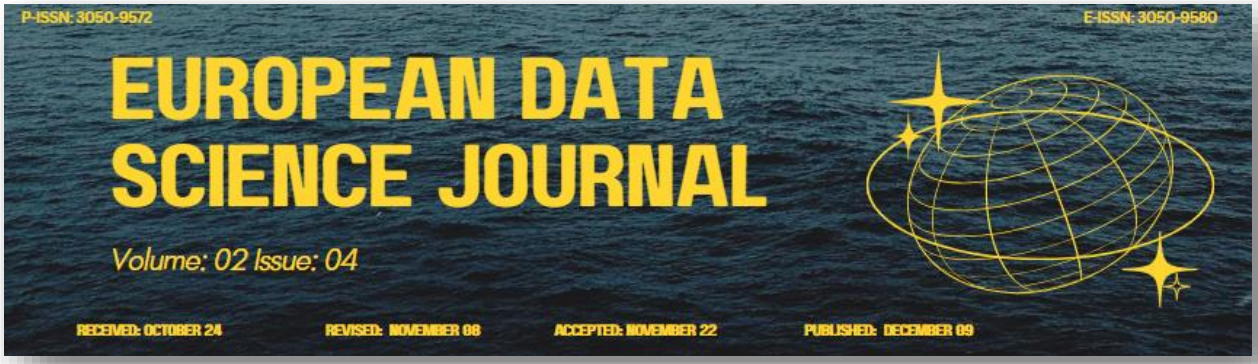


Machine Learning for Automated IT Support Ticket Routing

Emma Richardson
Asst Professor

Abstract

In a typical IT service desk, the volume of incoming service tickets can exceed several thousands per week. Handling these tickets relies on an underlying classification or routing process that maps each ticket to qualified expert resources for resolution. The classification can be based on a predefined taxonomy, which provides a multi-level hierarchy of various types of IT service requests and distinguishes between incidents, problems, changes, queries, and other service requests. A traditional approach for ticket classification is based on manually crafted rules. However, such a rules-based approach is not always feasible, and classifications can frequently need to account for very rare or less frequently seen tickets. Even a rules-based classifier cannot always guarantee the effective usage of resources.

Machine learning has increasingly been used for automating the classification operation and shifting the IT service desk to a model-driven setup, where the classical rules-based approach is enhanced by a learning-based operation that learns the mapping function. AI-driven intelligent classifiers can cater to frequently seen tickets and can learn either a one-vs-each approach for every type of service request or process the ticket in a multi-task learning setup.

Keywords : Enterprise IT Service Management, ITIL, Ticket Classification, Supervised Learning, Semi-Supervised Learning, Self-Supervised Learning,Data Foundations for Classification.

1. Introduction

Organizations rely on information technology (IT) to support operations, offer services to clients and partners, and enable collaboration among employees. IT service management (ITSM) refers to the set of activities that create value for customers by facilitating the use of IT in the organization. A service desk acts as a single point of contact for users and IT professionals and serves as the interface between these groups for rendering IT services. A range of processes coordinate IT resources to respond efficiently to service requests, incidents that disrupt the normal functioning of services and IT resources, and problems that are root causes of one or more incidents.

AI and machine learning offer companies tools to optimize their business operations. In the context of ITSM and service desk operations, such technologies can relieve staff members of tedious and repetitive tasks. Although common AI applications involve advanced capabilities in natural language generation, synthesis, and understanding, these capabilities can also be harnessed for ticket classification—the categorization of tickets from the service desk. The classification process is essential for routing requests to appropriate teams for resolution, yet it remains a potential source of bottlenecks and high workloads, both for query authors and support staff. An AI-based ticket classifier can automate this part of the ticket-handling process, thereby shortening response and resolution times, improving service quality, and enabling organizations to serve more users with the same resources.

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24 REVISED: NOVEMBER 08 ACCEPTED: NOVEMBER 22 PUBLISHED: DECEMBER 09

Information technology (IT) plays a vital role in supporting organizational operations, enabling service delivery to customers and partners, and facilitating collaboration among employees. To manage these technological resources effectively, organizations implement IT service management (ITSM), which consists of structured activities designed to create value by ensuring the efficient use of IT services. A key component of ITSM is the service desk, which functions as the primary point of contact between users and IT professionals, handling service requests, reporting incidents that disrupt normal services, and identifying underlying problems that may cause recurring issues. With the increasing volume of service desk tickets, artificial intelligence (AI) and machine learning (ML) technologies are being adopted to enhance efficiency. One important application is ticket classification, where AI systems automatically categorize service requests and incidents based on their content. By automating this process, organizations can quickly route tickets to the appropriate support teams, reducing manual workload, minimizing delays, and improving overall service quality. As a result, AI-powered ticket classifiers help streamline ITSM operations, shorten response and resolution times, and allow organizations to support a larger number of users without requiring additional resources.

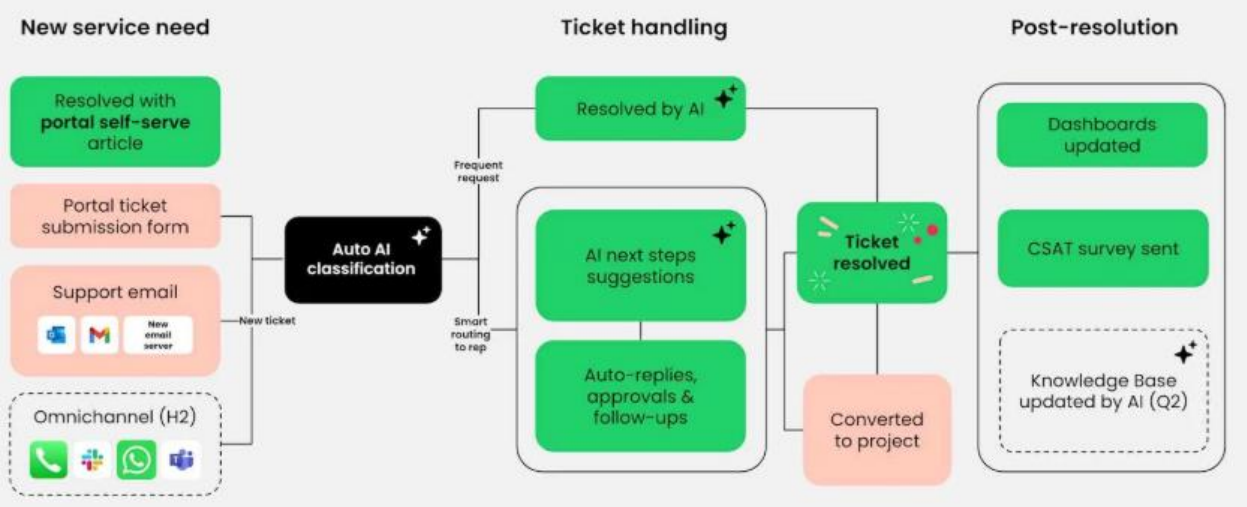


Fig 1: AI ticketing system will transform

1.1. Background and Significance

Despite the rapidly growing adoption of IT-enabled services for delivering business processes and applications, traditional service desk operations have not significantly improved in terms of efficiency or quality of service. High-quality historical data are collected for each ticket, and service-level agreements (SLA) are associated with tickets in the form of manual or automated routing. Nevertheless, approximate solutions based on trigger/hybrid steering or manual steering are still commonly detected. Consequently, attached service desks continue to be viewed as a burden rather than a resource or driver, and SLAs become a tick box for coverage rather than a meaningful measure of service delivery.

