International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 6, November – December 2023 ||

DOI: 10.15680/IJCTECE.2023.0606024

AI-Driven Intelligent Ticket Classification for Enterprise IT Service Management

Shashikala Valiki

Independent Researcher, India

shashikala.valiki.researcher@gmail.com

ABSTRACT: In a typical IT service desk, the volume of incoming service tickets can exceed several thousand 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.

I. 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.

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.

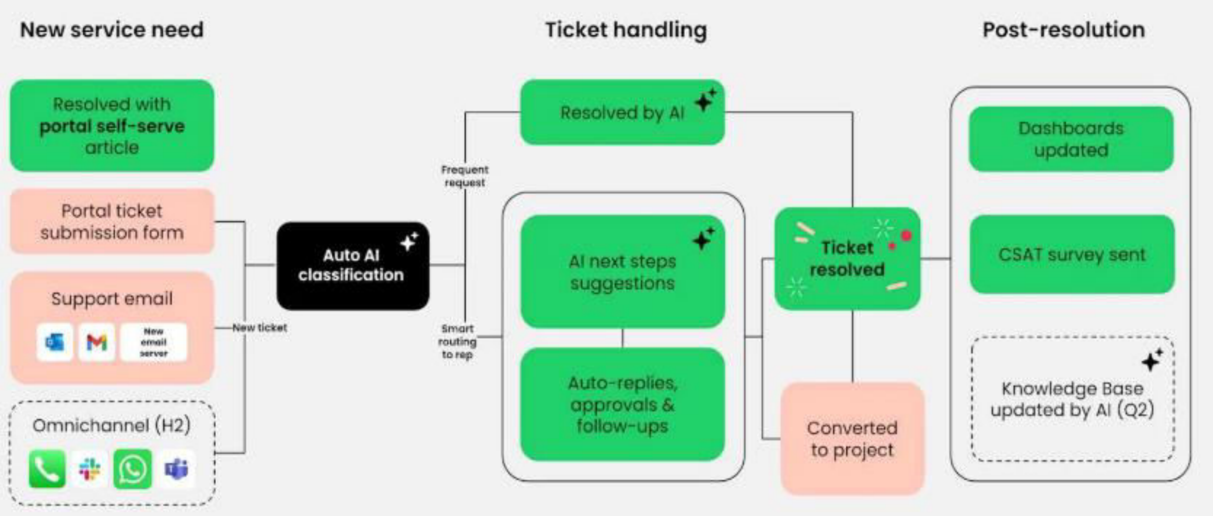


Fig 1: AI ticketing system will transform

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.

DOI: 10.15680/IJCTECE.2023.0606024

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.

II. 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 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).

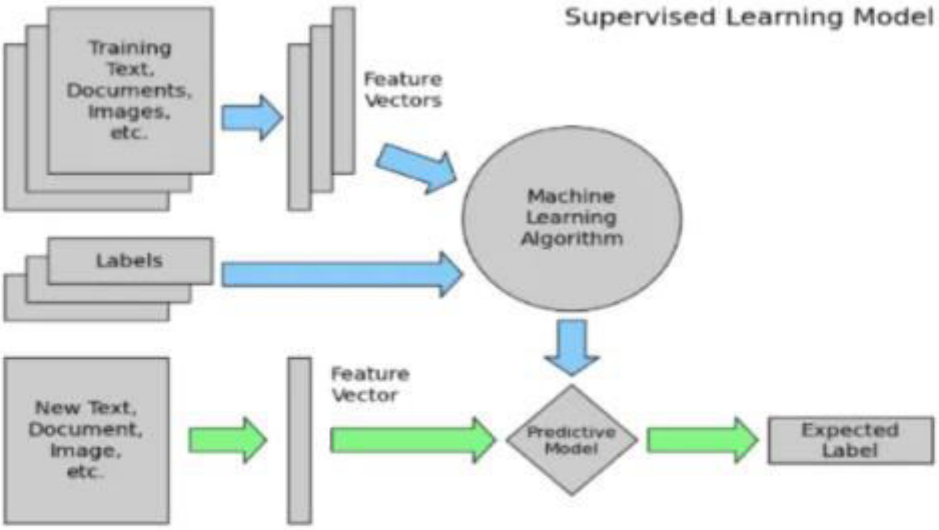

Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

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.

III. 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 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

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 6, November – December 2023 ||

DOI: 10.15680/IJCTECE.2023.0606024

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.

