



Inverse-problem perspective on indirect information leakage in AI systems

Pavel Galmanov*

Moscow Institute of Physics and Technology (National Research University), Moscow, Russia

Submission: December 02, 2025; **Published:** December 09, 2025

*Corresponding author: Pavel Galmanov, Moscow Institute of Physics and Technology (National Research University), Moscow, Russia

Abstract

Artificial intelligence (AI) systems routinely operate on Sensitive Big Data (SBD), often under strict access-control and privacy regulations. Even when direct access to protected attributes is blocked, attackers may still reconstruct hidden values from seemingly harmless outputs: aggregated statistics, model scores, or explanations. We argue that this phenomenon-indirect leakage-is naturally and fruitfully analyzed as an inverse problem. We formalize leakage functions as approximate inverses of the AI pipeline, characterize conditions under which accurate reconstructions are possible, and study how classical regularization, probabilistic modeling, and blind data-processing architectures can both enable and limit such leakage. This inverse-problem perspective unifies diverse attack strategies (model inversion, gradient inversion, symbolic regression, invertible neural networks, Bayesian reconstruction) and suggests principled defenses based on controlling the effective ill-posedness of the reconstruction task.

Introduction

Artificial intelligence (AI) systems routinely operate on Large-Scale Sensitive Big Data (SBD), where each record aggregates many attributes about individuals or organizations. Classical protection focuses on preventing direct leakage via access control, encryption, and privacy-preserving query mechanisms, ensuring that no unauthorized party can directly read protected fields. Yet attackers can often reconstruct protected values by combining many “safe” outputs from models and analytic pipelines, producing indirect information leakage even when no access rules are violated [1-3].

In multi-owner environments, blind data-processing architectures allow different data owners to contribute records without revealing raw data to the processing center or to one another. Instead, each owner applies local preprocessing or encryption, and only derived features or encrypted aggregates are supplied to the AI pipeline. Analysts see only reports and model outputs, never raw records. In this survey we adopt an inverse-problem perspective, in which the forward map

$$f: X \rightarrow Y$$

models the AI pipeline, while leakage functions

$$\varphi: Y \rightarrow X \text{ and } \omega: Y \rightarrow R$$

approximate inverse mappings that recover attributes from outputs. This framing enables systematic analysis of what can be reconstructed under various observational and architectural constraints, and of how regularization and blind-processing mechanisms can constrain it.

Formal Model of Indirect Leakage in AI Systems

We represent the SBD as an $N \times n$ table. Row ρ is an entity

$$x^{(\rho)} = (x_1^{(\rho)}, \dots, x_n^{(\rho)}) \in X \subseteq R^n,$$

and column j is attribute x_j , after encoding categorical and other structured values into R . Attributes are partitioned into visible (non-sensitive) and protected ones. Let X_{vis} be the subspace of visible attributes and X_{prot} the subspace of protected attributes. The SBD may be stored centrally or distributed across multiple owners, but we abstract away from storage details and work with the full SBD regarded as a collection of records $x^{(\rho)} \in X$.

An AI pipeline is a mapping

$$y = f(x) = (y_1, \dots, y_m) \in Y \subseteq R^m$$

where each y_k is a reported quantity (metric, score, stability measure, explanation component, or other output). In general, f

may involve multiple stages (preprocessing, feature extraction, model evaluation, postprocessing), but we treat it as a single operator from X to Y . We assume that direct access to protected attributes is blocked; analysts see only y .

A scalar leakage function for attribute x_i is any mapping

$$\omega: Y \rightarrow R, \omega(f(x)) \approx x_i$$

on a significant subset of X . A vector leakage function is a mapping

$$\varphi: Y \rightarrow X \text{ or } \varphi: Y \rightarrow R^S$$

that recovers either the full record or a subset of attributes. The approximation may be in mean-square, in probability, or in some application-specific loss. The existence of accurate φ or ω means that indirect leakage is possible: one can recover hidden attributes from outputs alone. The absence (or instability) of such functions under realistic observational constraints means indirect leakage is limited.

Inverse Problem Formulation and Ill-posedness

Observed outputs satisfy

$$y^{obs} = f(x^*) + \eta,$$

Where η models measurement noise, deliberate report noise, aggregation error, or other distortions between the true forward map and the attacker's observation mechanism. In a blind-processing architecture, f may already incorporate local transformations and noise mechanisms applied by data owners.

