Grenze International Journal of Engineering and Technology, Jan Issue

An Approach for Machine-First Incident Management

Shailesh K.S.[1], G.Sharada[2] and P.Venkata Suresh[3]
[1]Independent Researcher, Bengaluru, India
Email: shailesh.shivakumar@gmail.com
[2]Department of Information Technology, Malla Reddy College of Engineering & Technology, Hyderabad, India
Email: gsharada8@gmail.com
[3]School of Computer and Information Sciences, Indira Gandhi National Open University, New Delhi, India
Email: pvsuresh@ignou.ac.in

***Abstract*—The objective of the paper is to propose a Machine Learning based model for incident management. Incident management is a key activity during the operations phase. It's essential to address the incidents in an optimal time so that there is no disruption to the operations. As the number of incidents increase, it's essential to have algorithms which allocate the incident to the right bot or concerned department through which the incident gets resolved or closed. The paper takes machine-first, smart, automation focused operations approach for incident management. The machine first approach proposes various tools, methods and processes that use machine learning to optimize the overall ticket management process. The Machine First Incident Management (MFIM) approach improves the productivity as well as quality. We elaborate the detailed method for smart ticket management, pro-active failure prediction, smart problem management, proactive maintenance and smart shift left in this paper.**

***Index Terms*— Incident management, Ticket management, Machine-first operations, Automated operations, Operations management.**

I. Introduction

Incident management system (also known as ticket management system) is a labour intensive activity. Given below are the key problem areas in the ticket management system:

- Effort intensive: A significant volume of tickets are repetitive, but the ticket handling team repeats the solution process for each of the tickets leading to high manual effort.
- Resolution time: Since majority of the ticket management process is manual, the resolution time is high.
- Resolution quality: Manual resolution of tickets lead to quality issues
- Resource availability: For supporting tickets in multiple geographies and multiple time zones we need to staff the resources across all the geographies.
- Impact on mission critical setup: Mission critical applications need strict Service Level Agreements (SLA) that are difficult to manage.

II. Literature Review

The global smart ticketing market size was valued at USD 11.3 billion in 2018 and is expected to register a CAGR of 14.9% from 2019 to 2026 [1]. [2] demonstrates defining requirements for an incident management

Grenze ID: 01.GIJET.9.1.556

system by a case study. A holistic , process oriented approach to ISO/IEC 27001 compliant security incident management that integrates multiple state-of-the-art security tools and has been applied to a real-world scenario is present in [3]. [4] presents a case study on current practice of information security incident management in three large organizations. Implementation of incident management for data services using ITIL V3 in telecommunication operator company is presented in [5]. The article [6] provides the analysis of decision support in incident management systems. [7] proposed machine learning model for incident categorization and incident duration prediction that could be adopted into business process of incident handling.[8] provides performance analysis of machine learning algorithms for IT incident management.

III. MOTIVATIONS

We have listed the main motivations for the solution below:

- Automated failure prediction for the system and services
- Machine learning based models for automated incident management
- Automated methods for predictive maintenance
- Methods for shift left architecture
- Methods for ticket avoidance architecture

IV. PROBLEM DEFINITION

As most of the ticket management systems have huge volume of historical data that provide structured insights into the ticket volume, ticket category and the resolution method adopted, we can use this information to train a machine learning model. The machine learning model can learn to auto classify the ticket and use auto-resolution methods to complete the ticket.

V. SOLUTION METHODOLOGIES

In this section , we define the detailed methods for smart ticket management, pro-active failure prediction, smart problem management, proactive maintenance and smart shift left.

A. Methods and process for Smart ticket management

We have depicted the methods and process for smart ticket assignment and smart ticket raising in Fig. 1

Machine learning for ticket classification

As the first step in the process, we feed historical data of the ticket, enriched logs and the knowledge base to the machine learning algorithms as depicted in Fig. 2. The machine learning (ML) algorithms analyse ticket description and learn to classify the ticket into logical categories. ML algorithms are also trained to identify the appropriate bot to handle the ticket. The ML algorithm is trained to route the ticket to the corresponding resolver bot using the probability threshold for a given ticket category. The resolver bot then handles and resolves the ticket. If the ML algorithm cannot find a suitable resolver bot for the given ticket category or if the resolver bot cannot fully resolve the ticket, the ticket is then assigned to an agent to resolve the ticket. We can train the machine learning algorithms also to predict the most likely problem areas, seasonal trends , patterns and forecast the ticket volume for a given time window.