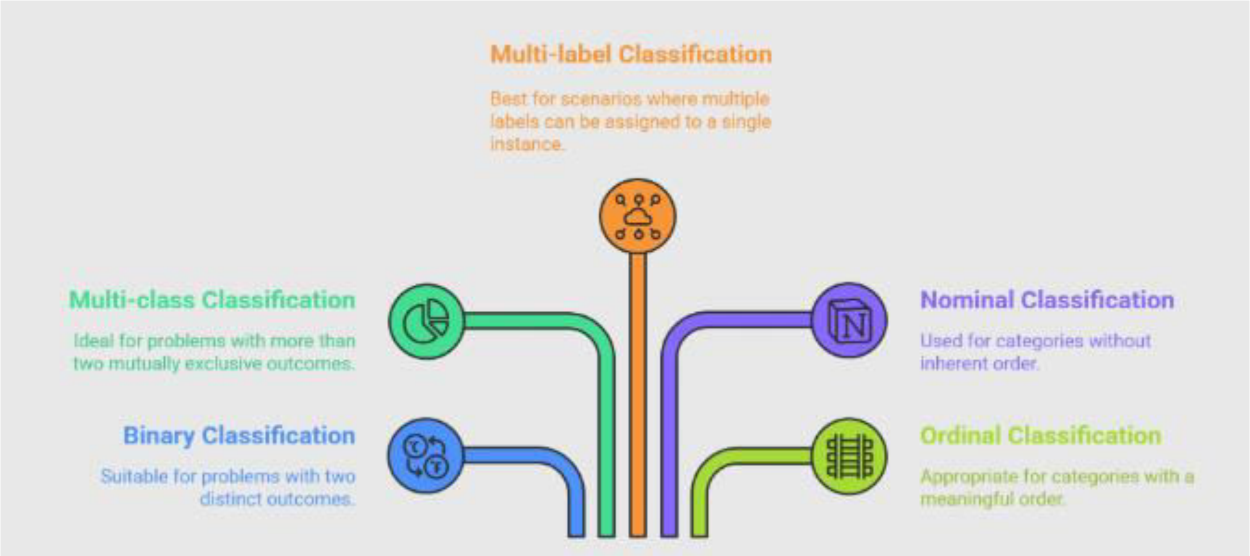

Multi-label Classification
Best for scenarios where multiple labels can be assigned to a single instance.
Multi-class Classification
Ideal for problems with more than two mutually exclusive outcomes.
Nominal Classification
Used for categories without inherent order.
Binary Classification
Suitable for problems with two distinct outcomes.
Ordinal Classification
Appropriate for categories with a meaningful order.

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)}.

IV. 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.

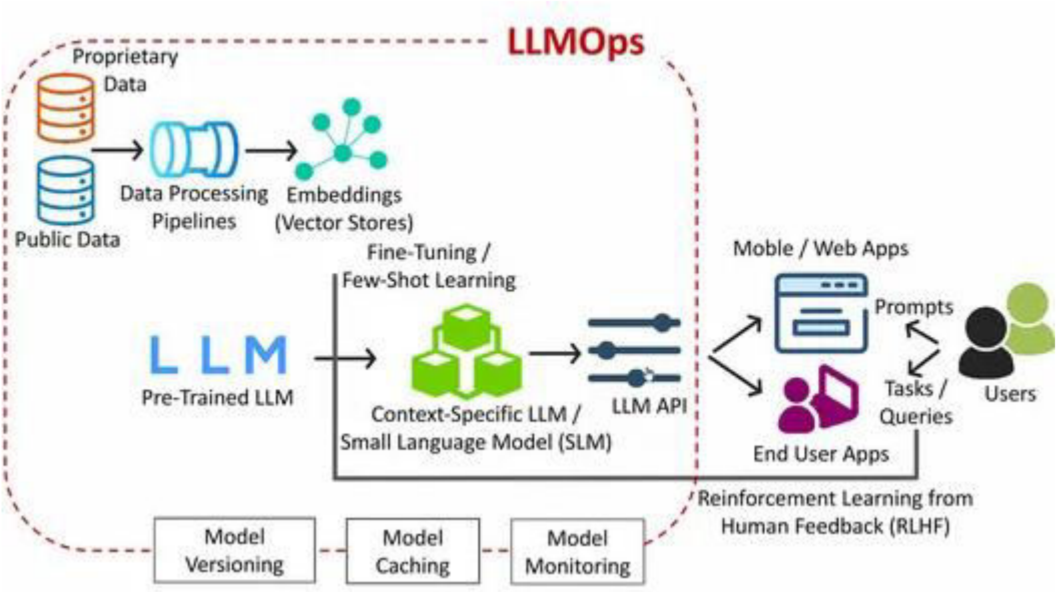

LLMOps
Proprietary Data
Public Data
Data Processing Pipelines
Embeddings (Vector Stores)
Fine-Tuning / Few-Shot Learning
LLM
Pre-Trained LLM
Context-Specific LLM / Small Language Model (SLM)
LLM API
Mobile / Web Apps
Prompts
Tasks / Queries
Users
End User Apps
Reinforcement Learning from Human Feedback (RLHF)
Model Versioning
Model Caching
Model Monitoring