The problem is typically *ill-posed* in the sense of Hadamard: small perturbations in y^{obs} (due to noise or deliberate obfuscation) can correspond to large changes in consistent x , and solutions may not exist or be non-unique. Inverse-problem theory has long studied such settings, emphasizing stability and regularization: one seeks approximate solutions that depend continuously on data and incorporate prior information about x . For indirect leakage, the ill-posedness of inverting f -under constraints imposed by the system architecture-determines how accurately hidden attributes can be reconstructed from outputs alone. In particular, if the inverse map is highly unstable, small report noise or architectural constraints can dramatically reduce reconstructive power, while stable inverse mappings imply strong leakage even under noisy observations. This stability-versus-reconstruction capability is central to analyzing and limiting indirect leakage [4-5].

Classical Regularization Methods for Recovering Hidden Attributes

Indirect leakage can be written as an operator equation

$$f(x) = y, y^{obs} = f(x^*) + \eta,$$

Where $f: X \rightarrow Y$ is the AI pipeline, x^* is the true record, and y^{obs} is the (noisy) observed output accessible to the attacker.

Classical inverse-problem theory studies regularized approximate inverses $R\gamma: Y \rightarrow X$ that map observed data to stable approximations of x^* , where γ is a regularization parameter (or set of parameters) encoding the strength of prior information and the trade-off between data fit and stability. In this language, a leakage function is simply a component of a regularized inverse,

$$x_\gamma(y^{obs}) = R\gamma(y^{obs}),$$

With

$$\varphi_\gamma(y) = x_\gamma(y), \omega_{i,\gamma}(y) = e_i^T x_\gamma(y),$$

and the key question is how small the reconstruction error $|\omega_{i,\gamma}(f(x^*)) - x_i^*|$ can be made.

Tikhonov regularization and spectral filtering

Variational (Tikhonov) regularization defines, for $\alpha > 0$, [4] [6]

$$x_\alpha(y^{obs}) \in \arg \min_{x \in X} \{ \|f(x) - y^{obs}\|_Y^2 + \alpha \Omega(x) \},$$

where Ω encodes prior information (e.g. small norm, smoothness, or feasible ranges). The mapping

$$\varphi_\alpha(y^{obs}) = x_\alpha(y^{obs}), \omega_{i,\alpha}(y) = e_i^T x_\alpha(y),$$

is a regularized inverse and associated scalar leakage function. As α increases, reconstructions become more stable but less accurate; as $\alpha \rightarrow 0$, one approaches exact inversion but instability resurfaces. The classical theory characterizes regimes where stable, convergent approximations exist and describes how reconstruction accuracy behaves as noise decreases and α is chosen appropriately.

In the linear case $f(x) = Ax$, a standard choice is

$$\min_x \{ \|Ax - y^{obs}\|_2^2 + \alpha \|Lx\|_2^2 \},$$

with solution

$$x_\alpha = (A^T A + \alpha L^T L)^{-1} A^T y^{obs},$$

so that $\omega_{i,\alpha}(y)$ is an explicit linear functional of y^{obs} . The spectral properties of A determine how much information about each component of x can be stably recovered. In particular, write the singular value decomposition (SVD) of A as

$$A = U \Sigma V^T,$$

with singular values σ_j . Then Tikhonov solutions can be represented as

$$x_\alpha = \sum_j g_\alpha(\sigma_j) (u_j^T y^{obs}) v_j, g_\alpha(\sigma) = \frac{\sigma}{\sigma^2 + \alpha}.$$

Here $g_\alpha(\sigma)$ acts as a spectral filter, damping the contribution of components associated with small singular values (which are most sensitive to noise). For general L , analogous expressions are obtained using a Generalized Singular Value Decomposition (GSVD) of the pair (A, L) . Directions with small σ_j (weakly reflected in outputs) are strongly damped and thus hard to reconstruct; attributes aligned with such directions will

exhibit low leakage even under optimal Tikhonov reconstruction, while attributes aligned with well-observed directions may leak strongly.

Iterative Regularization and Quasi-Reversibility

When f is nonlinear or large-scale, x^* is computed by minimizing the Tikhonov functional iteratively, e.g.

$$x^{(k+1)} = x^k - \lambda_k \Delta J_a(x^k), J_a(x) = \|f(x) - y^{obs}\|^2 + \alpha \Omega(x),$$

or via iteratively regularized Gauss–Newton schemes. Early stopping–terminating the iteration after a finite number of steps–acts as an implicit regularization: small λ yields stable but biased reconstructions; large λ reduces bias but amplifies noise. The mapping

$$\varphi_k(y^{obs}) = x^k(y^{obs}), \omega_{(i,k)}(y) = e_i^T x^k(y),$$

is then a family of leakage functions indexed by iteration count, with leakage increasing as k grows and reconstructions approximate the unregularized solution. Computational limits naturally bound feasible k and thus the accuracy of $\omega_{(i,k)}$.