We could also train the ML algorithms to handle the below given scenarios:

- **Automated triage -** Auto-classification, Auto-upgrade, Auto-assignment
- **Monitoring Analyser -** Auto-detect threshold variations with patterns
- **Failure predictor -** Auto-detect and correlate events to predict
- **Automated responder -** Auto-response to concerned along with resolution time
- **Automated bot classifier -** Auto-detect bot for fixes
- **SLA breach predictor -** Auto-detect and escalate to support team
- **Automated remediation -** Auto-fix identified issues

Automatic ticket assignment and ticket management

As depicted in Fig. 2, the trained ML model is used for automatic ticket management. The monitoring agents (log, key performance indicator (KPI), SLA monitoring agents) automatically log any breaches into the incident management system. Business users or customer service personnel can also log the incident. Once the ticket is logged, it is taken up by bot manager. The bot manager is trained for ticket classification and resolver bot identification. Based on the training, the bot manager classifies the ticket, does auto triage and routes the ticket to

the corresponding resolver Robotic Process Automation (RPA) bot. The query responder also handles the user queries and closes the ticket. If the ticket is complex enough for bot to handle, it will be routed to the team. For self-heal scenarios (such as restarts, patches, rules based configuration changes, rules-based database updates) the healing bot carries out the self-healing action.

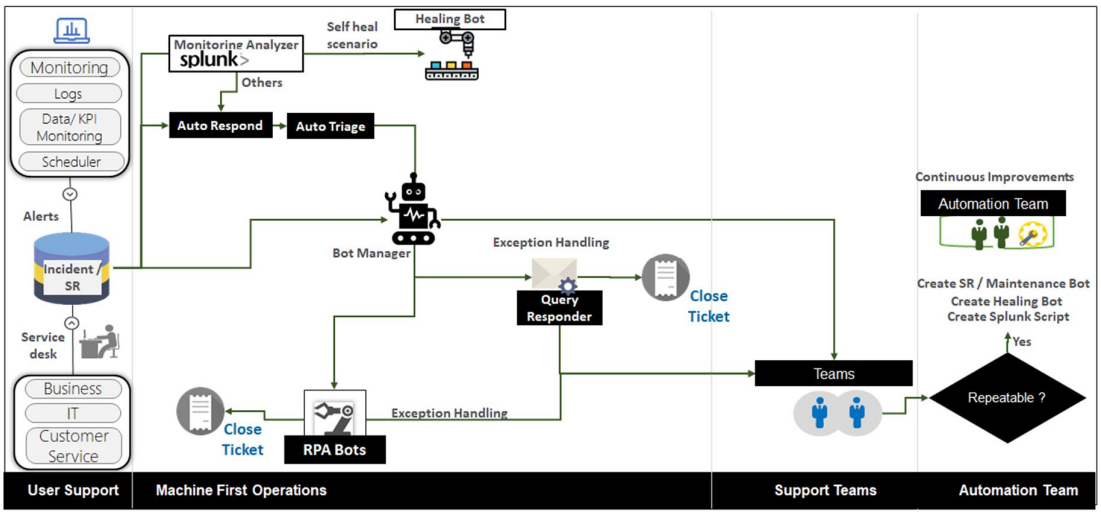

Figure 1. Automated Ticket Assignment and ticket management

Fig. 2 depicts training using ML algorithms.