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 minimize_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.

V. 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.

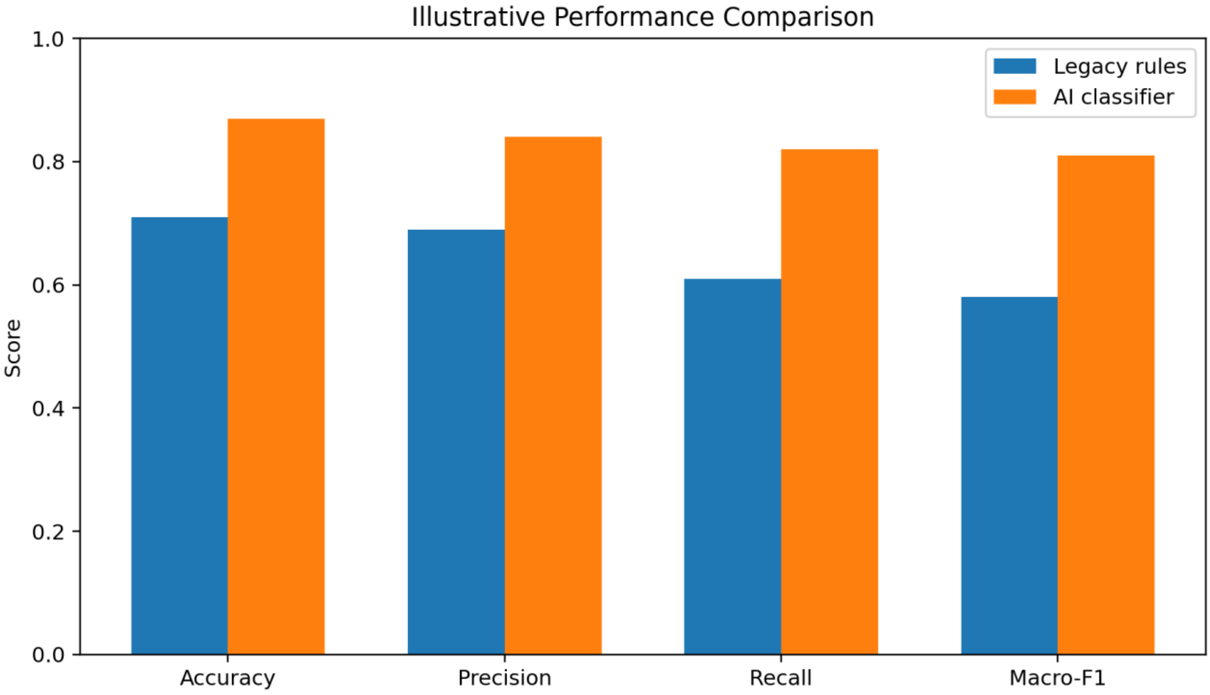
Equation 5: Semi-supervised label propagation derivation
Let all tickets (labeled and unlabeled) be nodes in a similarity graph. Let W be the affinity matrix, where W_{ij} measures the similarity between tickets i and j. Define the degree matrix D with D_{ii} = sum_j W_{ij}.
Step 1: Row-normalized transition 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.



