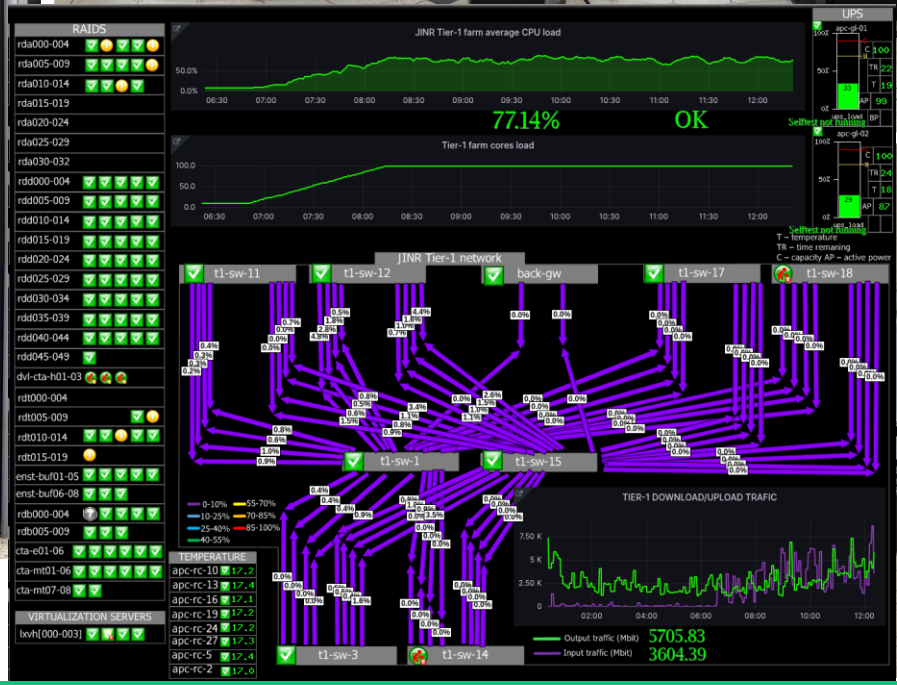
System logs automated analysis of serial consoles of JINR MLIT MICC servers

Kashunin I.,
Ososkov G., Baranov A.,
Lysenko E., Uzhinsky A.
July 2025

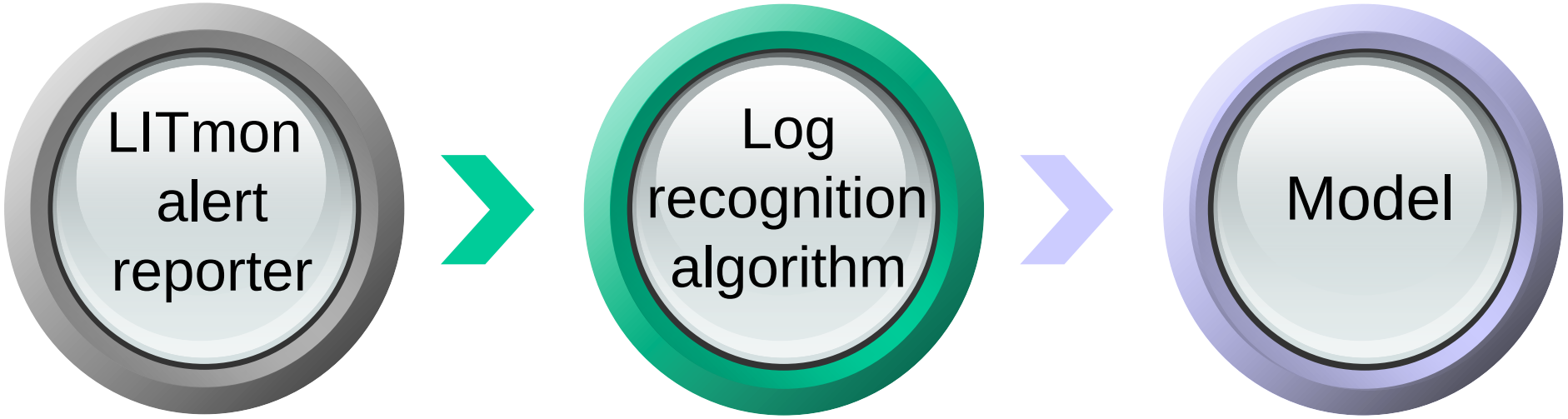
MICC – multifunctional information computing complex

General view of JINR MICC



LITmon – MICC monitoring system

Introduction



We need a convenient tool that will report possible problems and do the work of administrators itself, involving them only in extreme cases

It is necessary to develop a system that will classify logs and aggregate results under certain conditions

The neural network approach has proven itself well for analyzing text information