This challenges cost, productivity, and ultimately satisfaction-related KPIs. Implementing an AI-driven, intelligent ticket classifier provides one part of the solution and can operate alongside adaptive/automated steering and hybrid systems for both incident and service request tickets, improving the utilization and effectiveness of these important operations. AI-assisted

classification complements human-in-the-loop approaches, allowing humans to focus on complex or non-standard tickets while the bulk of tickets receive fact-based classification, potentially reducing dependence on trigger/conditional steering and improving operations at all layers of the architecture.

Equation 1: Text representation: TF-IDF derivation

Step 1: Term frequency. A raw term-frequency feature is tf_{ij} = n_{ij}. A normalized variant divides by ticket length L_i = sum_{j=1}^V n_{ij}, so tf_{ij} = n_{ij} / L_i.
Step 2: Document frequency. Let df_j be the number of tickets containing term j. If the corpus has N tickets, then idf_j = log((N+1)/(df_j+1)) + 1. The +1 terms avoid division by zero and keep the value finite.
Step 3: TF-IDF weight. The final feature for term j in ticket i is x_{ij} = tf_{ij} * idf_j.
Step 4: Vector form. The ticket becomes x_i = (x_{i1}, ..., x_{iV})^T. This representation increases the influence of rare but discriminative words such as a product name or a subsystem identifier, while decreasing the influence of ubiquitous words such as 'issue', 'help', or 'please'.

1.2. Research design

The investigations episodically test AI mothods for classifying enterprise IT tickets, drawing upon natural language and operational log data spanning multiple years. The practical utility rests on its structuring of a novel, functionally motivated taxonomy, along with the sound labelling schemes and considerable efforts directed towards supporting data. It also embarks upon supporting classic exploratory-use cases, such as rapid, accurate uncovering of service-engineering incompetences and the service environment's operational health.

The fully fledged data foundation enables evaluation that spans classical supervised ML, data-deficient semi-supervised paradigms, and general data-efficient self-supervised mechanisms. This breadth is important real-world deployment, given the invariably skewed label distribution in ticket routing scenarios. Supervised ML groundwork establishes industry benchmarks, while label-propagation and contrastive-learning systems further probe data-sparsity's practical limits. Additional alternative directions consideration includes transfer operations across not just the Space of task but also label sets, in pursuit of fully autoregressive models the setup's hierarchy.

2. Foundations of IT Service Management and Ticket Classification

The theoretical foundations of IT service management and ticket classification tasks establish the theoretical basis and practical context. The Information Technology Infrastructure Library (ITIL) best-practice framework for IT service management is profiled in a manner tailored to enterprise service management. A service desk operation, the classification of requests and incidents as a driver for further process stages, and the difference between service requests, incidents, problems, and changes are explained.

Service desks play an important role in the ITIL framework for IT service management. They are the first point of contact for users to log requests for service. Service requests and incidents are classified throughout their lifecycle. Accurate classification in these stages supports subsequent activities, such as assignment, prioritization, escalation to specialist support groups, and fulfillment or resolution. ITIL describes a range of possible service request types. When incidents are classified as problems, their

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24 REVISED: NOVEMBER 08 ACCEPTED: NOVEMBER 22 PUBLISHED: DECEMBER 09

underlying causes are identified so that permanent resolutions can be implemented. Similarly, changes can be classified as standard, emergency, or normal, the latter category being further grouped into types.

Service requests aimed at fulfillment are predominantly handled by the service desk. When these requests fall outside of the standard service catalog, they require consideration of the full life cycle—the development of a new offering, fulfillment, and either support or retirement. A service desk forms part of the support organization but has a broader remit than restoring normal service operation while minimizing impact on users. Following a well-defined service design phase, support teams deliver services according to defined service level agreements (SLAs) and operational level agreements (OLAs).

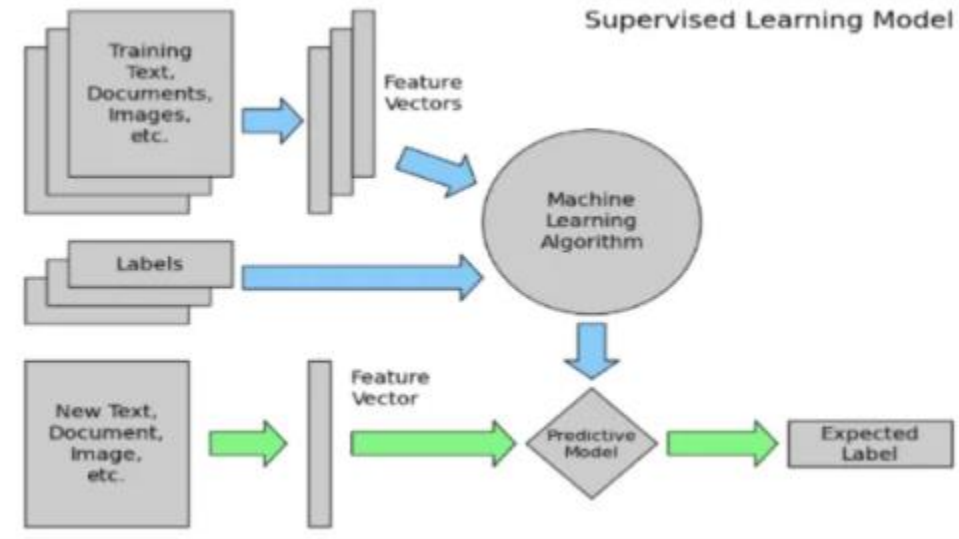


Fig 2: IT Service Management and Ticket Classification

2.1. ITIL and Service Desk Fundamentals

The Information Technology Infrastructure Library (ITIL) framework prescribes a set of best practices aligned with the needs of the business and aimed at supporting its core processes. At a high level, ITIL describes five processes: Service Strategy translates business interests into IT language and ensures IT services remain aligned with the business; Service Design builds new and modified services intended to satisfy the needs laid down in Service Strategy; Service Transition realizes the services designed in the previous phase; Service Operation delivers and supports services in accordance with the requirements and objectives defined in Service Strategy, Service Design, and Service Transition; and Continual Service Improvement identifies and implements improvements to IT services that support the ITIL processes. Supporting these five processes are internal and external procedures for Service Asset and Configuration Management, Knowledge Management, Availability Management, Capacity Management, Service Level Management, Financial Management, IT Service Continuity Management, Demand Management, Information Security Management, Supplier Management, and Release and Deployment Management.

Within the scope of IT service management, Service Requests represent a certain type of service interaction with the end-users and business units. ITIL identifies a Service Desk function, an operational team responsible for managing and fulfilling Service Requests and restoring failed IT services as quickly as possible. Service Requests can be categorized as an Incident, Problem, Change, or Expense. Incidents do not result from a failure in a Quality Control or Preventive Control procedure and require urgent correction. Problems result from a failure in a Quality Control or Preventive Control process and require corrective action. Changes are transactions that will incur extra expense, and Expense transactions do not affect the operational status of the IT services.

Equation 2: Multinomial Naive Bayes derivation

For class k, let prior probability be pi_k = P(y=k). Under the multinomial Naive Bayes model, the word counts in a ticket are assumed conditionally independent given the class.
Step 1: Bayes rule. P(y=k|x) = P(x|y=k) P(y=k) / P(x). Since P(x) is the same for every class, classification only needs the numerator.
Step 2: Likelihood. If x = (x_1,...,x_V) are token counts, then P(x|y=k) is proportional to product_{j=1}^V phi_{kj}^{x_j}, where phi_{kj} = P(term j | y=k).
Step 3: Parameter estimate with Laplace smoothing. If c_{kj} is the total count of term j across all training tickets in class k and C_k = sum_j c_{kj}, then phi_{kj} = (c_{kj}+alpha)/(C_k + alpha V).
Step 4: Log-posterior score. Taking log avoids numerical underflow: score_k(x) = log pi_k + sum_{j=1}^V x_j log phi_{kj}.
Step 5: Decision rule. Predict class y_hat = argmax_k score_k(x). Thus the model adds a prior term and then sums a contribution from each observed word.