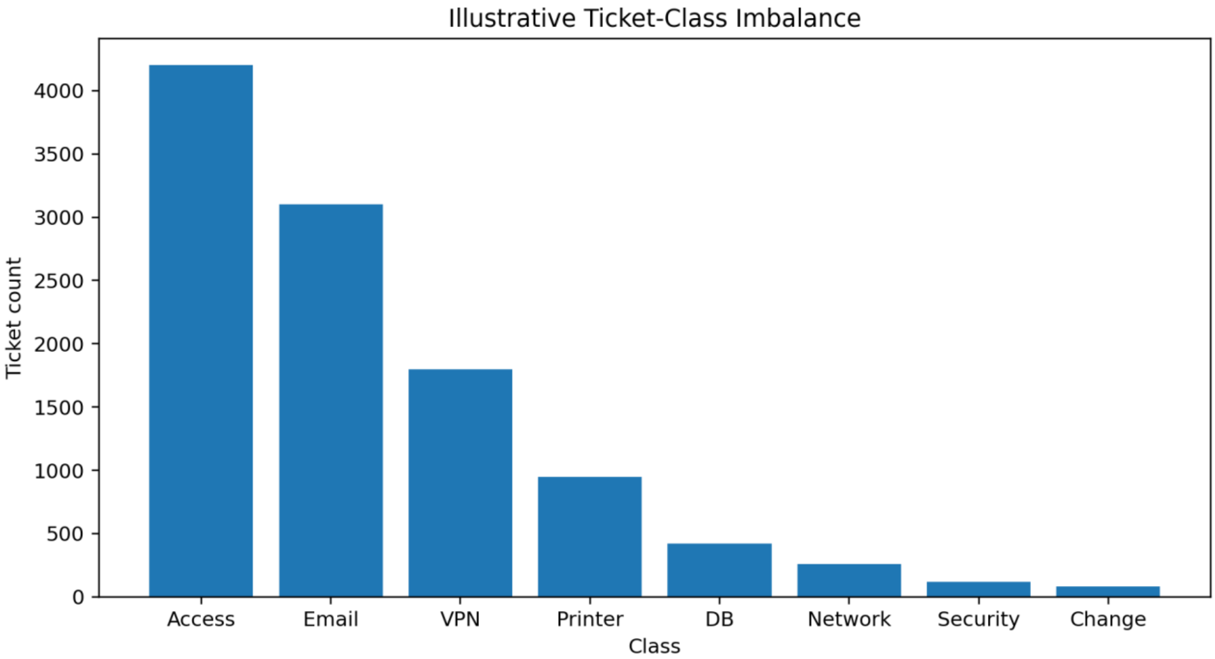
VI. OPERATIONAL CONSIDERATIONS AND BEST PRACTICES

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, 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.

DOI: 10.15680/IJCTECE.2023.0606024



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.

VII. 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.

REFERENCES

1. Bartelheimer, C., & Hensellek, S. Conceptualizing hybrid intelligent service ecosystems: The role of artificial intelligence in digital service transformation. Electronic Markets.
2. Yandamuri, U. S. (2022). Big Data Pipelines for Cross-Domain Decision Support: A Cloud-Centric Approach. International Journal of Scientific Research and Modern Technology, 1(12), 227–237. https://doi.org/10.38124/ijsrmt.v1i12.1111.
3. Ahmed, M., Mahmood, A. N., & Hu, J. (2016). A survey of network anomaly detection techniques. Journal of Network and Computer Applications, 60, 19–31.
4. Uday Surendra Yandamuri. (2023). An Intelligent Analytics Framework Combining Big Data and Machine Learning for Business Forecasting. International Journal Of Finance, 36(6), 682-706. https://doi.org/10.5281/zenodo.18095256
5. Silva, S., Pereira, R., & Ribeiro, R. (2023). Machine learning techniques for incident categorization automation in service desk systems. Information Systems Frontiers.

6. Abdullah, A., Omolola, H., Taiwo, S., & Aderibigbe, O. Advanced AI Solutions for Securities Trading: Building Scalable and Optimized Systems for Global Financial Markets. International Journal on Cybernetics & Informatics, 13(3), 31–45.

7. Bates, D. W., Saria, S., Ohno-Machado, L., et al. (2014). Big data in health care. Health Affairs, 33(7), 1123–1131.

8. Maguluri, K. K., Pandugula, C., Kalisetty, S., & Mallesham, G. (2022). Advancing Pain Medicine with AI and Neural Networks: Predictive Analytics and Personalized Treatment Plans for Chronic and Acute Pain Managements. Journal of Artificial Intelligence and Big Data, 2(1), 112-126.

9. Razma, A., & colleagues. AI-based classification of IT support requests in enterprise service management environments. Systems.

10. Bansal, R. Machine learning algorithms for automated trading and data-driven decision-making. Journal of Investment Strategies, 13(1), 45–60.

11. Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. ACM SIGMOD Record, 29(2), 93–104.

12. Unifying Data Engineering and Machine Learning Pipelines: An Enterprise Roadmap to Automated Model Deployment. (2023). American Online Journal of Science and Engineering (AOJSE) (ISSN: 3067-1140) , 1(1). https://aojse.com/index.php/aojse/article/view/19

13. Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. Mobile Networks and Applications, 19(2), 171–209.