Linux serial consoles

srlc01: Logs per chosen interval

```
2024-07-04 11:20:01.740 ################# -= New LOGs from srlc01 =- #################
2024-07-04 11:20:01.740 Thu Jul 4 11:20:01 MSK 2024 Last update log
2024-07-04 11:20:01.740  5 consoles != 0 size
2024-07-04 11:20:01.740 rdz14   [FAILED] Failed unmounting /e/p01.
2024-07-04 11:20:01.740 rdz14   [FAILED] Failed unmounting /e/p01/pool/meta.
2024-07-04 11:20:01.740 rdz14   [FAILED] Failed unmounting /e/p02.
2024-07-04 11:20:01.740 rdz14   [FAILED] Failed unmounting /e/p02/pool/meta.
2024-07-04 11:40:02.636 ################# -= New LOGs from srlc01 =- #################
2024-07-04 11:40:02.636 Thu Jul 4 11:40:02 MSK 2024 Last update log
2024-07-04 11:40:02.640 rdz14          iDRAC, Update FW, Install OS)
2024-07-04 11:40:02.640 rdz14    OK
2024-07-04 11:40:02.640 rdz14   BIOS Version: 2.13.0
```

It is necessary to analyze all existing serial consoles logs, which contain **more than a million lines**. The logs contain data for 3 years. **This will require a lot of time investment.**

LITmon alert reporter development

1) Collect training data for dataset

2) Develop and train a neural model

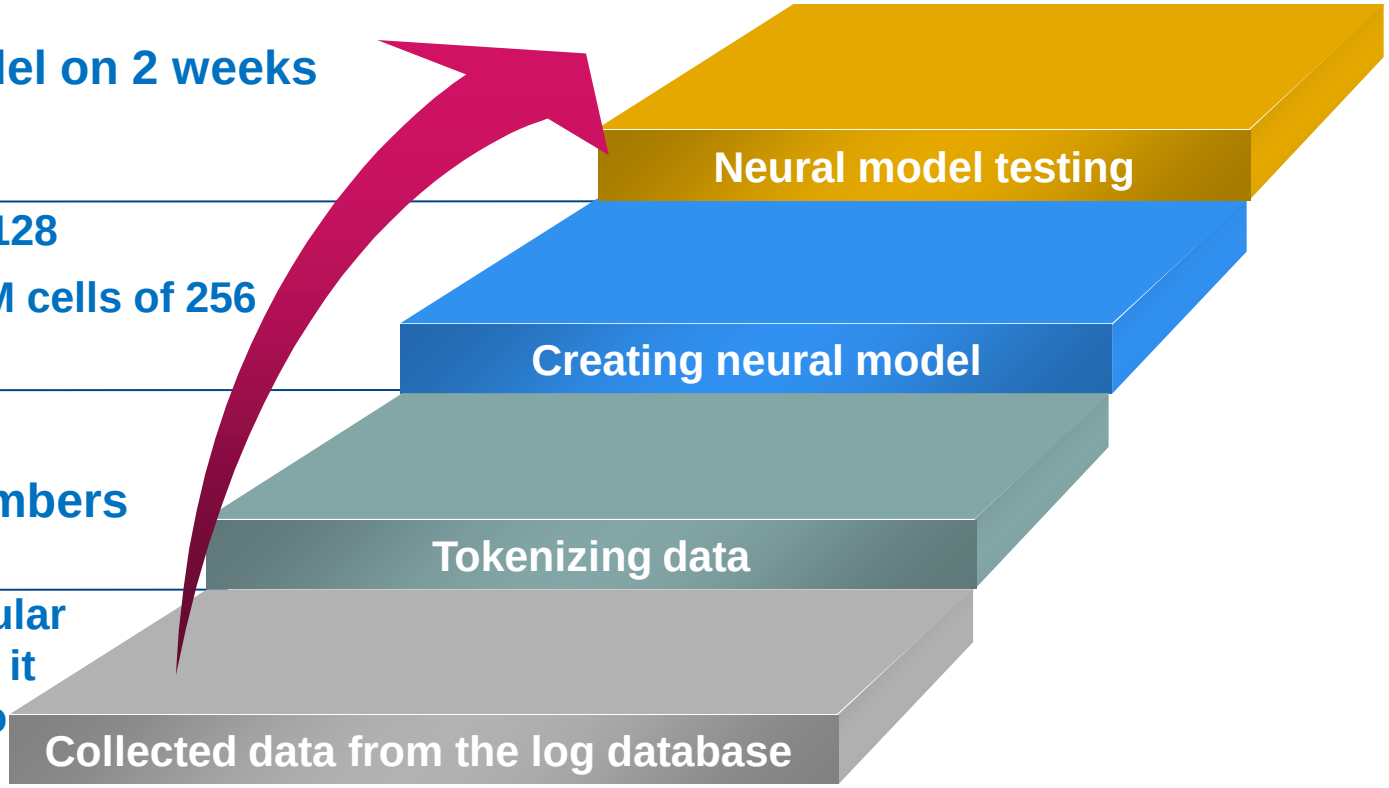
3) Automate loading logs into the model and display the results of recognition in a web interface

