. 4. Production rules analysis result.

The result obtained is presented in Fig. 5 in the form of a distribution of normal and working production rules.

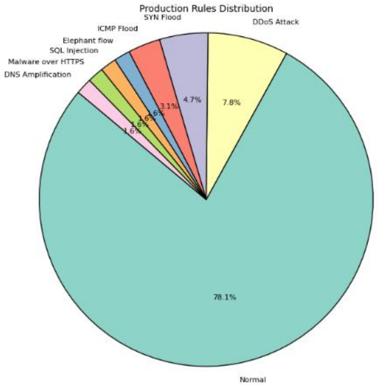

Fig. 5. Netflow general analysis.

The results of the network flow analysis recorded based on the developed production rules are presented in Fig. 6. Using the confusion matrix, the following results were obtained:

$$True\ Positive = 43, \quad False\ Positive = 2,$$

$$False\ Negative = 7, \quad True\ Negative = 12.$$

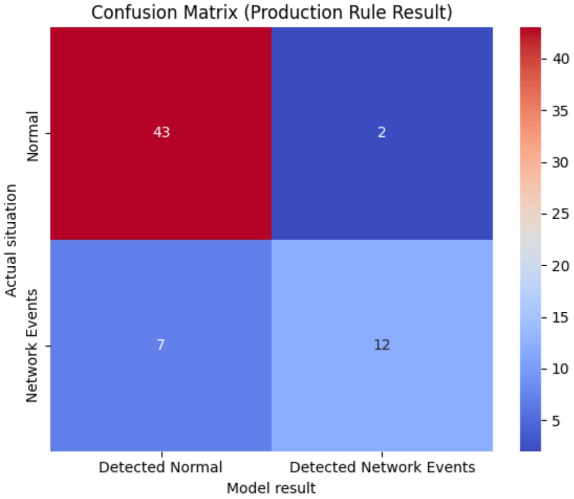

Fig. 6. Confusion matrix of production rules.

The network status detection methodology developed based on production rules achieved accuracy 85.94%, precision 95.56%, recall 86.00% and F1-score 90.53%. The results are presented in Table IV and Fig. 7.

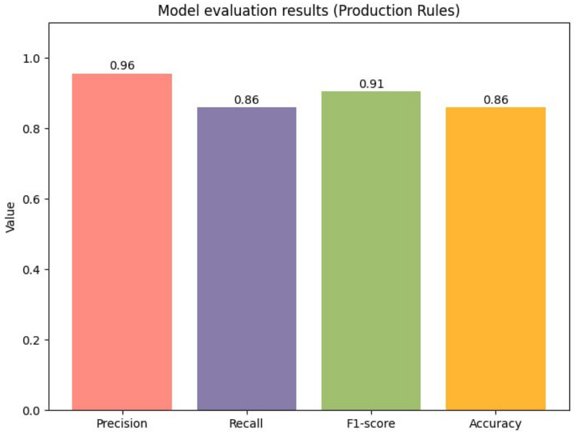

Fig. 7. Graph of model evaluation results.

TABLE IV. EVALUATION OF THE RESULT OF PRODUCTION RULES

№	Evaluation	Result
1	Accurancy	85,94%
2	Precision	95.56%
3	Recall	86.00%
4	F1-score	90.53%

The result obtained based on the ROC curve in Fig. 9 was 0.86, indicating that the developed production rules could effectively distinguish the network events that could occur. According to the Precision-Recall Curve results in Fig. 8, the average accuracy of the system was 0.92.

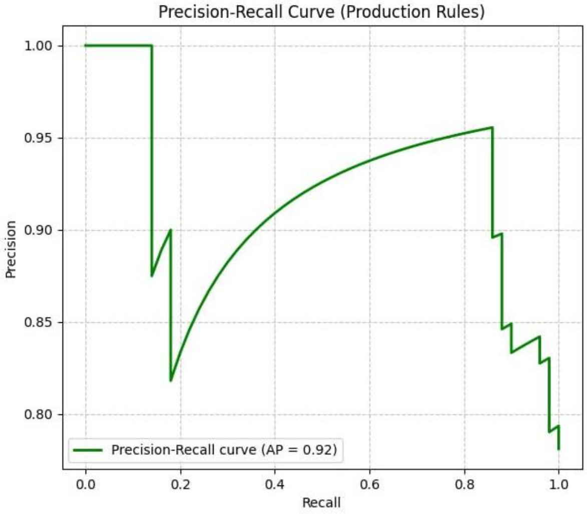

Fig. 8. Graph of precision-recall curve evaluation result.