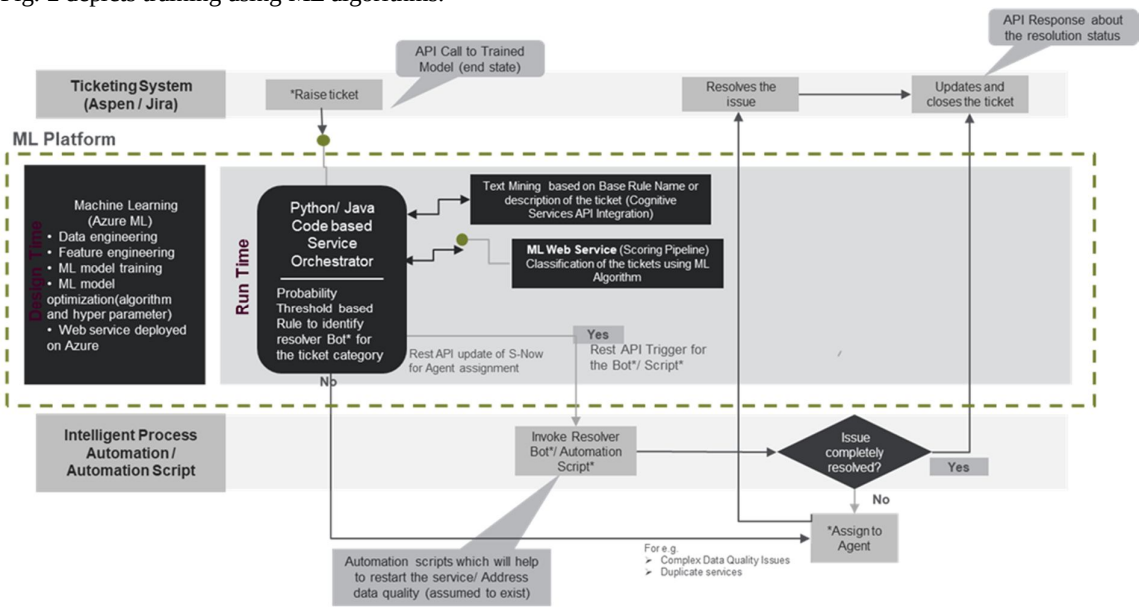

Figure 2. Machine Learning training for ticket resolution

The model is trained based on the historical data of the tickets during design time. We configure the thresholds during run time for the classification job. The machine learning model classifies the ticket using the name or the ticket description. Based on the identified classification, we invoke the corresponding ticket resolver script. If the ticket is resolved through the ticket resolver bot, we conclude the workflow else we assign the ticket to the agent.

B. Methods and process for failure predictor

Use the bots and statistical methods to continuously monitor the tickets, logs to predict the following:

- Ticket volume for a time period and for a given geography
- Error rate for a given category

- Main ticket categories
- Ticket Service Level Agreement (SLA) violation and turnaround prediction
- System failure prediction based on the historical failure incidents and outage incidents.

We have depicted the failure prediction process in Fig. 3. In case of SLA breaches, the bot notifies the incident management team.

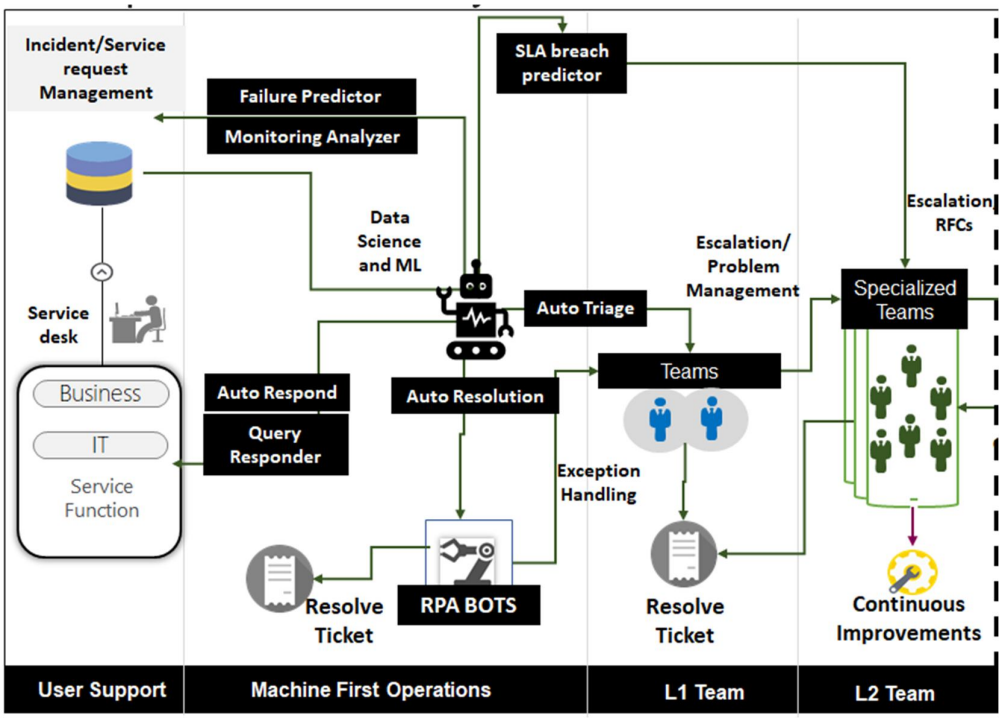

Figure 3. Failure predictor process

The machine first operations team monitors the incident management system. The system leverages the trained machine learning models for auto resolution using the RPA bots. If the tickets cannot be resolved, then they are auto triaged by assigning to the Level 1 (L1) and Level 2 (L2) teams.