2.2. Classical Ticket Routing and Its Limitations

Service desk ticket routing has traditionally employed rules or heuristics to assign incoming tickets to support agents. For example, log messages containing specific keywords might be directed to a specific team, with support agents often forced to pick new tickets manually. Indeed, this is often the only viable solution for tickets appearing rare within the service desk history, so that even classifiers trained in a one-vs-rest approach may prove inadequate. To address such shortcomings, AI-driven classification has become increasingly popular.

Automated classifiers capable of processing free-text ticket descriptions and suggesting appropriate departments or agents for assignment are theoretically straightforward to train. The machine learning literature is rich with approaches, including rule-based systems, Naïve Bayes, logistic regression, and multinomial support vector machines (SVMs). Furthermore, the introduction of large public datasets, such as the Stack Overflow question-answer pair repository, along with advancements in deep learning, have fostered interest in applying natural language processing (NLP) techniques. Yet developing effective classifiers for enterprise service desk problem tickets remains an open area of study. Labeling complexity, ticket sparsity and hence statistical power, and domain transferability of machine learning algorithms have all been cited as limiting factors.

3. Data Foundations for Classification

Accurate classification requires adequate data, especially when machine learning approaches are employed. Several sources contribute to the data foundation of IT Ticket Classification and together address a wide range of information that can influence

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24 REVISED: NOVEMBER 08 ACCEPTED: NOVEMBER 22 PUBLISHED: DECEMBER 09

model performance. Ticket metadata summarizes the request itself and includes fields such as the title, description, type of request, source of request, status, and resolution. Service Level Agreement conditions attached to individual requests allow model predictions to be aligned with business priorities. Additional information is available in operational logs that capture resolution activities performed by support analysts. In organizations where end-users interact with support personnel through chat, transcripts of the conversations may provide further insights. When users attach documents, screen shots, or logs to their requests, their content may also enhance comprehension. Finally, organizational metrics tracked over time provide feedback on broader operational efficiency, effectiveness, quality, and security that can be leveraged to prioritize model predictions. Such metrics capture become important when AI outcomes are integrated into business processes, such as automation or dialog scripts.

As discussed in Section 2.2, the primary class label takes the form of a predefined taxonomy that may include multiple levels of hierarchy. A multi-label annotation approach captures whether one or more service groups or subcategories are appropriate for the request. Inter-annotator agreement on the proposed schemes is also tracked to support overall quality and consistency of labeling. Where possible, generated labels are evaluated against a gold standard to assess the capability of the proposed labeling process. Further confidence in the quality of the class labels may be achieved through alignment with an established ontology or systematic taxonomy. Together, these aspects form the foundation for defining the operational data assets and processes necessary to support effective training, validation, and evaluation of classification models.

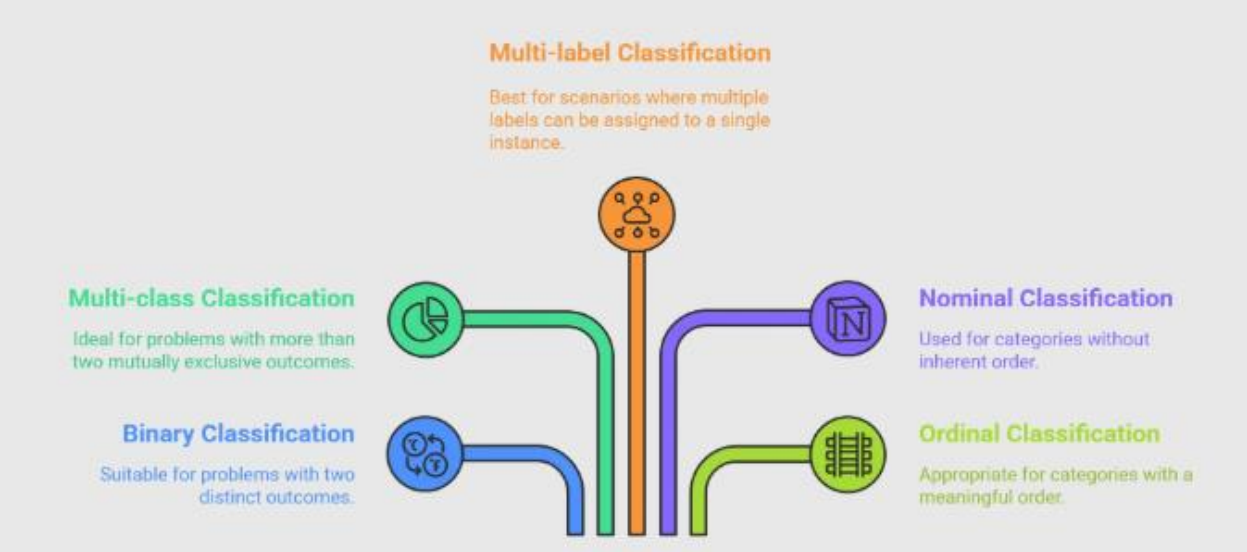


Fig 3: Data Foundations for Classification

3.1. Data Sources in Enterprise Service Management

Service tickets contain a wealth of textual and non-textual data, including information that is seldom seen in ticket classification applications. Organizations gather information from the ticket subject, description, chat sessions, attachments, comments, and status logs. These pieces of information may serve as useful indicators for classification. Additional features, such as service-

level agreements (SLAs) and other operational metrics, may also provide valuable signals. The ticket subject and description fields are the main sources of textual information and the sources commonly used in ticket classification tasks. Yet several additional features over and above the subject and description are also provided by IT service management (ITSM) solutions and can be helpful when coding classification tasks.

The number of tickets belongs to another class of indicators and can be analyzed by exploiting other sources of information. For any ticket, the number of tickets that satisfy a certain condition can be counted as a feature. Such features can help ensure that any type of classification task lends itself to anomaly detection. Besides standard ticket information, the logs that record the modifications made to the ticket can also contribute to a better understanding of the ticket evolution. Because these logs are composed mainly of short textual data, natural language processing techniques can help extract insights from comments. The logged status changes, which mark important transitions in the ticket lifecycle, may also serve as useful indicators.

3.2. Labeling Schemes and Ontologies

A software application must necessarily provide, in addition to the visible interface offered to users, an interface to other software applications that wish to make use of it. For the case of Enterprise Service Management, various CMDBs exist, holding information about the IT landscape of the enterprise and helping the decision-making process with respect to e.g. What is the impact of this IT incident? What is the risk of applying this change? Where are the end-users located for this service request so that the IT department can plan scheduling and logistics? Ticket handling could probably be automated through other applications (bots), but an AI-helper for humans in decision-approving or -classifying tasks could be developed and trained, in order to take the human knowledge in an automated way. Scripted ChatOps and even generator functions could be applied through these AIs.

Besides the standard fields of most tickets, that contain information like incident description, request title, affected service or asset, SLAs and support team there are three other hidden sources of information: chat logs between the user and the IT department (in some cases the user has to request for a change through a conversation), internal chat logs between the IT department and the assigned expert related to the ticket being processed, and the ticket itself. The chat log related to that specific ticket would carry information like "Hello, did you manage to connect the projector? I can see that it is malfunctioning and I can make a replacement" or "Hello, did you manage to connect the projector? I can see that it is malfunctioning and I can make a replacement" as part of the discussion."