Quasi-reversibility encodes regularization in an evolution equation, [7-8]

$$\frac{dx}{dt} + B(x, t) = 0, x(0) = x_0,$$

With B chosen so that stationary points approximate minimizers of a regularized functional. For finite time T ,

$$\varphi_T(y^{obs}) = (T; y^{obs})$$

provides another regularized inverse; the “time” parameter plays the role of a regularization parameter, with larger T reducing bias but amplifying sensitivity to noise. Again, in realistic systems, constraints on computation time and access to f (e.g. only through API calls) bound feasible T and hence limit the achievable leakage.

Parameter choice, stability, and leakage implications

Performance of all these schemes depends on the regularization parameter(s): in practice, attackers do not know the true noise level or forward model exactly. Classical parameter choice strategies (discrepancy principle, generalized cross-validation, L-curve, quasi-optimality, and others) aim to select $\alpha(\delta)$, $k(\delta)$ or $T(\delta)$ such that

$$\|f(x_\gamma) - y^{obs}\| \approx \tau \delta, \tau \geq 1,$$

thereby preventing both under- and over-regularization. When applied to leakage, suitable parameter choice determines how close attackers can get to optimal reconstruction for a given noise level and architectural constraint.

Under standard assumptions on f , Ω , and parameter choice, regularization theories provide convergence rates for $x_\gamma = x^*$ as noise decreases and observational richness increases. In the context of leakage, these results imply that if attackers can

repeatedly query the system or observe many outputs, and if their priors are roughly correct, reconstruction accuracy may approach that of an ideal inverse map–unless architectural constraints (such as blind processing or strict noise mechanisms) keep the inverse problem severely ill-posed. Conversely, if the forward map is highly smoothing or observation space is severely restricted, even optimally tuned regularized inverses remain too inaccurate on sensitive coordinates.

Data-driven and Probabilistic Inverse Models

When f is complex, partially unknown, or observable only through input–output samples or black-box queries, explicit analytic inversion may be intractable. Instead, attackers may construct data-driven approximations to the inverse map using machine learning methods. Rather than modeling f explicitly, they learn leakage functions $\omega(y)$ or $\varphi(y)$ directly from observed pairs (x, y) or from synthetic data generated by approximate forward models.

Kernel and RKHS-based leakage functions

Suppose we have M pairs $(y^{(j)}, x_i^{(j)})$ for $j = 1, \dots, M$, where x_i is a sensitive attribute and y are observable outputs. One approach is to learn ω in a Reproducing-Kernel Hilbert Space (RKHS) with kernel k , one solves [9-10]

$$\min_{\omega \in H_k} \frac{1}{M} \sum_{j=1}^M (\omega(y^{(j)}) - x_i^{(j)})^2 + \lambda \|\omega\|_{H_k}^2$$

By the representer theorem,

$$\omega(y) = \sum_{j=1}^M \beta_j k(y, y^{(j)})$$

with coefficients β obtained from a linear system. The learned mapping is a scalar leakage function; a vector leakage φ is obtained by learning several such scalars.

Kernel methods allow flexible nonlinear leakage functions within a capacity-controlled function class. Regularization via λ trades off data fit against complexity, analogous to Tikhonov regularization in function space. Attackers must choose kernels and hyperparameters; defenders may exploit this by designing f so that sensitive attributes only influence outputs through components poorly captured by natural kernels, or require large complexity to approximate.

Model-inversion and gradient-inversion attacks

Model-inversion attacks treat the deployed model as the forward map and reconstruct inputs by solving [11]

$$x^* = \operatorname{argmin}_x \{L(f(x), y^{obs}) + R(x)\},$$

Where L measures output mismatch and regularizes inputs toward plausible records. In classification, y^{obs} may be a target class or score vector; in regression, a continuous output. The quality of inversion depends on the expressiveness of f , the information content of outputs, and the strength of regularization R .

The resulting mappings

$$\varphi(y^{obs}) = x^{\wedge}(y^{obs}), \omega_i(y^{obs}) = x_i^{\wedge}(y^{obs})$$

are implicitly defined leakage functions, computed numerically by optimization for each observed output. When models are trained centrally and gradients are shared (e.g. in federated learning), gradient-inversion attacks reconstruct inputs from shared gradients by solving an analogous optimization problem, replacing $L(f(x), y^{obs})$ with a loss comparing model gradients at x to observed gradients. Both settings highlight how even limited outputs (scores, gradients) can enable high-fidelity reconstructions, especially when models are overparameterized and training data is scarce.