Creating of data set

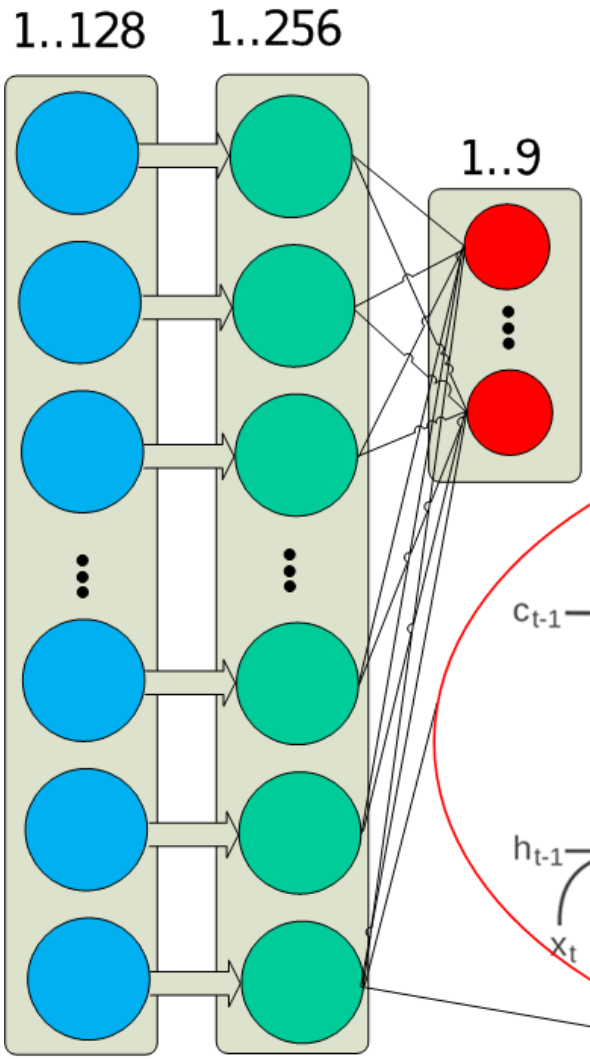
Number of errors in the training set by type	Real units	Generated units
Type 1 kernel panic - Linux kernel crashes	110	4190
Type 2 network errors - disconnection of network interfaces	460	4098
Type 3 critical medium errors - critical hard drive errors	1802	4102
Type 4 blk_update - correctable hard drive errors	4035	4035
Type 5 kernel errors - kernel errors	372	4257
Type 6 uncorrectable memory errors - uncorrectable memory errors	77	4077
Type 7 hardware errors - hardware errors	1137	3732
Type 8 raid errors	142	4102
Type 0 good - normal state of logs	3128	3128

Model development based on LSTM

- Testing neuro model on 2 weeks of logs
- Embedding layer of 128
- Hidden layer of LSTM cells of 256
- Output layer of 9
- From words to numbers
- Filtering logs by regular expression and drop it not corresponding to certain class



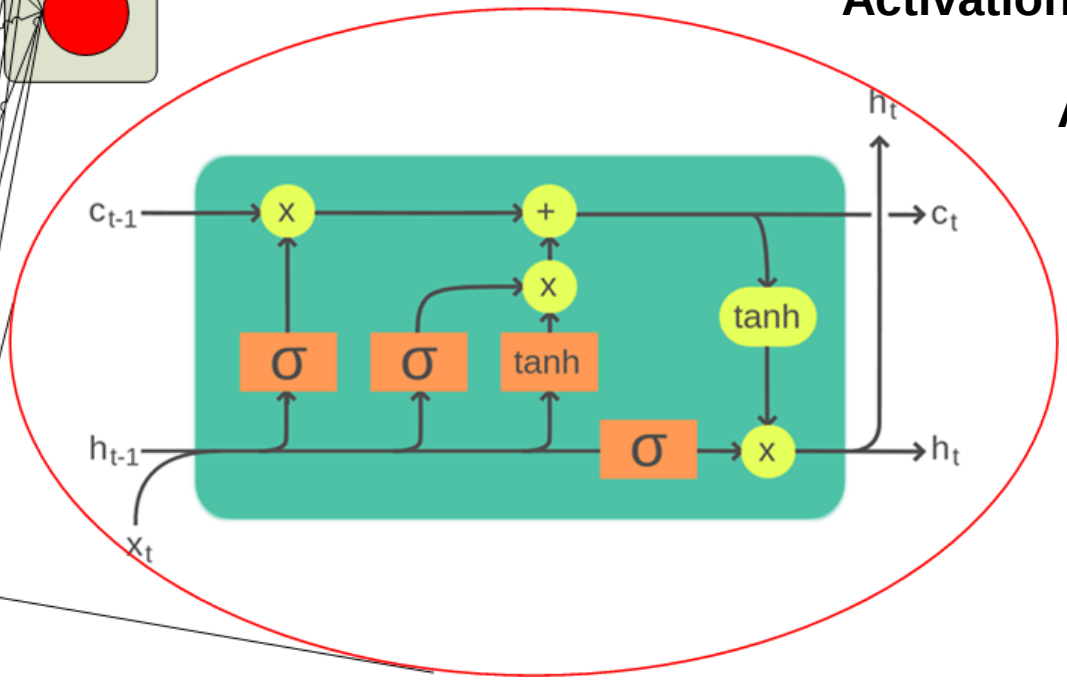
LSTM (Long short-term memory) neural model constriction