C. Methods and process for Smart problem management

Self service is an integral part of the smart problem management. In this category, we provide self-service tools, self-service applications and business self-service management interfaces for users. Few examples of self service include but not limited to the following:

- Admin Tool for business managers improves self-service and interactive user interface help for users.
- Self-service tool and self-service portals for on boarding and customer assignments.
- Self-service reports, calculators, smart search, smart recommendations and productivity improvement tools
- High Potential for Root Cause Analysis (RCA) and Problem Management to resolve repetitive issues across the application

D. Methods and process for proactive maintenance

As part of proactive maintenance, a continuous real time monitoring agent monitors the application and the servers. Based on the rule-based configured thresholds such as consistent performance time, system health check thresholds (CPU, memory, network utilization), error rate the maintenance bot classifies the issues. Table I provides the sample rules, issue categories and the possible proactive maintenance step:

E. Methods and process for smart shift left

Shift left involves structuring and automating the complex activities involving manual intervention. It includes but not limited to below given steps:

TABLE I. PROACTIVE MAINTENANCE RULES

Problem symptom	Problem category	Proactive maintenance step
The performance of the application is more than 5 minutes consistently for more than 10 minutes	Performance issue	Increase the number of service pods to handle high traffic Log the performance ticket for further debugging
The heartbeat ping responds with 40x response code	System outage	Attempt auto restart of the system and the related services Log the production incident ticket with high priority
The CPU, memory thresholds exceed consistently for more than 10 minutes	Resource issue	Increase the number of service pods to handle high traffic Log the resource ticket for further debugging

- Creation of standard operation procedures that codify the knowledge. It helps management team to follow up to one structured process to deliver high quality outcome.
- Identify the high volume issues and transactions and create SOP (standard operating procedure) for them.
- Create tools to carry out the most frequently used processes or common requests such as reports, content publishing etc.
- Create tool to clear cache entries.
- Create admin access tool for server maintenance (server restarts).

F. Methods and process for smart ticket avoidance

Ticket avoidance and ticket reduction model involves taking many pro-active measures to minimize the tickets. It includes but not limited to the below given methods.

- Shift left of routine activities
- Perform root cause analysis for top tickets to understand the problem category and proactively improve the code quality for the identified areas
- Identify the frequent structured activities and automate them.

G. Methods and process for smart monitoring

We have defined the detailed monitoring setup in Fig. 4.

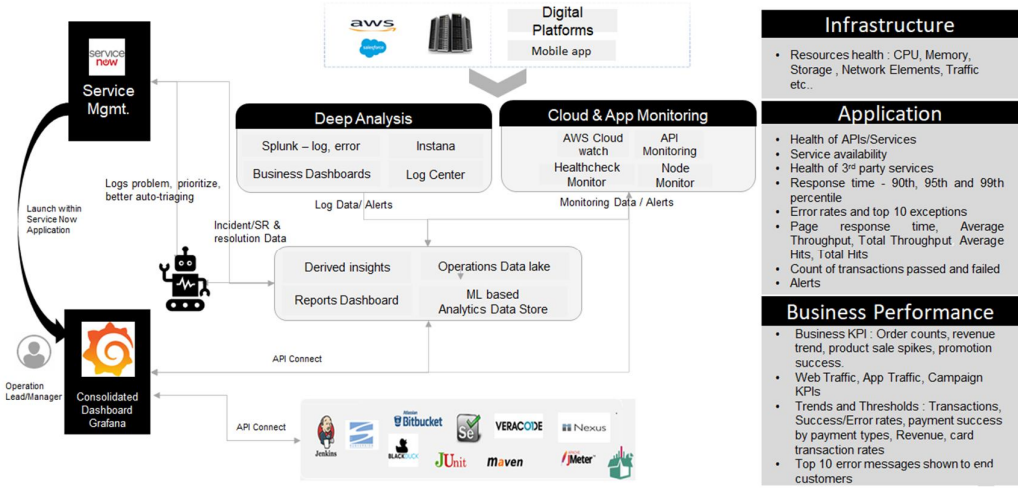

Figure 4. Monitoring Setup

The cloud monitoring setup uses the cloud based monitoring such as AWS cloudWatch and other monitoring systems such as API monitoring, health check monitoring, node monitoring to collect the detailed metrics related to health, performance, availability of the services and systems. We collect detailed performance metrics such as 90^{th} , 95^{th} percentile response time, throughput and such. As part of deep analysis we also collect the logs from these systems to understand the error rate. The collected metrics, insights and dashboard are shown to the user. The bot looks at the issues and logs the tickets along with appropriate priority.