Equation 3: One-vs-rest logistic regression derivation

The article discusses one-vs-all supervised classifiers. For class k, define binary target y_i^{(k)} in {0,1}, equal to 1 if ticket i belongs to class k and 0 otherwise.
Step 1: Linear score. s_i^{(k)} = w_k^T z_i + b_k.
Step 2: Sigmoid probability. p_i^{(k)} = sigma(s_i^{(k)}) = 1/(1 + e^{-s_i^{(k)}}).
Step 3: Binary cross-entropy loss for class k. L_k = - sum_{i=1}^N [y_i^{(k)} log p_i^{(k)} + (1-y_i^{(k)}) log(1-p_i^{(k)})].
Step 4: Gradient with respect to the score. Because d sigma(s)/ds = sigma(s)(1-sigma(s)), differentiating the binary cross-entropy gives dL_k/ds_i^{(k)} = p_i^{(k)} - y_i^{(k)}.
Step 5: Gradient with respect to parameters. dL_k/dw_k = sum_i (p_i^{(k)} - y_i^{(k)}) z_i and dL_k/db_k = sum_i (p_i^{(k)} - y_i^{(k)}).

Step 6: Update rule. With learning rate eta, gradient descent updates w_k <- w_k - eta dL_k/dw_k and b_k <- b_k - eta dL_k/db_k.
Step 7: Final multiclass prediction. After training K one-vs-rest models, the class with largest score or probability is selected: y_hat = argmax_k p_i^{(k)}.

4. Modeling Approaches for Ticket Classification

Specific AI techniques and their applicability are examined in some detail. Selecting appropriate modeling approaches is critical for achieving good classification performance, especially for sub-problems with limited labeled data or with coarse granularity. Supervised learning with classical text classifiers like logistic regression, XGBoost, or linear Support Vector Machines is suitable for the central monolingual text-based classification task. A rich set of features built on both text and auxiliary data sources supports these classifiers. Because the ticket data is naturally multi-class, a one-vs-all strategy is employed. For traditional classifiers, additional model ensembles can be used.

Semi-supervised and self-supervised methods offer data-efficient ways to tackle hard problems. A small number of English-Russian bilingual tickets can be exploited to build a minimalist parallel corpus allowing for multilingual training and some degree of zero-shot transfer. Domain-adapted label propagation aids ticket routing, while contrastive learning uses cross-modal audio-vision pairs to learn a meaningful representation for the agent action prediction task. Finally, semi-supervised learning with a mix of labeled and unlabeled chat logs is applied to build a nested model for deciding why a helpdesk agent approved or rejected a user request.

Semi-supervised and self-supervised methods provide data-efficient strategies for addressing complex machine learning problems, particularly when labeled data is scarce. By leveraging a small set of English–Russian bilingual support tickets, it is possible to construct a minimalist parallel corpus that enables multilingual model training and facilitates a degree of zero-shot transfer across languages. For ticket routing tasks, domain-adapted label propagation can effectively utilize relationships between labeled and unlabeled tickets to improve classification accuracy. In addition, contrastive learning can exploit cross-modal audio-vision pairs to learn meaningful representations that support agent action prediction. Finally, combining labeled and unlabeled chat logs in a semi-supervised framework enables the development of a nested model that can infer the reasoning behind helpdesk agents’ decisions to approve or reject user requests, improving interpretability and decision support in customer service systems.

P-ISSN: 3050-9572 E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24 REVISED: NOVEMBER 08 ACCEPTED: NOVEMBER 22 PUBLISHED: DECEMBER 09

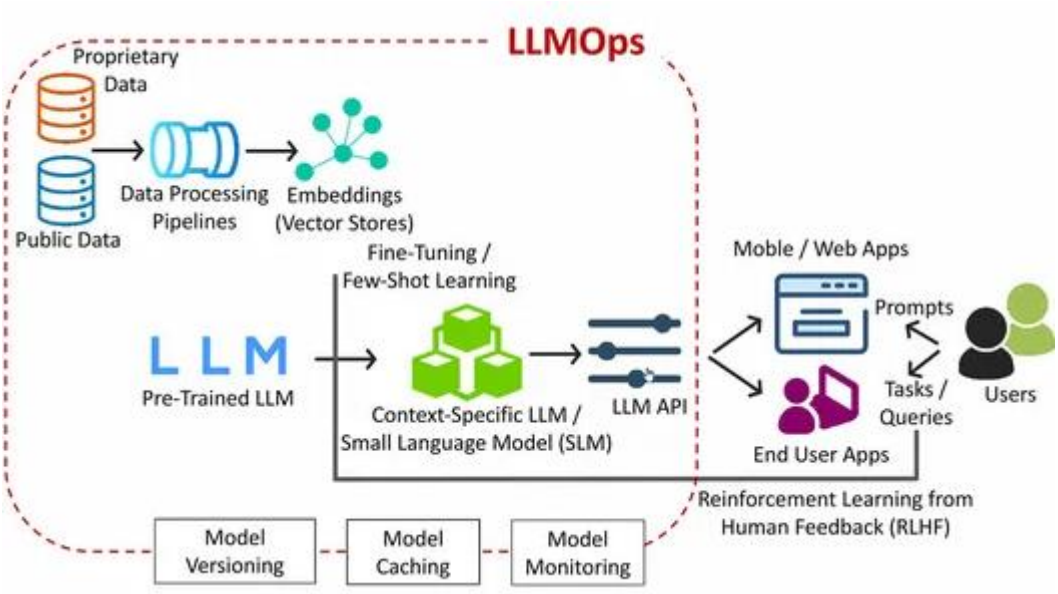


Fig 4: Modeling Approaches for Ticket Classification

4.1. Supervised Learning Techniques

AI techniques for intelligent ticket classification fall broadly into supervised and non-supervised approaches. Although the latter classes typically require less labeled data, they also benefit from annotation. Semi-supervised and self-supervised techniques are discussed in separate sections. This section focuses on the key ingredients of supervised learning, consolidating content from recent studies.

Text representation is pivotal in supervised approaches, which address ticket classification as a multi-class or multi-label problem depending on whether the output is a single label, drawn from a single label hierarchy, or a set of labels or multiple hierarchies. A ticket's title or summary is commonly considered the primary input. Heuristic feature engineering also plays an important role, with custom features being added to the text representation, often including counts of past tickets recorded over the history of the service management platform. A histogram of records of past tickets assigned to each operational group also proves effective. Other feature sources include chat conversation logs and attachment filenames, although their utility often remains limited since only a small proportion of tickets capture such information. All features are then combined into an integrated representation.

A wide variety of classifiers have been explored for the task, from classical approaches such as naive Bayes and support vector machines to modern deep-learning methods like transformers, which often achieve state-of-the-art performance. The ticket data associated with enterprise information technology groups is usually highly imbalanced for both the primary label and those in the auxiliary labels, with relatively few tickets attributed by the operational teams and only a few assigned to each auxiliary label. A set of common text-vectorization and model-training pipeline steps is therefore required, combined with techniques designed specifically for class-imbalance challenges.

4.2. Semi-Supervised and Self-Supervised Methods

The availability of extensive enterprise IT ticket corpora—often exceeding millions of records—make data-hungry learning methods attractive when only a small fraction is labeled. Semi-supervised learning techniques exploit the large volume of unlabeled data alongside a limited amount of labeled examples. Label propagation on the known labels of the limited labeled data can identify the labels for the other incoming instances and has demonstrated promise on task-specific taxonomies with a small height. The technique has also been extended to entail different aspects of a ticket simultaneously, permitting the learning of a multi-label representation from costly annotating efforts. Contrastive methods attempt to learn good representation for target tasks by contrasting not just the instance within the task but also other randomly sampled instances.