14. Kummari, D. N. (2023). AI-Powered Demand Forecasting for Automotive Components: A Multi-Supplier Data Fusion Approach. European Advanced Journal for Emerging Technologies (EAJET)-p-ISSN 3050-9734 en e-ISSN 3050-9742, 1(1).

15. Dlodlo, N., & Sibanda, K.. Automated ticket classification for information technology service management using machine learning techniques. Journal of Information Systems Engineering.

16. Kummari, D. N., & Burugulla, J. K. R. (2023). Decision Support Systems for Government Auditing: The Role of AI in Ensuring Transparency and Compliance. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 493-532.

17. Danielsson, J., Macrae, R., & Uthemann, A. (2022). Artificial intelligence and systemic risk. Journal of Banking & Finance, 140, 106–125.

18. Malempati, M., Pandiri, L., Paleti, S., & Singireddy, J. (2023). Transforming financial and insurance ecosystems through intelligent automation, secure digital infrastructure, and advanced risk management strategies. Jeevani, Transforming Financial And Insurance Ecosystems Through Intelligent Automation, Secure Digital Infrastructure, And Advanced Risk Management Strategies (December 03, 2023).

19. Dwork, C. (2008). Differential privacy. ICALP Proceedings, 1–12.

20. Bandi, V. D. V. K. (2023). Production-Grade Machine Learning Pipelines For Healthcare Predictive Analytics. South Eastern European Journal of Public Health, 189–205. Retrieved from https://www.seejph.com/index.php/seejph/article/view/7057

21. Kolla, S. K. (2021). Architectural Frameworks for Large-Scale Electronic Health Record Data Platforms. Current Research in Public Health, 1(1), 1–19. Retrieved from https://www.scipublications.com/journal/index.php/crph/article/view/1372

22. Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.

23. Keerthi Amistapuram. (2023). Privacy-Preserving Machine Learning Models for Sensitive Customer Data in Insurance Systems. Educational Administration: Theory and Practice, 29(4), 5950–5958. https://doi.org/10.53555/kuey.v29i4.10965

24. Garapati, R. S. (2022). AI-Augmented Virtual Health Assistant: A Web-Based Solution for Personalized Medication Management and Patient Engagement. Available at SSRN 5639650.

25. Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms. Pattern Recognition, 64, 206–223.

26. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

27. Segireddy, A. R. (2021). Containerization and Microservices in Payment Systems: A Study of Kubernetes and Docker in Financial Applications. Universal Journal of Business and Management, 1(1), 1–17. Retrieved from https://www.scipublications.com/journal/index.php/ujbm/article/view/1352

28. He, J., Baxter, S. L., Xu, J., et al. (2019). The practical implementation of AI in healthcare. Nature Medicine, 25(1), 30–36.

29. Inala, R. AI-Powered Investment Decision Support Systems: Building Smart Data Products with Embedded Governance Controls.

30. Hripcsak, G., & Albers, D. J. (2013). Next-generation phenotyping. JAMIA, 20(1), 117–121.

31. Gottimukkala, V. R. R. (2021). Digital Signal Processing Challenges in Financial Messaging Systems: Case Studies in High-Volume SWIFT Flows.