This further confirms the reliability of the model in identifying attack flows. Thus, the developed rule-based model can make correct decisions in most of the netflows, but it is desirable to extend and refine the rules to increase sensitivity.

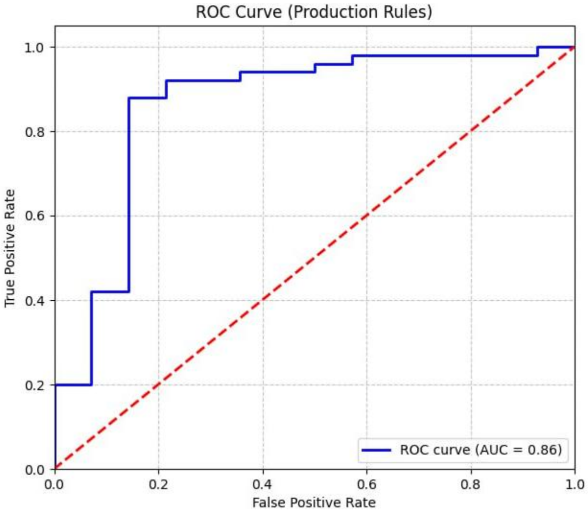

Fig. 9. Graph of ROC curve evaluation result.

V. Conclusion and Future Works

In automated and digital form, expert research results were presented in NetFlow monitoring analysis, mathematical representation in production logic decision-making. Real-time processing and collection of NetFlow traffic events in cyberspace were programmed. NetFlow data was converted to CSV and collected in PCAP format by performing a three-step algorithmic sequence. In a classical theoretical-fundamental way, cyber threats, cyber threats at application stages, anomalies and simple digital data flows were presented. These features identified 110 frequent network events and synthesized them into seven sets. Each event was formalized by the implementation of Boolean rules. These production rules were organized into a knowledge base that supports optimal identification of cyber threats. The protection of the infrastructure and the effective organization of the network events identification in the digital data flow were provided by architecture and analysis. A proposed decision tree based on Production logic was supported by fusion analysis.

At the same time, we plan to expand the work studied above in our future research. To do this, first of all, we want to analyze new unexplored areas of the industry in the implementation of a new scientific innovation project funded by the Ministry of Higher Education, Science and Innovation of the Republic of Uzbekistan.

References

[1] Yarashov, I., Rakhimberdiev, K., Jumabayeva, A., Subhonov, M., & Oydinoy, K. (2024, December). Modelling the Process of Ensuring Information Security in Inclusive Education Platforms Based on Functional Tables. In Proceedings of the 8th International Conference on Future Networks & Distributed Systems (pp. 881-889).

[2] Yarashov, I. (2022, September). Development of a reliable method for grouping users in user access control based on a Functioning table. In 2022 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-5). IEEE.

[3] Toshmatov, S., Yarashov, I., Otakhonov, A., & Ismatillayev, A. (2022, September). Designing an algorithmic formalization of threat actions based on a Functioning table. In 2022 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-5). IEEE.

[4] A. Kabulov, I. Saymanov, I. Yarashov and A. Karimov, "Using Algorithmic Modeling to Control User Access Based on Functioning Table," 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2022, pp. 1-5, doi: 10.1109/IEMTRONICS55184.2022.9795850.

[5] Kabulov, A.; Baizhumanov, A.; Saymanov, I. Synthesis of Optimal Correction Functions in the Class of Disjunctive Normal Forms. Mathematics 2024, 12, 2120. https://doi.org/10.3390/math12132120

[6] Kabulov, A., Baizhumanov, A., Berdimurodov, M. (2024). On the minimization of k-valued logic functions in the class of disjunctive normal forms. Journal of Mathematics, Mechanics and Computer Science, 121(1), 37–45. https://doi.org/10.26577/JMMCS202412114

[7] Sikos L. F. Knowledge representation to support partially automated honeypot analysis based on Wireshark packet capture files //Intelligent Decision Technologies 2019: Proceedings of the 11th KES International Conference on Intelligent Decision Technologies (KES-IDT 2019), Volume 1. – Singapore : Springer Singapore, 2019. – С. 345-351.