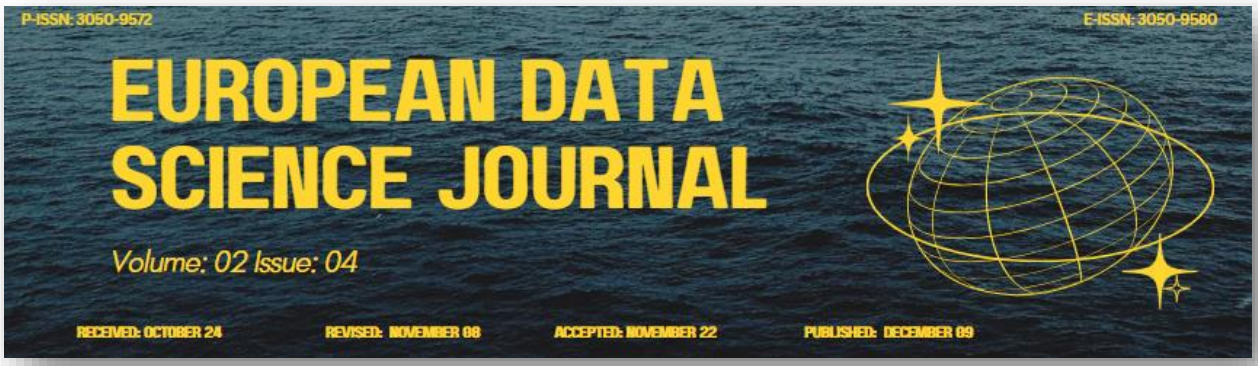
Self-supervised learning methods try to leverage easily available supervision signals within the data itself. Domain-adaptation approaches tackle the natural domain-shift problem while training models on underlying-domain collections instead of the application-specific ticket data. Visual class-imbalance mitigation strategies can also be successfully borrowed from standard zero-shot learning to aid tasks which display classical class-imbalance scenarios. Finally, the requirement for prompt engineering in the existing LLM setups has also been highlighted and partially mitigated for vertically structured datasets with the creation of topic-wise attributes. The development of knowledge distillation on the multimodal space is another promising future direction which accounts for the increasingly multimodal nature of real-world data.

Equation 4: Linear SVM derivation

For a binary one-vs-rest subproblem, use labels y_i in $\{-1,+1\}$. The decision function is $f(z_i)=w^T z_i + b$.
Step 1: Margin condition. Correct classification with margin requires $y_i (w^T z_i + b) >= 1$.
Step 2: Slack for non-separable data. Introduce $xi_i >= 0$ with $y_i (w^T z_i + b) >= 1 - xi_i$.
Step 3: Primal optimization. Minimize $(1/2)\|w\|^2 + C \sum_i xi_i$ subject to $xi_i >= 0$ and the margin constraints above.
Step 4: Equivalent hinge-loss form. The optimization becomes $\min_{w,b} (1/2)\|w\|^2 + C \sum_i \max(0, 1 - y_i (w^T z_i + b))$.
Step 5: Subgradient. If $1 - y_i f(z_i) > 0$, the sample contributes gradient $-C y_i z_i$ to w and $-C y_i$ to b; otherwise it contributes 0 to the hinge term. The regularizer always contributes w.
Step 6: Prediction. Choose class by the highest one-vs-rest margin.

5. System Architecture for an AI-Driven Classifier

The end-to-end architecture for an AI-driven ticket classification system is described here, covering components for data ingestion, preprocessing, model training, evaluation, deployment, and operational management. Real-time ticket classification can significantly improve IT service management by enabling automation capabilities, reducing ticket-routing overheads and response times, and augmenting knowledge bases. An ideal system runs in a production environment, continuously processing incoming tickets and automatically classifying them into predefined categories, which may be used to direct tickets to correct teams, route them to the appropriate support levels, suggest solutions to the service desk, or trigger automated workflows. Data ingestion, cleaning, and normalization, including anonymization of sensitive information, provide an upstream pipeline for the system. Feature extraction prepares data for training, which can be conducted offline or incrementally adapted to capture the latest patterns in the data.



Offline model training is usually done outside production systems but is critical to operationalize AI-classification capabilities. Training organization can differ from normal IT operations because model training typically consumes considerable resources. Therefore, intensive training work is often scheduled during off-peak hours and performed with minimal operational impact. Distributed training is a common practice to accelerate the process. Validation is crucial for evaluating model performance and deploying only those models that meet the quality targets. Continuous monitoring helps ensure that deployed models do not drift over time. A/B testing enables new models to be further validated under realistic workloads. When validated, models can be seamlessly rolled out into production. These processes are applied to the classification model but can also be generalized to other models within an ITSM setting.

5.1. Data Ingestion and Preprocessing Pipeline

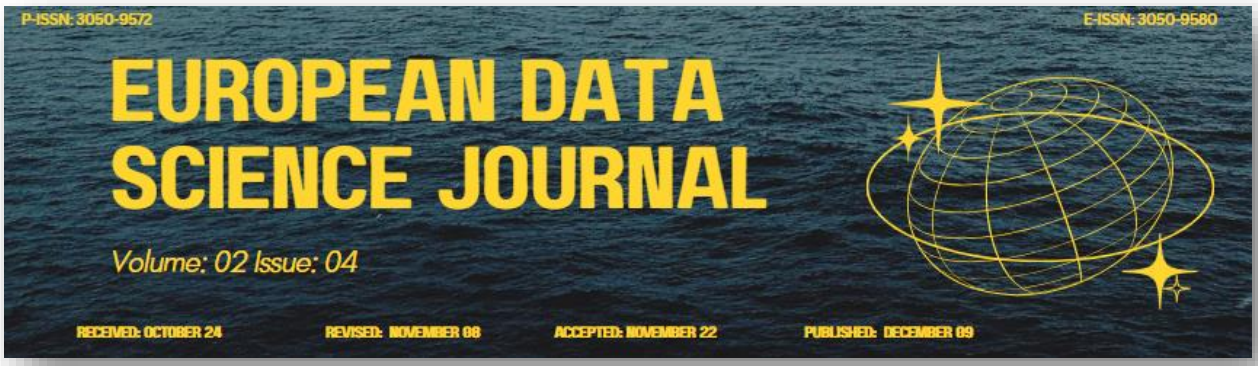
The proposed intelligent classification model requires preprocessed data tailored to its downstream learning objective. A data ingestion and preprocessing pipeline fitting the overall architecture provides the required inputs. Configured as an extraction–transformation–loading (ETL) toolchain, the pipeline collects, cleans, normalizes, and augments data into the desired operational schema. Each subprocess is briefly explained.

The system archetype provides enterprise organizations with data-driven predictive insights on ticket volume and ticket category distribution across future time periods. Such predictions are instrumental for capacity planning and forecasting service desk costs. However, the communication provided by the requestors during the ticket-creation phase is often noisy. Misspelled words, language discrepancies, abbreviations, omitted information, and nonsensical phrases can all be detected. Therefore, formal language is utilized to improve the classifier's perception. Although natural language processing models offer spelling-checking capabilities, they generally lack support for acronyms or foreign languages. Hence, often-studied spell-checkers are employed. Moreover, simple language translation and back-translation to English are applied. A popular language-detection model is used to identify the language of each request. Requests in other languages are translated into English.

To improve the training of word-embedding-based representations, the textual data is further augmented by applying the back-translation method. Sentences are translated into a random foreign language and back-translated into English. Chat-based data corresponding to the same natural language question are also used to generate similar training samples for the word-embedding-based representation. A well-trained model initialized with a domain-oriented word-embedding-based representation can also capture such text. Moreover, privacy concerns regarding WhatsApp and iMessage chat conversations are mitigated by performing a privacy-preserving sanitization task. Because user identities are linked to privacy concerns, the labeled user IDs are replaced by the category name of the user.

Equation 5: Semi-supervised label propagation derivation

Let all tickets (labeled and unlabeled) be nodes in a similarity graph. Let W be the affinity matrix, where W_{ij} measures the similarity between tickets i and j. Define the degree matrix D with D_{ii} = sum_j W_{ij}.
Step 1: Row-normalized transition matrix. S = D^{-1} W. Each row sums to 1, so S behaves like a random-walk transition matrix.
Step 2: Label matrix. Let Y be an n x K matrix whose rows contain one-hot labels for labeled tickets and zeros for unlabeled ones.
Step 3: Propagation update. Initialize F^{(0)} = Y and iterate F^{(t+1)} = alpha S F^{(t)} + (1-alpha) Y, where 0 < alpha < 1.