Symbolic regression: interpretable leakage functions

Symbolic regression searches directly for analytic expressions $\omega(y)$ within a predefined grammar (e.g. polynomials, rational functions, and elementary functions) and selects ones with low error and low complexity [12-13].

The resulting $\Phi(y)$ is simultaneously a leakage function and an interpretable analytical model of relationships between outputs and hidden attributes. This interpretability can help attackers understand which outputs drive leakage and how to refine queries, while offering defenders insight into which reports or noise mechanisms to modify. Defenders may choose to prohibit certain simple functional forms from appearing in reports or add targeted noise to destroy the key dependencies identified by symbolic regression.

Invertible neural networks and normalizing flows

Invertible Neural Networks (INNs) and normalizing flows parameterize bijective mappings between latent variables and observed data, with tractable Jacobian determinants. In inverse-problem settings, they are used in two main ways: [14-15]

1.1. Approximate forward model with explicit inverse.

An INN g^{θ} is trained so that $y \approx g^{\theta}(x)$. After training, the inverse

$$x \approx (g^{\theta})^{-1}(y)$$

defines a vector leakage function $\varphi_{\theta}(y)$.

1.2. Conditional flows. Conditional INNs model the conditional distribution $p(x | y)$ via transforms

$$x = g^{\theta}(u, y)$$

with latent u . For fixed y^{obs} , sampling yields samples from an approximate posterior over x , and scalar leakage functions can then be defined as summaries of this distribution, e.g.

$$\omega_i(y) = E[x_i | y]$$

These approaches capture complex nonlinear relations but require substantial training data and computational resources. From a defense perspective, they represent powerful attackers;

designing architectures and noise mechanisms that remain secure even against such sophisticated inversion models is more demanding than for simple linear or kernel-based leakage functions.

Bayesian and variational reconstruction of hidden attributes

Bayesian inverse modeling treats hidden attributes as random variables with prior distributions and asks what can be inferred from outputs. For a sensitive attribute x_i , [16-17]

$$p(x_i | y) \propto p(y | x_i) p(x_i),$$

Where $p(x_i)$ encodes prior knowledge (ranges, sparsity, correlations with other attributes) and $p(y | x_i)$ is the likelihood induced by the forward map and any noise mechanisms. Leakage functions become posterior summaries such as $\omega_{mean}(y) = E[x_i | y]$, $\omega_{med}(y) = median(x_i | y)$ and credible intervals quantify residual uncertainty.

Since exact posteriors are rarely tractable, variational inference, Markov chain Monte Carlo, and related approximation schemes are used. From the attacker's side, Bayesian modeling naturally incorporates side information and structural priors (e.g. sparsity, monotonicity, or known correlations). From the defender's side, Bayesian analysis provides a way to reason about posterior uncertainties given a specified prior and observational model, offering tools for quantifying residual privacy risks under worst-case priors or realistic attacker beliefs. For systems subject to stringent privacy requirements, formal guarantees (e.g. differential privacy) or robust Bayesian bounds may be needed in addition to empirical or per-user risk measures.

Constraints of Blind Data Processing Architectures

The inverse-problem perspective developed above applies both to conventional centralized databases and to blind data-processing SBDs. In the latter case, additional constraints limit what attackers can observe and how accurately they can estimate f or its inverse.

In blind data-processing systems, data from multiple owners is processed through a pipeline where raw records never leave the owners' control. Instead, each owner applies local transformations or cryptographic protections and sends only derived features, encrypted aggregates, or masked contributions to the central system. Reports are computed on the combined processed data, and internal model parameters or intermediate representations may be hidden. The effective observation space is thus a small, structured subset of

$$Y = f(x),$$

and the forward map is often only partially known (architectural details and loss functions may be public, parameters and preprocessing are not) [18-19].

Such architectures also introduce explicit report noise. In some designs, reported values are randomized by adding noise to outputs or by sampling from distributions conditioned on $f(x)$. In others, reports are quantized, truncated, or aggregated over groups of records (e.g. cohort-based statistics). In multi-owner systems, each owner may apply independent noise mechanisms or perturbations based on their local privacy or business requirements.

Conceptually, these mechanisms can be modeled as a noise operator acting on outputs,

$$y^{obs} = N(f(x^*), \xi),$$

Where ξ collects all randomization and perturbations. An attacker sees only y^{obs} and perhaps some public metadata about N (e.g. reported error bars, confidence intervals, or stability summaries). The effective inverse problem is then to reconstruct x (or sensitive components thereof) from noisy, partially aggregated outputs under an uncertain forward model and noise mechanism.