As part of comprehensive monitoring, we also collect the test coverage reports, code quality reports, unit test results, build reports, functional reports from tools such as Maven, Junit, JMeter, Veracode, Selenium, Jenkins to intuitive dashboards.

Proactive Alert Monitoring to Predict & Prevent Failure Modes
We have depicted the pro-active monitoring setup in Fig. 5.

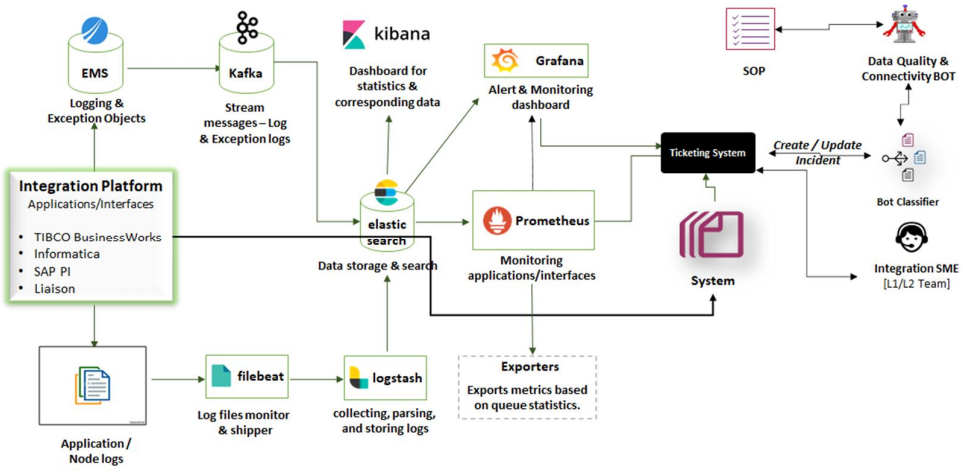

Figure 5. Proactive Monitoring

The main components of the solution are described below:

1. Enterprise Service Bus (ESB) Applications / Interfaces – Publish the logging & exceptions in two possible options to new Alert & Monitoring solution to elastic search. Application data via EMS topics/queues during the business transactions in each application/interface Application / Node logs – log files which contain all log level information into the application logs into file shares
2. Elasticsearch, Kibana, Beats and Logstash -- the Elastic Stack. Securely and reliably search, analyse, and visualize incoming data. This is doen to ,ainly to deal with data issues, connectivity issues metrics/ statistics & reports. Elastic search is leveraged to store, search & analyse the structured / un-structured data. Kibana is an open source data visualization plugin for Elasticsearch. It provides visualization capabilities on top of the content indexed on an Elasticsearch cluster. Users can create bar, line, scatter plots, pie charts and maps on top of large volumes of data
3. Prometheus (Alerts/ Monitoring dashboards) to provide multi-dimensional data model with time series data identified by metric name and key/value pairs. It provides PromQL, a flexible query language to leverage this dimensionality. It has no reliance on distributed storage; single server nodes are autonomous. Time series collection happens via a pull model over Hyper Text Transfer Protocol (HTTP). Pushing time series is supported via an intermediary gateway. Targets are discovered via service discovery or static configuration. Multiple modes of graphing and dash boarding support
4. Exporters : Java Message Service **(**JMS), Node, Blackbox – all the exporters are custom built to integrate with Elastic Search/Prometheus to retrieve the metrics, statistics, logs of ESB applications/interfaces

We should automate Monitoring Coverage and eliminate manual health checks of application to improve service reliability and for reducing the ticket handling efforts.

Monitoring metrics
We have depicted the monitoring metrics and SLAs in Fig. 6.
We should setup the monitoring infrastructure to monitor these metrics.

VI. RESULTS

We have noticed the below given improvements on the ticket management system by the adoption of the proposed methods:

- The self-service tools reduce on an average about 10-15% of tickets and helps us to complete the tasks quickly.
- ML based smart ticket management tools were able to auto resolve about 20% of the overall tickets leading to problem avoidance architecture
- The ML based models were able to pro-actively predict the problem with more than 90% accuracy leading to better maintenance.