Step 4: Fixed point. At convergence F = alpha S F + (1-alpha) Y. Rearranging gives (I - alpha S)F = (1-alpha)Y, hence F = (1-alpha)(I - alpha S)^{-1}Y.

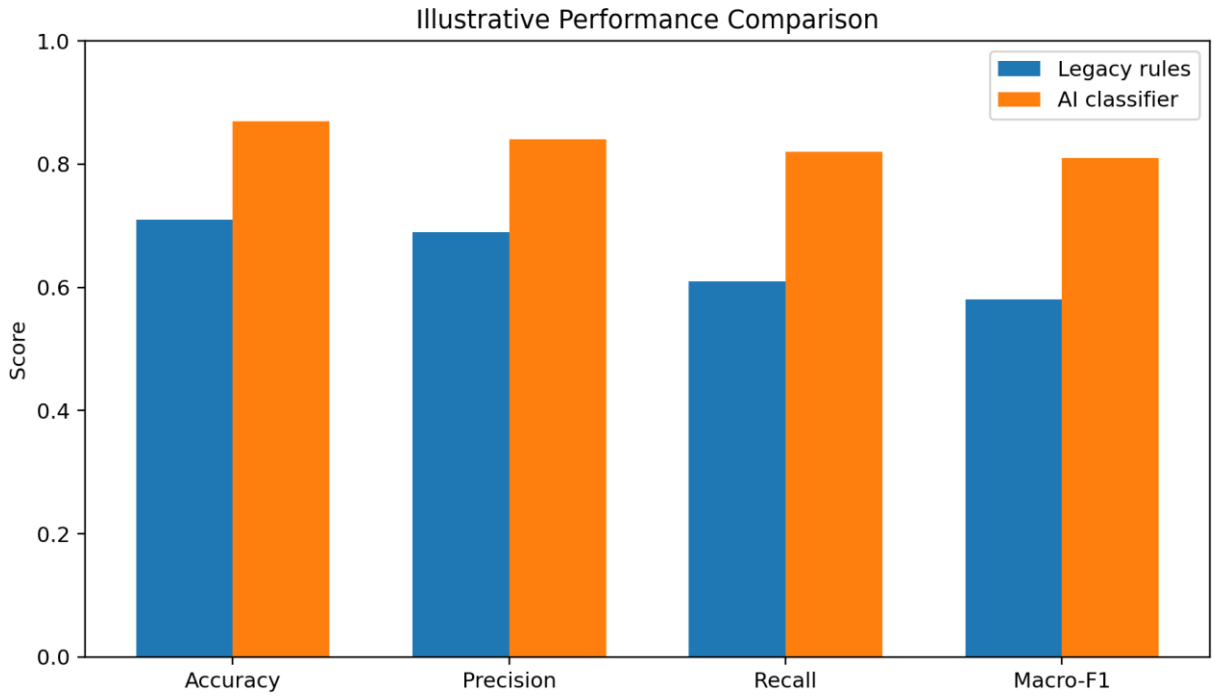
5.2. Model Training, Evaluation, and Deployment

After a model is trained and validated on the training and validation sets, it is ready for operation. Most class-agnostic classification performance metrics, such as accuracy, F1 score, precision, and recall, can be used to evaluate its performance. Metrics are computed on the test and validation sets to gauge the model's performance. In production scenarios, metrics are calculated as classes with ever-increasing amounts of data owing to different services being available. Progressively smaller chunks of the test set can be used to ensure that the classifier is indeed generalizing to new requests and that the services are capable of handling such requests. For class-aware metrics, the classifier can be validated using a support threshold that restricts the computation to classes that are sufficiently represented in the test set, e.g. with at least 100 requests.

Operation in a production environment is typically done via an A/B test design with the new classifier exposed to end users only some cases and a random control continuing to work with the legacy approach. The actual operational rollout of the classifier has to take into account the possible supporting infrastructure, which can range from the creation of enabling rules and accepting dummy services able to catch and classify all tags to developing a command-line interface for other people to test it and proposing a simple natural language processing request wording without the need for technical terms. As the framework handles the request as any other service is capable of throwing a response back; once acceptable returns are provided, the classifier can simply be trained to do all the rest.

In a production environment, deploying a new classifier is typically carried out through an **A/B testing strategy** to ensure reliability and minimize risk. In this approach, the new classifier is exposed to only a subset of users or requests, while a randomly selected control group continues using the legacy system. This allows developers to compare performance, accuracy, and user impact before a full rollout. The operational deployment must also consider the supporting infrastructure required to make the classifier functional and accessible. This may include creating enabling rules, implementing dummy services that capture and classify incoming tags, and developing a command-line interface so that other teams can test the classifier easily. Additionally, designing a simple natural language interface allows users to submit requests without needing technical knowledge. Since the framework processes these requests like any other service, it can return responses seamlessly. Once the system consistently produces acceptable outputs, the classifier can be further trained and refined to handle the remaining tasks automatically, leading to a smooth and scalable integration into the production workflow.

P-ISSN: 3050-9572
E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24
REVISED: NOVEMBER 08
ACCEPTED: NOVEMBER 22
PUBLISHED: DECEMBER 09



6. Operational Considerations and Best Practices

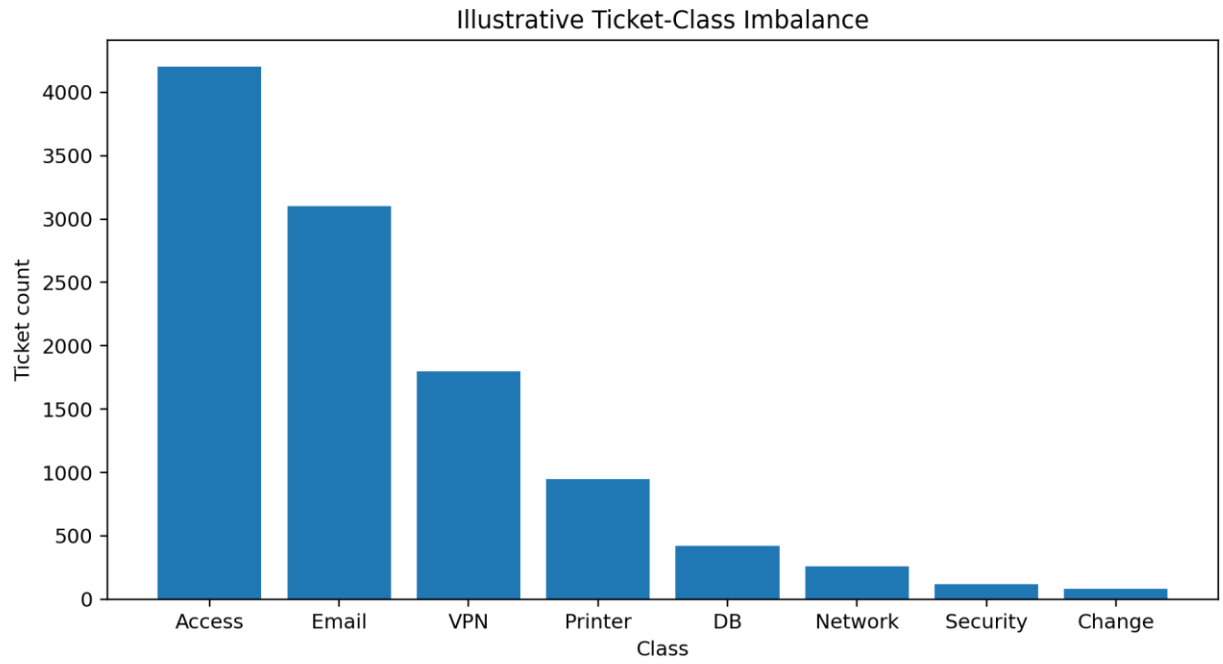
Deploying any machine learning model entails certain technical and non-technical aspects that need to be taken into account, including preparation of the data-collection pipeline for inference, training, validation of the model, monitoring of performance, revisiting the model, and, lastly, the governance of the entire process. Therefore, the use of a neural model to classify service tickets should also consider these aspects to avoid problems once the model is effectively running into production. Addressing these points upfront can lead to a smoother deployment of the model and to a minimization of headaches afterwards. When planning how to exploit an AI-driven classification system, three questions generally arise: i) how to deal with class imbalance and rare classes, ii) how frequently the system should be retrained, and iii) how to govern the development and maintenance of the system.