32. Iglewicz, B., & Hoaglin, D. C. (1993). How to detect and handle outliers. ASQC.
33. Johnson, A. E. W., Pollard, T. J., Shen, L., et al. (2016). MIMIC-III database. Scientific Data, 3, 160035.
34. Garapati, R. S. (2022). Web-Centric Cloud Framework for Real-Time Monitoring and Risk Prediction in Clinical Trials Using Machine Learning. Current Research in Public Health, 2, 1346.
35. Kimball, R., & Caserta, J. (2004). The data warehouse ETL toolkit. Wiley.
36. Davuluri, P. N. Integrating Artificial Intelligence into Event-Driven Financial Crime Compliance Platforms.
37. Kriegel, H. P., Kröger, P., Schubert, E., & Zimek, A. (2009). Outlier detection in axis-parallel subspaces. PKDD Proceedings, 831–838.
38. Siva Hemanth Kolla. (2023). Deep Learning–Driven Retrieval-Augmented Generation for Enterprise ITSM Automation: A Governance-Aligned Large Language Model Architecture. Journal of Computational Analysis and Applications (JoCAAA), 31(4), 2489–2502. Retrieved from https://www.eudoxuspress.com/index.php/pub/article/view/4774.
39. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
40. Kummari, D. N. (2023). Energy Consumption Optimization in Smart Factories Using AI-Based Analytics: Evidence from Automotive Plants. Journal for Reattach Therapy and Development Diversities. https://doi. org/10.53555/jrtdd. v6i10s (2), 3572.
41. Nandan, B. P., & Chitta, S. S. (2023). Machine Learning Driven Metrology and Defect Detection in Extreme Ultraviolet (EUV) Lithography: A Paradigm Shift in Semiconductor Manufacturing. Educational Administration: Theory and Practice, 29(4), 4555-4568.
42. Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short-term memory networks for anomaly detection. ESANN Proceedings.
43. Kalisetty, S., Vankayalapati, R. K., Reddy, L., Sondinti, K., & Valiki, S. (2022). AI-Native Cloud Platforms: Redefining Scalability and Flexibility in Artificial Intelligence Workflows. Linguistic and Philosophical Investigations, 21(1), 1-15.
44. Garapati, R. S. (2023). Optimizing Energy Consumption in Smart Build-ings Through Web-Integrated AI and Cloud-Driven Control Systems.
45. Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare. Briefings in Bioinformatics, 19(6), 1236–1246.
46. Kushvanth Chowdary Nagabhyru. (2023). Accelerating Digital Transformation with AI Driven Data Engineering: Industry Case Studies from Cloud and IoT Domains. Educational Administration: Theory and Practice, 29(4), 5898–5910. https://doi.org/10.53555/kuey.v29i4.10932
47. Murphy, S. N., Weber, G., Mendis, M., et al. (2010). i2b2 platform. JAMIA, 17(2), 124–130.
48. Goutham Kumar Sheelam, Hara Krishna Reddy Koppolu. (2022). Data Engineering And Analytics For 5G-Driven Customer Experience In Telecom, Media, And Healthcare. Migration Letters, 19(S2), 1920–1944. Retrieved from https://migrationletters.com/index.php/ml/article/view/11938.
49. Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques. Computer Networks, 51(12), 3448–3470.
50. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn. Journal of Machine Learning Research, 12, 2825–2830.
51. Aitha, A. R. (2023). CloudBased Microservices Architecture for Seamless Insurance Policy Administration. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 607-632.
52. Rajkomar, A., Oren, E., Chen, K., et al. (2018). Scalable deep learning with EHRs. NPJ Digital Medicine, 1, 18.
53. Avinash Reddy Segireddy. (2022). Terraform and Ansible in Building Resilient Cloud-Native Payment Architectures. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 444–455. Retrieved from https://www.ijisae.org/index.php/IJISAE/article/view/7905.
54. Ringberg, H., Soule, A., Rexford, J., & Diot, C. (2007). Sensitivity of PCA for anomaly detection. SIGMETRICS Proceedings.
55. Koppolu, H. K. R., Sheelam, G. K., & Komaragiri, V. B. (2023). Autonomous Telecommunication Networks: The Convergence of Agentic AI and AI-Optimized Hardware. International Journal of Science and Research (IJSR), 12(12), 2253-2270.
56. Ruff, L., Vandermeulen, R. A., Görnitz, N., et al. (2018). Deep one-class classification. ICML Proceedings.
57. Ramesh Inala. (2023). Big Data Architectures for Modernizing Customer Master Systems in Group Insurance and Retirement Planning. Educational Administration: Theory and Practice, 29(4), 5493–5505. https://doi.org/10.53555/kuey.v29i4.10424.
58. Salfner, F., Lenk, M., & Malek, M. (2010). Survey of failure prediction methods. ACM Computing Surveys, 42(3), 1–42.