- Embedding layer size - 128
- Hidden layer of LSTM cells of 256
- Output layer size - 9
- Loss function: cross entropy

Activation function: linear

ADAM optimizer



LOGmon model testing results for 2 weeks

probability of error detection = true_negative model prediction / true_negative real * 100%

Number of errors in the training set by type	Units LOGmon	Units real
correctable_disk_error	927	31
kernel_errors	13	13
hardware_errors	36	0
critical_disk_errors	10	9
kernel_panic	10	2
network_errors	3	1
uncorrectable_memory _errors	3	1
raid_errors	7	1
unknown	69	-

number_of_test = 29242

If probability < 60% - **unknown**

accuracy = 0.9771
precision = 1.0
recall = 0.9769
Sp = 0.0133
purity = 0.0
f1_score = 0.9883
log_recognition = 0.99

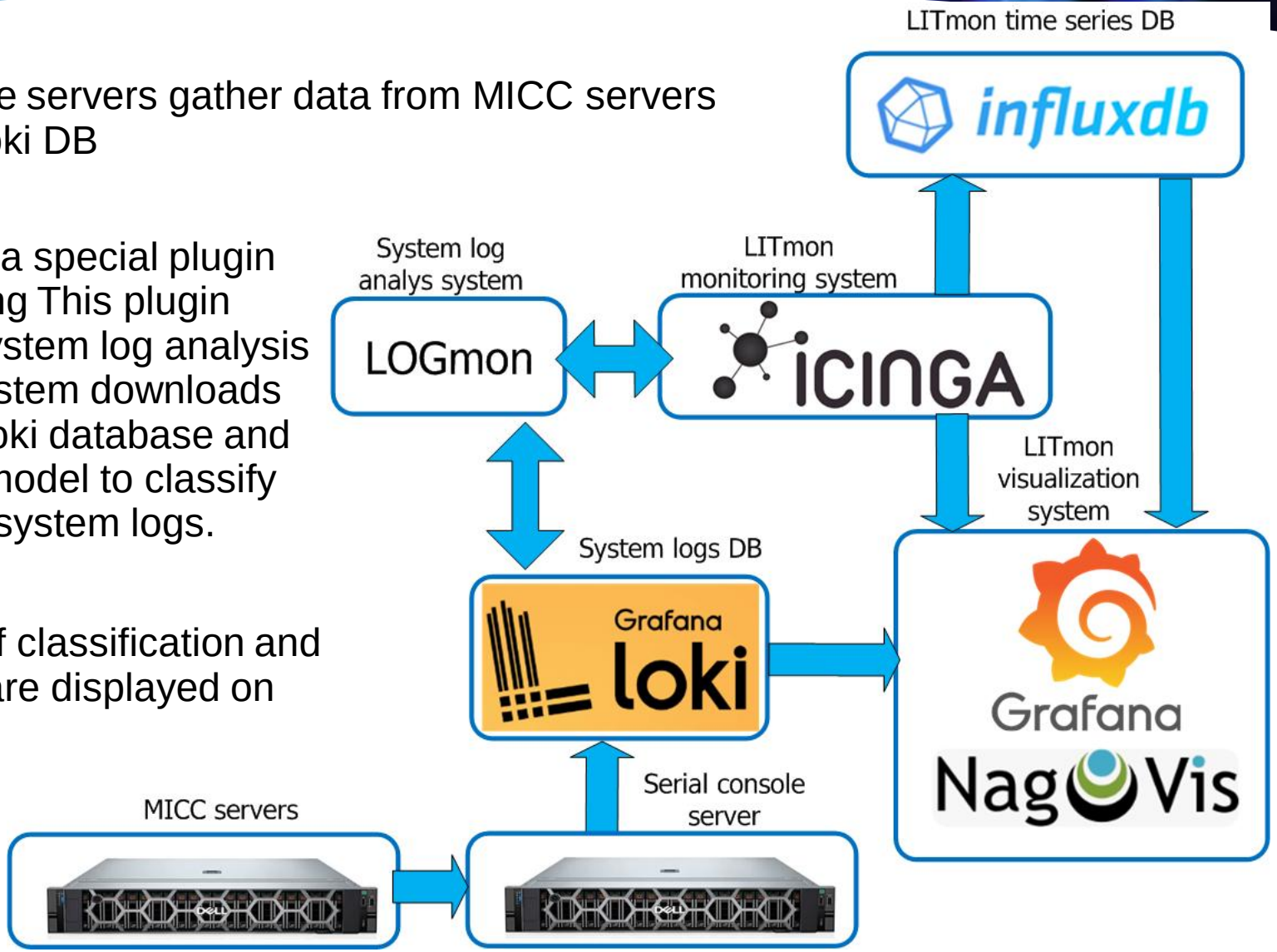
probability_error_recognition = 100%

Program realization: structural diagram for automation of log recognition process