[8] Kabulov, A., & Yarashov, I. (2021, November). Mathematical model of information processing in the ecological monitoring information system. In 2021 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-4). IEEE.

[9] Kabulov, A.; Saymanov, I.; Babadjanov, A.; Babadzhanov, A. Algebraic Recognition Approach in IoT Ecosystem. Mathematics 2024, 12, 1086. https://doi.org/10.3390/math12071086

[10] Saymanov, I. (2024). Logical automatic implementation of steganographic coding algorithms. Journal of Mathematics, Mechanics and Computer Science, 121(1), 122–131. https://doi.org/10.26577/JMMCS2024121112

[11] Kabulov, A.; Normatov, I.; Saymanov, I.; Baizhumanov, A. On the Completeness of Classes of Correcting Functions of Heuristic Algorithms, Azerbaijan Journal of Mathematics, 2025, vol 15, no. 2. https://doi.org/10.59849/2218-6816.2025.2.51

[12] A. Kabulov, I. Normatov, A. Seytov and A. Kudaybergenov, "Optimal Management of Water Resources in Large Main Canals with Cascade Pumping Stations," 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2020, pp. 1-4, doi: 10.1109/IEMTRONICS51293.2020.9216402.

[13] Boateng A., Rahim M. M. Statistical analysis of network data flows and predictions using statistical and machine learning regression models : дис. – Brac University, 2024.

[14] Kabulov, A. V., Normatov, I. H., Ashurov A.O. Computational methods of minimization of multiple functions. Journal of Physics: Conference Series, 1260(10), 10200, 2019. doi:10.1088/1742-6596/1260/10/102007.

[15] Islambek Saymanov, et al. Numerical Methods of Synthesis of a Correct Algorithm for Solving Recognition Problems. Advances in Artificial Intelligence and Machine Learning. 2025;5(1):202. https://dx.doi.org/10.54364/AAIML.2025.51202

[16] Normatov, I., Yarashov, I., Otakhonov, A., & Ergashev, B. (2022, September). Construction of reliable well distribution functions based on the principle of invariance for convenient user access control. In 2022 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-5). IEEE.

[17] Maruf Juraev, Inomjon Yarashov, Adilbay Kudaybergenov, Alimdzhan Babadzhanov and Zilolaxon Mamatova, "Research on Network Flow Based on Statistical Analysis Methods" International Journal of Advanced Computer Science and Applications(ijacsa), 16(6), 2025. http://dx.doi.org/10.14569/IJACSA.2025.0160618

[18] A. Kabulov, I. Yarashov and A. Otakhonov, "Algorithmic Analysis of the System Based on the Functioning Table and Information Security," 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2022, pp. 1-5, doi: 10.1109/IEMTRONICS55184.2022.9795746.

[19] E. Navruzov and A. Kabulov, "Detection and analysis types of DDoS attack," 2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2022, pp. 1-7, doi: 10.1109/IEMTRONICS55184.2022.9795729.

[20] Batista, E., Alencar, B., Silva, E., Canário, J., Rios, R. A., Dustdar, S., . . . Prazeres, C. (2024). A new intelligent scheduler to improve reactive OpenFlow communication in SDN-based IoT data streams. Discover Internet of Things, 4(1), 15. doi:https://doi.org/10.1007/s43926-024-00068-3

[21] Yarashov, I., Juraev, M., Ismatillayev, A., & Rakhimberdiev, K. (2024, December). Behavior Modeling in Computerized Production Systems. In Proceedings of the 8th International Conference on Future Networks & Distributed Systems (pp. 873-880).

[22] Sikos, Leslie F. "Packet analysis for network forensics: A comprehensive survey." Forensic Science International: Digital Investigation 32 (2020): 200892.

[23] Laptiev, O., Musienko, A., Nakonechnyi, V., Sobchuk, A., Gakhov, S., & Kopytko, S. (2023, June). Algorithm for recognition of network traffic anomalies based on artificial intelligence. In 2023 5th International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA) (pp. 1-5). IEEE.