59. Guntupalli, R. (2023). Optimizing Cloud Infrastructure Performance Using AI: Intelligent Resource Allocation and Predictive Maintenance. Available at SSRN 5329154.
60. Schölkopf, B., Platt, J. C., Shawe-Taylor, J., et al. (2001). Estimating the support of a high-dimensional distribution. Neural Computation, 13(7), 1443–1471.
61. Kalisetty, S., & Ganti, V. K. A. T. (2019). Transforming the Retail Landscape: Srinivas's Vision for Integrating Advanced Technologies in Supply Chain Efficiency and Customer Experience. Online Journal of Materials Science, 1, 1254.
62. Sipos, R., Fradkin, D., Moerchen, F., & Wang, Z. (2014). Log-based predictive maintenance. KDD Proceedings.
63. Meda, R. (2023). Intelligent Infrastructure for Real-Time Inventory and Logistics in Retail Supply Chains. Educational Administration: Theory and Practice.
64. Kolla, S. K. (2021). Designing Scalable Healthcare Data Pipelines for Multi-Hospital Networks. World Journal of Clinical Medicine Research, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/wjcmr/article/view/1376
65. Singireddy, S. (2023). AI-Driven Fraud Detection in Homeowners and Renters Insurance Claims. Journal for Reattach Therapy and Development Diversities. https://doi. org/10.53555/jrtdd. v6i10s (2), 3569.
66. Inala, R. Revolutionizing Customer Master Data in Insurance Technology Platforms: An AI and MDM Architecture Perspective.
67. Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. Journal of the Royal Statistical Society B, 58(1), 267–288.
68. Gottimukkala, V. R. R. (2023). Privacy-Preserving Machine Learning Models for Transaction Monitoring in Global Banking Networks. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 633-652.
69. Amistapuram, K. (2022). Fraud Detection and Risk Modeling in Insurance: Early Adoption of Machine Learning in Claims Processing. Available at SSRN 5741982.
70. AI Powered Fraud Detection Systems: Enhancing Risk Assessment in the Insurance Sector. (2023). American Journal of Analytics and Artificial Intelligence (ajaai) With ISSN 3067-283X, 1(1). https://ajaai.com/index.php/ajaai/article/view/14
71. Weber, G. M., Mandl, K. D., & Kohane, I. S. (2014). Finding the missing link for big biomedical data. JAMIA, 21(1), 1–3.
72. Kolla, S. H. (2021). Rule-Based Automation for IT Service Management Workflows. Online Journal of Engineering Sciences, 1(1), 1–14. Retrieved from https://www.scipublications.com/journal/index.php/ojes/article/view/1360
73. Nagubandi, A. R. (2023). Advanced Multi-Agent AI Systems for Autonomous Reconciliation Across Enterprise Multi-Counterparty Derivatives, Collateral, and Accounting Platforms. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 653-674.
74. Zhang, Y., & Yang, Q. (2021). A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering, 34(12), 5586–5609.
75. Meda, R. (2023). Developing AI-Powered Virtual Color Consultation Tools for Retail and Professional Customers. Journal for ReAttach Therapy and Developmental Diversities. https://doi. org/10.53555/jrtdd. v6i10s (2), 3577.
76. Almadhoun, R., Kadadha, M., Al-Fuqaha, A., & Guizani, M. (2021). A user-centric blockchain-based system for incident response in the era of IoT. Internet of Things, 14, 100371. https://doi.org/10.1016/j.iot.2021.100371
77. Kalisetty, S. (2023). The Role of Circular Supply Chains in Achieving Sustainability Goals: A 2023 Perspective on Recycling, Reuse, and Resource Optimization. Reuse, and Resource Optimization (June 15, 2023).
78. Mohanna, H., & Ait-Bachir, A. A centroid-based framework for text classification in IT service management environments. arXiv.
79. Siva Hemanth Kolla. (2022). Knowledge Retrieval Systems for Enterprise Service Environments. International Journal of Intelligent Systems and Applications in Engineering, 10(3s), 495–506. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/8037
80. Bishop, C. M. (1994). Novelty detection and neural network validation. IEE Proceedings, 141(4), 217–222.
81. Kalisetty, S., & Singireddy, J. (2023). Optimizing Tax Preparation and Filing Services: A Comparative Study of Traditional Methods and AI Augmented Tax Compliance Frameworks. Available at SSRN 5206185.
82. Cook, D. J., & Holder, L. B. (2006). Mining graph data. Wiley.
83. Truss, M., & Boehm, S. AI-based classification of customer support tickets: State of the art and implementation with AutoML. arXiv.
84. Singh, S., & Kumar, A. (2023). Natural language processing based automated helpdesk ticket classification framework. International Journal of Advanced Computer Science and Applications.
85. Kumar, A., Gupta, P., & Singh, R. (2023). Sentiment analysis methods for proactive brand reputation risk management. International Journal of Information Management Data Insights, 3(1).