1) Serial console servers gather data from MICC servers and sent it to Loki DB

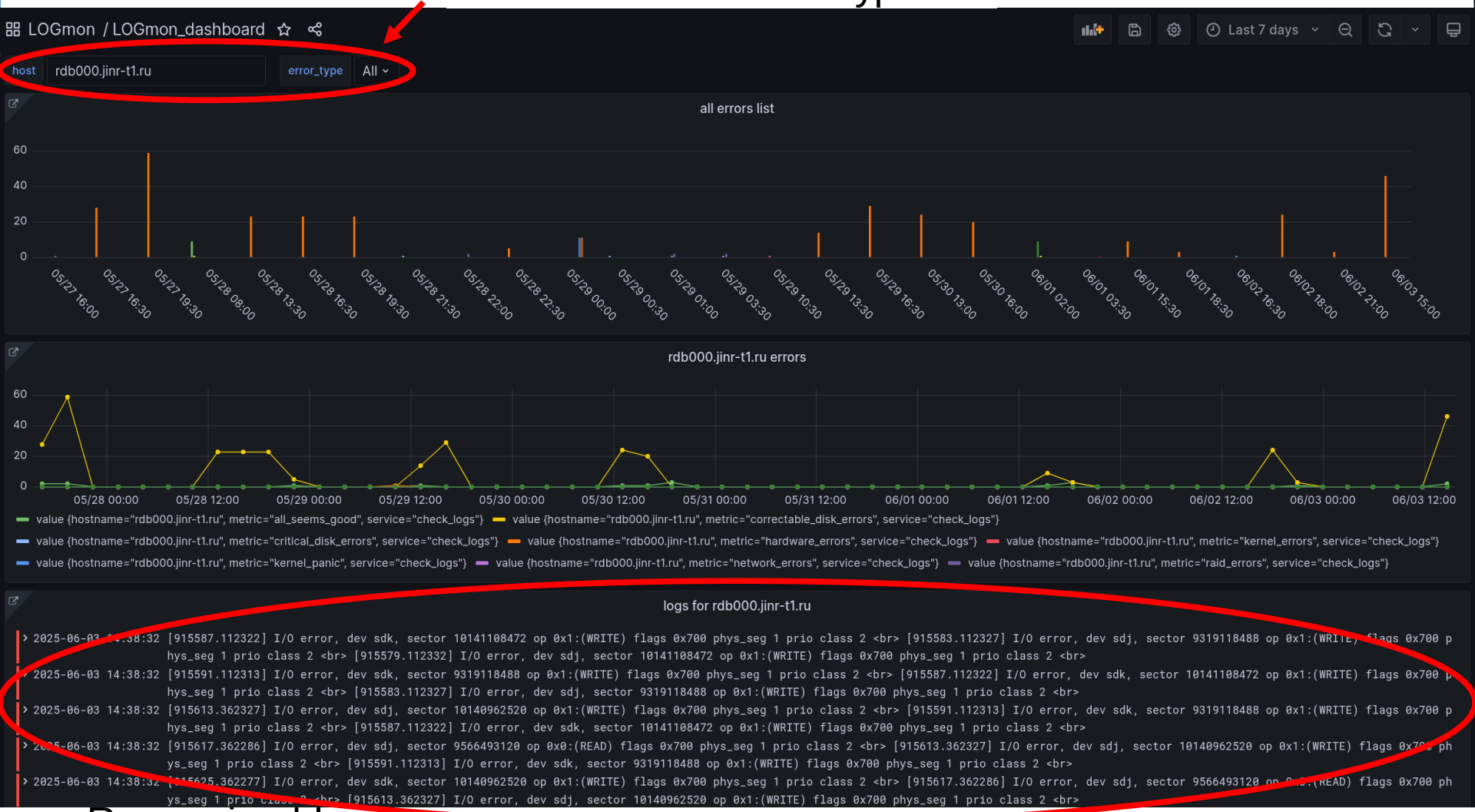
2) LITmon runs a special plugin for log monitoring This plugin accesses the system log analysis system. This system downloads data from the Loki database and runs LOGmon model to classify serial consoles system logs.

3) The results of classification and statistical data are displayed on various dashboards



Program realization: Grafana dashboard

Select Host and error type



Recognized logs

Real results: examples

LOGmon estimation

Время состояния 2025-04-30 22:05:15

==========correctable_disk_errors==========

2025-04-30T22:20:01 **rdd017.jinr-t1.ru**

[371995.155171] XFS (md0): Please unmount the filesystem and rectify the problem(s)
[371995.155169] XFS (md0): Filesystem has been shut down due to log error (0x2).
[371995.155165] XFS (md0): log I/O error -5

Время состояния 2025-05-28 21:30:44

==========hardware_errors==========

2025-05-28T23:20:01 **rdb01.jinr.ru**
[2801357.821692] ixgbe 0000:3b:00.0 ens2f0: Reset adapter
2025-05-28T23:20:01 **rdb01.jinr.ru**
[2801357.821692] ixgbe 0000:3b:00.0 ens2f0: Reset adapter
[2801357.821650] tx_buffer_info[next_to_clean]
[2801357.821650] ixgbe 0000:3b:00.0 ens2f0: Detected Tx Unit Hang
==========correctable_disk_errors==========
2025-05-28T23:20:01 **rdb01.jinr.ru**
[2801357.821692] ixgbe 0000:3b:00.0 ens2f0: Reset adapter
[2801357.821650] tx_buffer_info[next_to_clean]