In the context of enterprise service management, class imbalance is mainly due to the presence of numerous standard tickets. These tickets represent the service requests that are frequently requested by customers, such as access requests to applications and servers, software installations, VLANs, and so on. While these requests can be accurately classified with high confidence, most of the remaining classes are very far from being as popular. In some cases, a class can be deemed rare with only a few hundreds of instances. However, a conventional classifier will never classify any new incoming instance into such a rare class,

P-ISSN: 3050-9572
E-ISSN: 3050-9580
EUROPEAN DATA SCIENCE JOURNAL
Volume: 02 Issue: 04
RECEIVED: OCTOBER 24
REVISED: NOVEMBER 08
ACCEPTED: NOVEMBER 22
PUBLISHED: DECEMBER 09

leading to a serious issue if these instances actually need to be routed to the corresponding business unit. Nevertheless, anomaly detection solvers can help here, and their integration with the neural model is highly desired.

In enterprise service management, **class imbalance** commonly arises because a large portion of incoming tickets belong to a small set of frequently requested services known as *standard tickets*. These include routine requests such as application or server access, software installations, and network configurations like VLAN provisioning. Since these requests occur very often, machine learning classifiers can learn their patterns easily and classify them with high confidence. However, the remaining ticket categories are significantly less frequent, and some may contain only a few hundred examples. Such rare classes pose a challenge for conventional classifiers, which tend to favor the majority classes during training and therefore may fail to correctly identify or route rare but important requests. As a result, tickets that actually belong to these uncommon categories might never be predicted correctly, potentially delaying their routing to the appropriate business unit. To address this limitation, anomaly detection techniques can be integrated with neural classification models. These techniques help identify instances that deviate from common patterns, enabling the system to flag or detect rare ticket types more effectively and improving the overall robustness of the service management workflow.



6.1. Handling Class Imbalance and Rare Tickets

Many deployed classifiers exhibit class imbalance, most classified tickets belonging to frequent classes. Investigators reduce training examples for dominant classes and increase rarer classes—resampling helps mitigate overfitting and raises overall performance, yet can hamper accuracy on low-frequency classes. For single-label schemes, cost-sensitive objectives penalize

mistakes differently according to class distribution; these pursuits may be adapted to multilabel cases, raising costs of false negatives in both low-frequency and relevant classes.

Risk-aware classifiers offer another avenue for tackling skewed classes or rare instances. Commercial applications already exploit these techniques for credit fraud detection; an analogous notion applies here. Fielded classifiers highlight tickets requiring special handling—for example, fraud detection or risk assessment for support requests involving Toronto or Canada geese within the Bahamas. Such tickets often constitute anomalies with respect to the wider input space yet receive little or no training data. Despite these limitations, classifying these instances enables efficient handling. Further investments in appropriate routing procedures offer additional potential for these rare but important cases. Such considerations are integral to responsible classifiers and assist in developing goal-oriented AI.

6.2. Continuous Learning and Model Governance

The evolving nature of enterprise environments makes ticket-category distributions highly dynamic, requiring careful monitoring of classification performance and retraining of operational classifiers whenever they degrade. Even with such a retraining cadence, the growing volume of tickets introduces a longer-term challenge: the gradual onset of class imbalance. When training data are extremely expensive to produce, as often is the case in enterprises, rare categories are frequently underrepresented, degrading classifier performance on such tickets. One potential mitigation involves cost-sensitive objectives and evaluation metrics. Another is anomaly-aware techniques that exploit abundant negative examples. Radial-distribution-based methods, which compare a prototype to all training samples, naturally generalize to low-density areas and have successfully handled rare events in fields such as video-surveillance anomaly detection.

For text-based tickets, a lack of sufficient labeled data is often addressed by leveraging unlabelled data. Semi-supervised methods aim to allocate labels to unannotated samples, taking advantage of co-training scenarios. Self-supervised techniques aim to generate ‘pretrained’ models capable of learning general representations from easily-available data, followed by transfer-learning to address specific tasks. In extreme-learning circumstances, general-domain models like CLIP enable zero-shot classification, although with limited accuracy. Yet enterprise environments also offer a natural source of multimodal data: textual tickets are commonly accompanied by supporting system logs, chat transcripts, and attachments. In this context, the task can therefore be classified as zero-shot or few-shot learning, as supervised models can quickly exploit label knowledge through signal-driven approaches like label propagation, contrastive-loss methods learn to classify even with annotation at the instance level.

7. Conclusion

AI support for enterprise IT Service Management (ITSM) is a significant topic in enterprise service delivery. This support may start with intelligent ticket classification linking incoming tickets with suitable fulfillment procedures, as proposed here. The classification of incoming tickets into a sufficient number of typical categories can reduce the direct burden on the service desk and substantially improve ticket routing into second-/third-line resolution teams or automatic fulfillment procedures associated with a Suitable fulfilment Procedure. The impact can be assessed qualitatively against ticket-handling KPIs without need for large-scale controlled experiments.

The assessment shows that the development of suitable AI-based classification approaches is still an open research topic. First, a collection of common challenges faced when building AI-driven classifiers—specifically using supervised learning techniques

and primarily for natural language text input—has been presented. These challenges cannot simply be mitigated using direct human effort—through additional labelling, defining rules, etc.—since they arise precisely as a consequence of the length and diversity of the ticket data available in ESMS. Such approaches would require resources on the scale of large technology companies and most probably still end up producing sub-optimal classifiers. By framing the approach as that of a Jeopardy-style classification task with class-specific choices enabled by low-resource domains, the relevant ticket classifications can be effectively handled with the support of AI.

Measure	Formula	Value	Operational meaning
Precision	TP/(TP+FP)	0.800	How many routed tickets were correct
Recall	TP/(TP+FN)	0.706	How many true tickets were recovered
F1	2PR/(P+R)	0.750	Balance between precision and recall
Accuracy	(TP+TN)/Total	0.976	Can look strong even when rare classes are weak