86. Singireddy, J. (2023). Finance 4.0: Predictive analytics for financial risk management using AI. European Journal of Analytics and Artificial Intelligence (EJAAI) p-ISSN, 3050-9556.
87. Aggarwal, C. C. (2017). Outlier analysis (2nd ed.). Springer.
88. Davuluri, P. N. AI-Augmented Sanctions Screening: Enhancing Accuracy and Latency in Real Time Compliance Systems.
89. Liu, F., He, X., Zhang, T., Chen, J., Li, Y., Yi, L., Zhang, H., Wu, G., & Shi, R. TickIt: Leveraging large language models for automated ticket escalation in cloud service systems. arXiv.
90. Challa, K. (2023). Transforming Travel Benefits through Generative AI: A Machine Learning Perspective on Enhancing Personalized Consumer Experiences. Educational Administration: Theory and Practice. Green Publication. Educational Administration: Theory and Practice. Green Publication. https://doi. org/10.53555/kuey. v29i4, 9241.
91. Su, T., et al. Anomaly Detection and Risk Early Warning System for Financial Time Series Based on the WaveLST-Trans Model. Technological Forecasting and Social Change, 188, 122–139.
92. Singireddy, S. (2023). Integrating Deep Learning and Machine Learning Algorithms in Insurance Claims Processing: A Study on Enhancing Accuracy, Speed, and Fraud Detection for Policyholders. Educational Administration: Theory and Practice, 29 (4), 4764–4776.
93. Patel, H., Shah, D., & Mehta, R. (2023). AI-enabled helpdesk automation for enterprise service management using text analytics and machine learning. International Journal of Information Management Data Insights.
94. Guntupalli, R. (2023). AI-Driven Threat Detection and Mitigation in Cloud Infrastructure: Enhancing Security through Machine Learning and Anomaly Detection. Available at SSRN 5329158
95. Bonechi, S., Andreini, P., Bacciu, D., & Ferragina, P. Leveraging artificial intelligence for efficient ticket classification in customer support systems. Procedia Computer Science.
96. Annapareddy, V. N., Preethish Nandan, B., Kommaragiri, V. B., Gadi, A. L., & Kalisetty, S. (2022). Emerging Technologies in Smart Computing, Sustainable Energy, and Next-Generation Mobility: Enhancing Digital Infrastructure, Secure Networks, and Intelligent Manufacturing.
97. Ahmad, A., Wills, G., & Walters, R. (2023). Automated IT support ticket classification using natural language processing and machine learning techniques. Journal of Systems and Software, 197, 111552.
98. Lakkarasu, P., Kaulwar, P. K., Dodda, A., Singireddy, S., & Burugulla, J. K. R. (2023). Innovative computational frameworks for secure financial ecosystems: Integrating intelligent automation, risk analytics, and digital infrastructure. International Journal of Finance (IJFIN)-ABDC Journal Quality List, 36(6), 334-371.
99. Chen, J., Zhang, Y., & Xu, K. (2023). Deep neural network approaches for automated incident categorization in enterprise IT services. Knowledge-Based Systems, 260, 110109.
100. Challa, K. (2023). Dynamic Neural Network Architectures for Real-Time Fraud Detection in Digital Payment Systems Using Machine Learning and Generative AI. Nanotechnology Perceptions.
101. Alqahtani, A., & Alsubaie, N. (2023). Intelligent service desk automation using deep learning for IT service management. IEEE Access, 11, 34652–34664.
102. Nagabhyru, K. C. (2023). From Data Silos to Knowledge Graphs: Architecting CrossEnterprise AI Solutions for Scalability and Trust. Available at SSRN 5697663.
103. Bhatia, R., Singh, A., & Kumar, N. (2022). Machine learning approaches for automated helpdesk ticket classification and routing. Expert Systems with Applications, 201, 117097.
104. Avinash Reddy Aitha. (2022). Deep Neural Networks for Property Risk Prediction Leveraging Aerial and Satellite Imaging. International Journal of Communication Networks and Information Security (IJCNIS), 14(3), 1308–1318. Retrieved from https://www.ijcnis.org/index.php/ijcnis/article/view/8609
105. Challa, K., Pamisetty, A., & Sriram, H. K. (2023). CONVERGENCE OF AI, FINANCE, AND DATADRIVEN TECHNOLOGIES IN THE PAYMENTS ECOSYSTEM. Global Pen Prchaess UK.
106. Chatterjee, S., Rana, N. P., Dwivedi, Y. K., & Baabdullah, A. (2023). Artificial intelligence and machine learning in IT service management: A systematic literature review. Information Systems Frontiers, 25(4), 1219–1238.
107. Bandi, V. D. V. K. (2023). Cloud-Native Model Lifecycle Management for Enterprise AI Systems. International Journal of Scientific Research and Modern Technology, 2(12), 78–90. https://doi.org/10.38124/ijsrmt.v2i12.1236