Human expert estimation

Valery Mitsyn 4/30/25, 22:22

Hi All,

опять проблема с s/w raid
{{{
[371995.155165] XFS (md0): log I/O error -5
[371995.155169] XFS (md0): Filesystem has been shut down due to log error (0x2)
[371995.155171] XFS (md0): Please unmount the filesystem and rectify the problem(s).
}}}

Best regards,
Valery Mitsyn

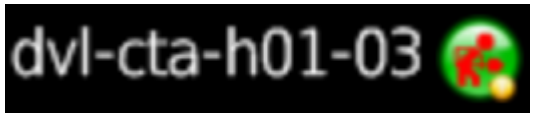
Valery Mitsyn 5/29/25, 00:06

уже несколько таких ошибок на консоле:
{{{
rdb01 [2800971.777422] ixgbe 0000:3b:00.0 ens2f0: Reset adapter
rdb01 [2801357.821650] ixgbe 0000:3b:00.0 ens2f0: Detected Tx Unit Hang
rdb01 [2801357.821650] Tx Queue <22>
rdb01 [2801357.821650] TDH, TDT <151>, <1ab>
rdb01 [2801357.821650] next_to_use <1ab>
rdb01 [2801357.821650] next_to_clean <151>
rdb01 [2801357.821650] tx_buffer_info[next_to_clean]
rdb01 [2801357.821650] time_stamp <1a6f5332b>
rdb01 [2801357.821650] jiffies <1a6f54381>
}}}

Best regards,
Valery Mitsyn

Real results: conclusions

1)The model was able to detect errors that the expert identified during real-life operation and displays them in the form of a pictogram



2) The automated algorithm works faster than a human

3) The model is able to detect errors that were not contained in the training sample

Conclusions

- Developed a neural network algorithm for log monitoring
- An augmented training dataset was created
- The developed LSTM neural model is trained and tested
- Testing of the model showed that it recognizes 100% of errors in serial console log by tested period
- The model is implemented in the form of a software and hardware complex for the tasks of automating log recognition
- The software allows report about suspension logs in real-life operation

Thank you for your attention!

Backup

Logs errors type

Error type example	Error description
Kernel panic - not syncing: Fatal exception Kernel panic - not syncing: Fatal hardware error!	kernel panic - Linux kernel crashes
ixgbe 0000:01:00.0 em1: NIC Link is Down	network errors - disconnection of network interfaces
blk_update_request: critical medium error, dev sdn, sector 18015416216	critical medium errors - critical hard drive errors
blk_update_request: I/O error, dev sdh, sector 12810658040	correctable hard drive errors
<IRQ> [<ffffffff997b1bec>] dump_stack+0x19/0x1f segfault at a9 ip 00007f46d2dff535 sp 00007f46af7fc6f0 error 4 in libpython2.7.so.1.0[7f46d2d77000+17e000]	kernel errors
UEFI0330: One or more memory errors have occurred during the Double Data Rate	uncorrectable memory errors
mce: [Hardware Error]: Machine check: Processor context corrupt	hardware errors
md/raid1:md1: Disk failure on sda1, disabling device	Raid errors
1657080.156077] systemd-shutdown[1]: Failed to wait for process: Protocol error	good - normal state of logs

Solution to the problem of unbalanced classes: generating logs

Logs gathering

```
;;;
2023-06-21T13:40:03Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wn383"}     wn383       [385249.097562] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2023-06-21T13:40:03Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wn383"}     wn383       [385249.104260] CR2: 00002aacf6f21000 CR3: 00000005823e2000 CR4: 00000000000207e0
2023-06-21T13:40:03Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wn383"}     wn383       [385249.112364] Call Trace:
;;;
```