[24] Carlet, Claude. "Boolean functions." Encyclopedia of Cryptography, Security and Privacy. Cham: Springer Nature Switzerland, 2025. 292-296.

[25] Yao, Yuxiang, Zi-Gang Huang, and Duanqing Pei. "Diversified dynamic effects and their order origins in Boolean functions." Chaos, Solitons & Fractals 191 (2025): 115830.

[26] Carlet, C., Propagation Characteristics of Boolean Functions. In Encyclopedia of Cryptography, Security and Privacy (pp. 1984-1985). (2025).

[27] Kellerer, N., Sánchez, G., Alberto, H., Hagenmeyer, V., & Elbez, G., Attacks on the Siemens S7 Protocol Using an Industrial Control System Testbed. In Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems (pp. 770-779). (2025, June).

[28] Ţălu, M., DDoS Mitigation in Kubernetes: A Review of ExtendedBerkeley Packet Filtering and eXpress Data Path Technologies. JUTI: Jurnal Ilmiah Teknologi Informasi, 60-73. (2025).

[29] Bittencourt, G., & Marengoni, M., A customizable tool for the generation of production-based systems. WIT Transactions on Information and Communication Technologies, 2. (2025).

[30] Rahmatullah, R., Gao, H., Utama, R. P., Adi, P. D. P., Mubashir, J., Muwardi, R., ... & Dwiyanti, H., Multinode LoRa-MQTT of Design Architecture and Analyze Performance for Dual Protocol Network IoT. International Journal of Advanced Computer Science & Applications, 16(1). (2025).

[31] Li, X., & Chen, X. (2025). Intelligent Identification of Pile Defects Based on Improved LSTM Model and Wavelet Packet Local Peaking Method. International Journal of Advanced Computer Science & Applications, 16(5).

[32] Jing, G. U. O., Dejun, Z. H. U., & Qing, X. U. (2025). Metaheuristic-Driven Feature Selection for IoT Intrusion Detection: A Hierarchical Arithmetic Optimization Approach. International Journal of Advanced Computer Science & Applications, 16(6).

[33] Tahboush, M., Hamdan, A., Klaib, M., Adawy, M., & Alzobi, F. (2025). Detection Optimization of Brute-Force Cyberattack Using Modified Caesar Cipher Algorithm Based on Binary Codes (MCBC). International Journal of Advanced Computer Science & Applications, 16(3).

[34] Guo, L., Yan, F., Li, T., Yang, T., & Lu, Y. (2022). An automatic method for constructing machining process knowledge base from knowledge graph. Robotics and Computer-Integrated Manufacturing, 73, 102222.

[35] Yarashov, I. (2021, November). Algorithmic Formalization Of User Access To The Ecological Monitoring Information System. In 2021 International Conference on Information Science and Communications Technologies (ICISCT) (pp. 1-3). IEEE.

[36] Yarashov, I., Kuvonchbek, R., Juraev, M., & Rahmonov, F. (2024, December). Processing and Analysis of Network Flow Data Using Intelligent Technologies. In Conference on Internet of Things and Smart Spaces (pp. 40-51). Cham: Springer Nature Switzerland.

[37] Normatov, I., Yarashov, I., & Toshmatov, S. (2024). Research and modeling of authentication process using functioning table. Journal of Mathematics, Mechanics and Computer Science, 124(4), 71-85.

[38] Kabulov, A. V., Normatov, I. H. About problems of decoding and searching for the maximum upper zero of discrete monotone functions. Journal of Physics: Conference Series, 1260(10), 102006, 2019. doi:10.1088/1742-6596/1260/10/102006.

[39] A. Kabulov, I. Normatov, E. Urunbaev and F. Muhammadiev, "Invariant Continuation of Discrete Multi-Valued Functions and Their Implementation," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, pp. 1-6, doi: 10.1109/IEMTRONICS52119.2021.9422486.

[40] A. Kabulov, I. Saymanov, I. Yarashov and F. Muxammadiev, "Algorithmic method of security of the Internet of Things based on steganographic coding," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), 2021, pp. 1-5, doi: 10.1109/IEMTRONICS52119.2021.9422588.