Table : Compact worked example for metrics

7.1. Future Trends

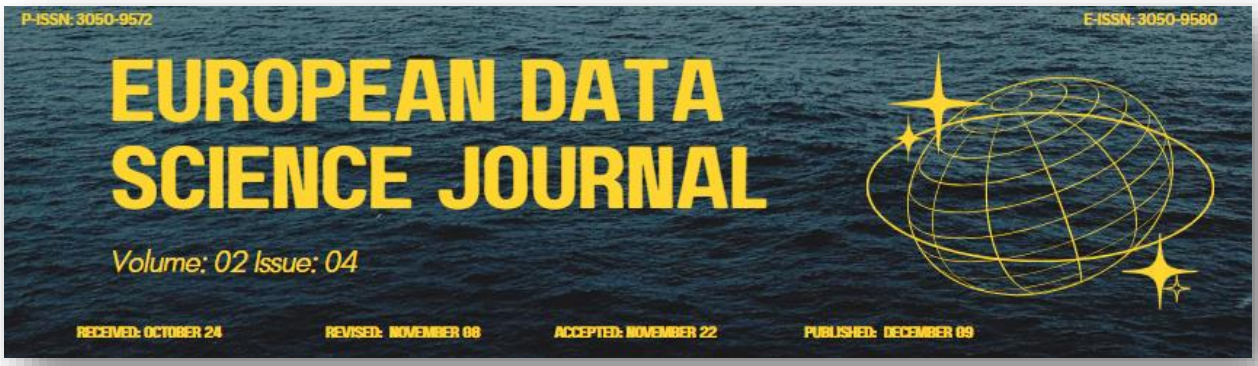
Emerging trends for enterprise IT service management composition systems include modeling multimodal ticket data, integrating information from field values, textual descriptions, and images; addressing zero- or few-shot learning with novel tickets from unobserved categories; supporting incident and service request automation; and aiding service desk agents with instant suitable-knowledge-base suggestions. Other developments focus on self-supervised or contrastive learning to alleviate the dependency on labeled samples, noise-based label propagation to transfer label information from high- to low-density regions, and consistent feature learning to support domain adaptation.

8. References

1. Al-Hawari, F., & Barham, H. (2021). A machine learning based help desk system for IT service management. Journal of King Saud University - Computer and Information Sciences, 33(6), 702–718.
2. Ramya, C., Paramesh, S. P., & Shreedhara, K. S. (2021). Classifying the unstructured IT service desk tickets using ensemble of classifiers. arXiv.
3. Tezgider, M. (2021). Text classification using improved bidirectional transformer. Concurrency and Computation: Practice and Experience, 34(1), e6486.

4. Inala, R. (2023). Revolutionizing Customer Master Data in Insurance Technology Platforms: An AI and MDM Architecture Perspective. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 579-606.
5. Riekert, M., Riekert, M., & Klein, A. (2021). Simple baseline machine learning text classifiers for small datasets. SN Computer Science, 2, 178.
6. Gao, S., Alawad, M., Young, M. T., Gounley, J., Schaefferkoetter, N., Yoon, H.-J., & others. (2021). Limitations of transformers on clinical text classification. IEEE Journal of Biomedical and Health Informatics, 25(9), 3596–3607.
7. Guo, H., Yuan, S., & Wu, X. (2021). LogBERT: Log anomaly detection via BERT. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), 1–8.
8. Mangalampalli, B. M. Generative AI Applications In Healthcare Data Mart Design And Optimization.
9. Schad, J., Sambasivan, R., & Woodward, C. (2022). Predicting help desk ticket reassignments with graph convolutional networks. Machine Learning with Applications, 7, 100237.
10. Seanbudh, S., & Siriborvornratanakul, T. (2022). Using machine learning to improve ticket classification for IT service management system. Science and Engineering Connect.
11. Paramesh, S. P., & Shreedhara, K. S. (2022). A deep learning based IT service desk ticket classifier using CNN. ICTACT Journal on Soft Computing, 13(1), 2805–2812.
12. Aitha, A. R. (2021). Dev Ops Driven Digital Transformation: Accelerating Innovation In The Insurance Industry. Journal of International Crisis and Risk Communication Research.
13. Oliveira, D. F., Nogueira, A. S., & Brito, M. A. (2022). Performance comparison of machine learning algorithms in classifying information technologies incident tickets. AI, 3(3), 601–622.
14. Gómez-Jaramillo, P. A., González-Echavarría, F., & Pérez-Rave, J. I. (2022). Technological incident classification model from a machine learning approach in insurance services. DYNA, 89(221).
15. Inala, R. (2023). Big Data Architectures for Modernizing Customer Master Systems in Group Insurance and Retirement Planning. Educational Administration: Theory and Practice, 29(4), 5493-5505.
16. Feng, L., Senapati, J., & Liu, B. (2022). TaDaa: Real time ticket assignment deep learning auto advisor for customer support, help desk, and issue ticketing systems. arXiv.

17. Zicari, P., Folino, G., Guarascio, M., & Pontieri, L. (2022). Combining deep ensemble learning and explanation for intelligent ticket management. Expert Systems with Applications, 206, 117815.
18. Rodrawangpai, B., & Daungjaiboon, W. (2022). Improving text classification with transformers and layer normalization. Machine Learning with Applications, 10, 100403.
19. Occhipinti, A., Rogers, L., & Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. Expert Systems with Applications, 201, 117193.
20. Yandamuri, U. S. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. International Journal Of Finance, 36(6), 682-706.
21. Dogra, V., Verma, S., Kavita, Chatterjee, P., Shafi, J., Choi, J., & Ijaz, M. F. (2022). A complete process of text classification system using state-of-the-art NLP models. Computational Intelligence and Neuroscience, 2022, Article 1883698.
22. Li, R., Liu, M., Xu, D., Gao, J., Wu, F., & Zhu, L. (2022). A review of machine learning algorithms for text classification. In W. Lu, Y. Zhang, W. Wen, H. Yan, & C. Li (Eds.), Cyber security (pp. 226–234). Springer.
23. Aravind, R., Surabhi, S. N. D., & Shah, C. V. (2023). Remote Vehicle Access: Leveraging Cloud Infrastructure for Secure and Efficient OTA Updates with Advanced AI. EuropeanEconomic Letters (EEL), 13 (4), 1308–1319. Retrieved fromhttps.
24. Li, Y., Zhang, X., He, S., Chen, Z., Kang, Y., Liu, J., Li, L., Dang, Y., Gao, F., Xu, Z., Rajmohan, S., Lin, Q., Zhang, D., & Lyu, M. R. (2022). An intelligent framework for timely, accurate, and comprehensive cloud incident detection. ACM SIGOPS Operating Systems Review, 56(1), 1–7.
25. Thota, R. C. (2022). AIOps in cloud computing: Automation performance monitoring with AI and New Relic. International Journal of Scientific Research and Management, 10(3), 809–817.
26. Arias-Barahona, M. X., Arteaga-Arteaga, H. B., Orozco-Arias, S., Flórez-Ruíz, J. C., Valencia-Díaz, M. A., & Tabares-Soto, R. (2023). Requests classification in the customer service area for software companies using machine learning and natural language processing. PeerJ Computer Science, 9, e1016.



27. Ackerman, S., Alexander, L., Bennett, M., Chen, D., Farchi, E., Houseknecht, A., & Santhanam, P. (2023). Deploying automated ticket router across the enterprise. AI Magazine, 44(1), 97–111.
28. Aravind, R., Surabhi, S. N. D., & Shah, C. V. (2023). Remote Vehicle Access: Leveraging Cloud Infrastructure for Secure and Efficient OTA Updates with Advanced AI. Eur. Econ. Lett, 13, 1309-1319.
29. Ahmed, S., Singh, M., Doherty, B., Ramlan, E., Harkin, K., Bucholc, M., & Coyle, D. (2023). Knowledge-based intelligent system for IT incident DevOps. In Proceedings of the 2023 IEEE/ACM International Workshop on Cloud Intelligence and AIOps, 1–7.
30. Ahmed, S., Singh, M., Doherty, B., Ramlan, E., Harkin, K., Bucholc, M., & Coyle, D. (2023). An empirical analysis of state-of-art classification models in an IT incident severity prediction framework. Applied Sciences, 13(6), 3843.
31. Bruni, R., Bianchi, G., & Papa, P. (2023). Hyperparameter black-box optimization to improve the automatic classification of support tickets. Algorithms, 16(1), 46.
32. Zangari, A., Marcuzzo, M., Schiavinato, M., Gasparetto, A., & Albarelli, A. (2023). Ticket automation: An insight into current research with applications to multi-level classification scenarios. Expert Systems with Applications, 225, 119984.