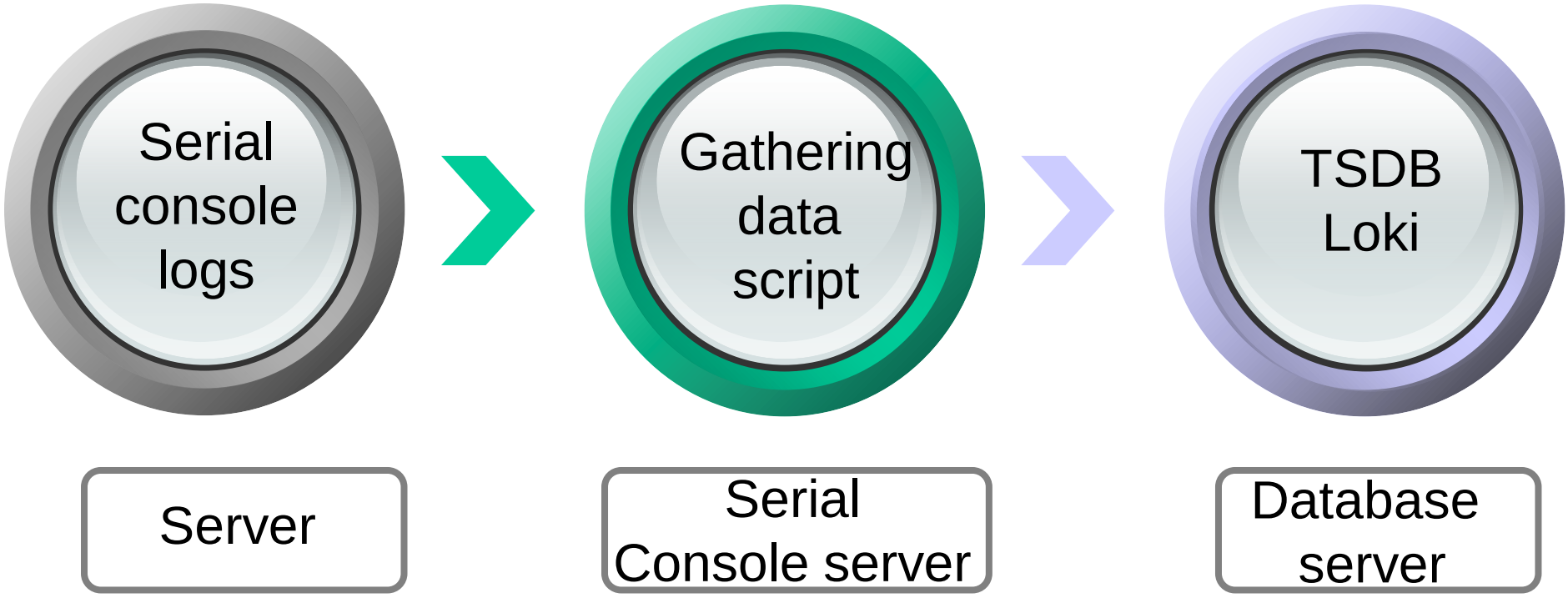
Logs processing

```
def data_template11(year,day,mounth,hours,min,sec,host,six_number_gen,six_number_gen1,twelve_number_and_text_gen,nine_number_and_text_gen,):
    data=r"""%s-%s-%sT%s:%s:%sZ {filename="/var/log/srcnsl_logs/srlc02.log", host="%s"}     %s       [%s.%s] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
%s-%s-%sT%s:%s:%sZ {filename="/var/log/srcnsl_logs/srlc02.log", host="%s"}     %s       [%s.%s] CR2: 0000%s CR3: %s CR4: 00000000000207e0
%s-%s-%sT%s:%s:%sZ {filename="/var/log/srcnsl_logs/srlc02.log", host="%s"}     %s       [%s.%s] Call Trace:
;;;""" % (year,day,mounth,hours,min,sec,host,host,six_number_gen,six_number_gen1,\
year,day,mounth,hours,min,sec,host,host,six_number_gen,six_number_gen1,twelve_number_and_text_gen,nine_number_and_text_gen,\
year,day,mounth,hours,min,sec,host,host,six_number_gen,six_number_gen1)
    return data
```