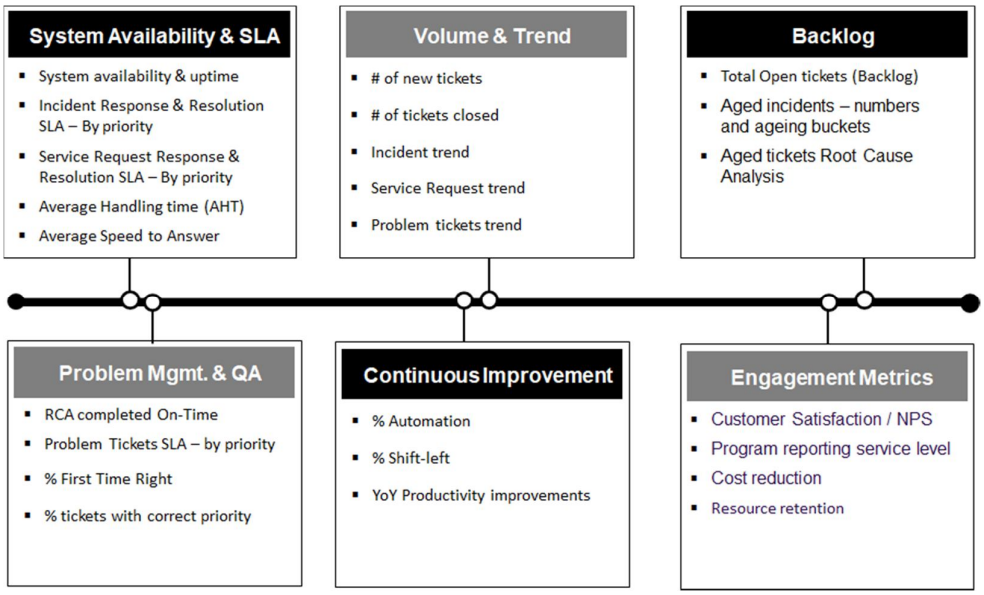

Figure 6. Monitoring metrics

By using the combination of self-service tools and ML based automation tools we were able to reduce the overall ticket volume by more than 30%.

VII. Conclusions

In this paper, we proposed ML based smart ticket management methods that trained the models to auto-classify tickets and auto remediate them. The ML based ticket resolution methods auto resolved about 20% of the tickets. We also proposed methods for smart shift left to improve the ticket resolution quality. We also proposed methods for ticket avoidance and smart monitoring to improve the response times of the ticket management system.

References

[1] Smart Ticketing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Service), By Product (E-Ticket, Smart Parking System), By System, By End Use, By Region, And Segment Forecasts, 2019-2026, Report ID: GVR-4-68038-063-7, Historical Range: 2015-2017

[2] M. Jäntti, "Defining Requirements for an Incident Management System: A Case Study," 2009 Fourth International Conference on Systems, 2009, pp. 184-189, doi: 10.1109/ICONS.2009.17.

[3] S. Metzger, W. Hommel and H. Reiser, "Integrated Security Incident Management -- Concepts and Real-World Experiences," 2011 Sixth International Conference on IT Security Incident Management and IT Forensics, 2011, pp. 107-121, doi: 10.1109/IMF.2011.15.

[4] C. Hove, M. Tårnes, M. B. Line and K. Bernsmed, "Information Security Incident Management: Identified Practice in Large Organizations," 2014 Eighth International Conference on IT Security Incident Management & IT Forensics, 2014, pp. 27-46, doi: 10.1109/IMF.2014.9.

[5] A. D. Nugraha and N. Legowo, "Implementation of incident management for data services using ITIL V3 in telecommunication operator company," 2017 International Conference on Applied Computer and Communication Technologies (ComCom), 2017, pp. 1-6, doi: 10.1109/COMCOM.2017.8167093.

[6] E. Lavrov, P. Paderno, O. Siryk, E. Burkov, N. Pasko and V. Nahornyi, "Decision Support in Incident Management Systems. Models of Searching for Ergonomic Reserves to Increase Efficiency," 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), 2020, pp. 653-658, doi: 10.1109/PICST51311.2020.9467991.

[7] M. A. Prihandono, R. Harwahyu and R. F. Sari, "Performance of Machine Learning Algorithms for IT Incident Management," 2020 11th International Conference on Awareness Science and Technology (iCAST), 2020, pp. 1-6, doi: 10.1109/iCAST51195.2020.9319487.

[8] Smart Ticketing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Service), By Product (E-Ticket, Smart Parking System), By System, By End Use, By Region, And Segment Forecasts, 2019-2026, Report ID: GVR-4-68038-063-7, Historical Range: 2015-2017.