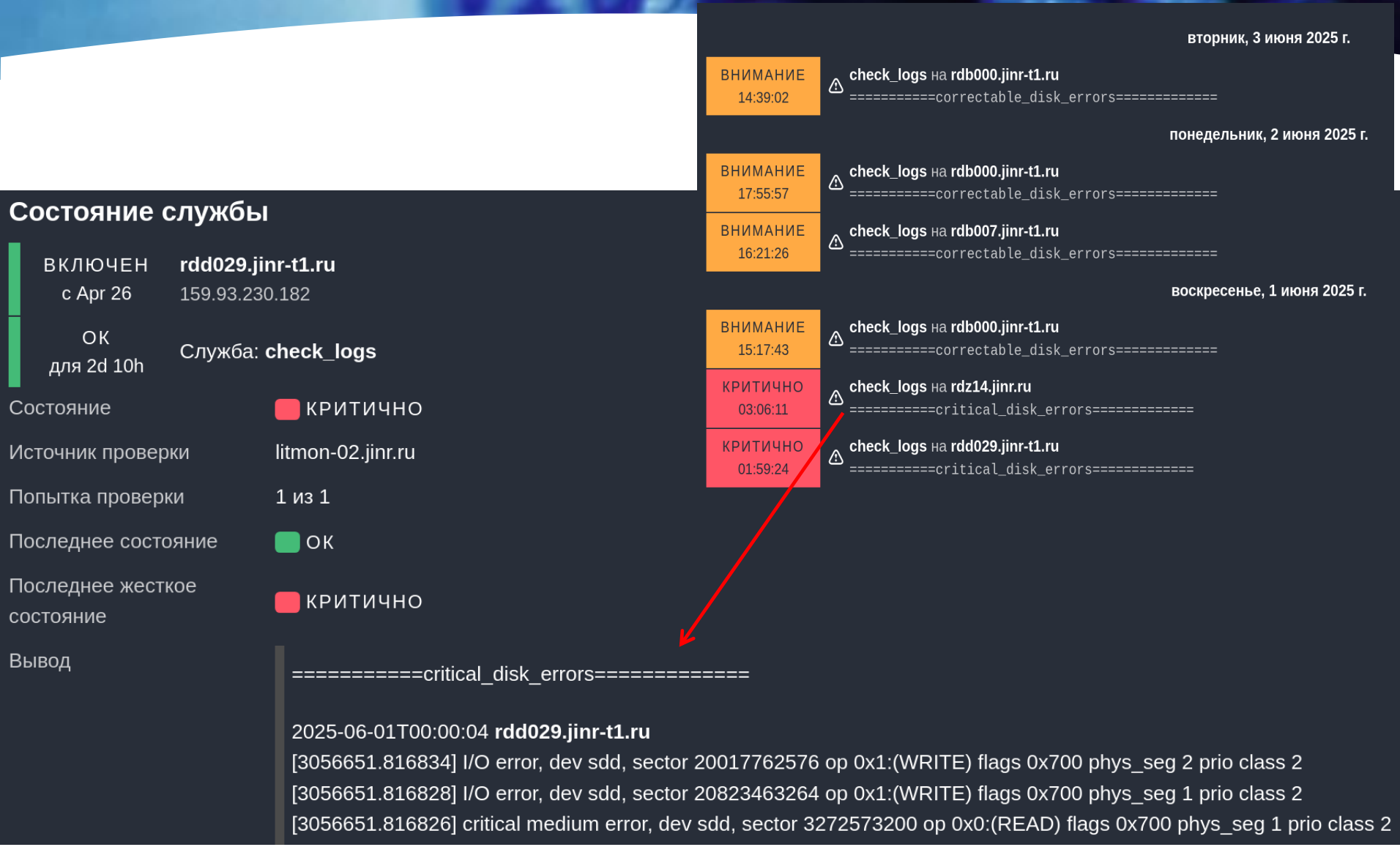
Logs generation

```
2023-10-10T01:24:47Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna394"}     wna394       [527288.800235] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2023-10-10T01:24:47Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna394"}     wna394       [527288.800235] CR2: 00001pens4muzuj7 CR3: t3mq7vfk1 CR4: 00000000000207e0
2023-10-10T01:24:47Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna394"}     wna394       [527288.800235] Call Trace:
;;;
2024-21-01T01:55:18Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna220"}     wna220       [374626.703578] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2024-21-01T01:55:18Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna220"}     wna220       [374626.703578] CR2: 0000k8hp85ar5br5 CR3: 52rv8tgtj CR4: 00000000000207e0
2024-21-01T01:55:18Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna220"}     wna220       [374626.703578] Call Trace:
;;;
2022-20-07T15:53:30Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna269"}     wna269       [186567.606716] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2022-20-07T15:53:30Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna269"}     wna269       [186567.606716] CR2: 0000ftrsrw2ez5gt CR3: 4makwfu22 CR4: 00000000000207e0
2022-20-07T15:53:30Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna269"}     wna269       [186567.606716] Call Trace:
;;;
2023-12-06T14:51:01Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna119"}     wna119       [764360.806681] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2023-12-06T14:51:01Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna119"}     wna119       [764360.806681] CR2: 0000dq58c68zuzru CR3: 9anj29xkf CR4: 00000000000207e0
2023-12-06T14:51:01Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna119"}     wna119       [764360.806681] Call Trace:
;;;
2024-16-09T11:23:27Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna108"}     wna108       [825350.660801] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2024-16-09T11:23:27Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna108"}     wna108       [825350.660801] CR2: 0000kvdcygwc9u35 CR3: 948yzvcf3 CR4: 00000000000207e0
2024-16-09T11:23:27Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna108"}     wna108       [825350.660801] Call Trace:
;;;
2023-05-03T07:10:54Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna281"}     wna281       [563050.308830] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
2023-05-03T07:10:54Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna281"}     wna281       [563050.308830] CR2: 0000bgracyqyyk7d CR3: hm2v84cpe CR4: 00000000000207e0
2023-05-03T07:10:54Z {filename="/var/log/srcnsl_logs/srlc02.log", host="wna281"}     wna281       [563050.308830] Call Trace:
```

Gathering data from serial consoles



Program realization: icingaweb interface



Metrics for LOGmon Efficiency

accuracy = (true_positive+true_negative) / (true_positive+true_negative+false_positive+false_negative) * 100%

precision = (true_positive)/(true_positive+false_positive) * 100%

recall = TPR = true_positive/(true_positive+false_negative) * 100%

probability of error detection = true_negative model prediction / true_negative real * 100%

Metrics for LOGmon Efficiency

accuracy = (true_positive+true_negative) / (true_positive+true_negative+false_positive+false_negative)

precision = (true_positive)/(true_positive+false_positive)
recall = TPR = true_positive/(true_positive+false_negative)
Sp = TNR = true_negative/(true_positive)+false_positive)
purity = (1 - precision)

f1_score = F1 = (2 * true_positive)/ (2*true_positive_list+false_positive+false_negative)

probability of error detection = true_negative model prediction / true_negative real * 100%