When noise mechanisms depend on ranges or variability of inputs, they can be seen as implicitly defining feasible sets for x . For example, suppose that before reporting $y^{obs} = f(x^*) + \eta$, the system samples from values consistent with input ranges

$$\prod_{j=1}^n X_j, X_j = \begin{cases} [(1-\sigma_j)x_j^*, (1+\sigma_j)x_j^*], & j \in K, \\ \{x_j^*\}, & j \notin K, \end{cases}$$

Here $K \subset \{1, \dots, n\}$ indexes the coordinates subject to multiplicative perturbations, and σ_j are relative perturbation bounds. From the attacker's viewpoint, this translates into admissible sets for reconstructed attributes and constraints on inversion algorithms. The larger the σ_j (for sensitive j), the more ill-posed the reconstruction becomes, and the larger the reconstruction errors can be made by Tikhonov, iterative, or Bayesian methods.

Many system reports are non-smooth functionals: quality metrics, top- k rankings, decision thresholds, or discrete risk categories. Such functionals can make the effective inverse problem even more ill-posed, as small changes in x may not affect reported y at all (flat regions) or may produce abrupt jumps. From an attacker's perspective, this reduces usable information and can force leakage functions to be highly nonlinear or discontinuous. From a defender's perspective, it suggests preferring reports that aggregate information in ways that make reconstruction difficult, while still preserving utility for legitimate monitoring and decision-making.

Defensive Use of Regularization Against Indirect Leakage

Regularization theory can be turned into a design tool for privacy-aware AI systems. From the defender's perspective, one seeks to design f , noise mechanisms, and reporting rules so that for any feasible leakage function in a suitably restricted class

\mathcal{W} (e.g. Lipschitz, bounded RKHS norm, bounded-depth neural networks),

$$\inf_{\omega \in \mathcal{W}} E[\omega(f(x)) - x_i]^2 \geq \varepsilon_i$$

for prescribed privacy thresholds $\varepsilon_i > 0$. Here the expectation is taken over the data-generating distribution and any internal noise in f or the reporting mechanism. This formulation parallels minimax optimal design in inverse problems: the defender designs f and noise to maximize worst-case reconstruction error for sensitive attributes, subject to utility constraints on f 's performance for legitimate tasks. Conversely, an attacker seeks $\omega \in \mathcal{W}$ that minimizes reconstruction error on sensitive coordinates; it recasts defense as a minimax inverse-problem design task.

Classical regularization suggests concrete levers. Increasing smoothing, adding noise targeted at directions most informative about sensitive attributes, or restricting the complexity of reports (e.g. limiting dimensionality or nonlinearity) all increase ill-posedness of the inverse problem faced by attackers. However, these same modifications may also degrade utility for legitimate tasks, leading to trade-offs. From a design standpoint, one aims to adjust f and reporting mechanisms so that sensitive attributes correspond primarily to directions in which f is heavily regularized or noise-dominated, while useful task-related information remains stably recoverable.

In practice, defenders can treat attack algorithms as calibration tools: by simulating attacks (Tikhonov-based, kernel-based, model-inversion, invertible-flow, Bayesian) on synthetic or historical data, they can estimate achievable leakage under different architectural choices and noise levels. This allows explicit tuning of regularization strength, noise parameters, and reporting granularity to meet target privacy thresholds while maintaining acceptable utility.

These ideas point to actionable research directions: (i) developing theory that links architectural constraints and noise mechanisms to lower bounds on worst-case leakage by classes of attackers; (ii) designing automated tools that, given a target system and privacy requirements, propose concrete modifications (in added noise, reporting rules, or model architecture) that provably or empirically limit indirect leakage; and (iii) integrating such tools into the design and certification processes for AI systems operating on sensitive big data.

Conclusion

We have described indirect information leakage in AI systems as an inverse problem: given outputs of a complex pipeline, to what extent can hidden attributes of input records be reconstructed? By formalizing leakage functions as approximate inverses of the AI pipeline, we can analyze how architectural choices, regularization mechanisms, and noise affect reconstruction accuracy. Classical

regularization methods, data-driven inverse models, and Bayesian approaches provide powerful tools for attackers; but the same concepts, when used deliberately by system designers, offer a language for defending against such leakage.

Blind data-processing architectures and defensive regularization strategies together suggest a path toward AI systems that offer high utility while limiting reconstructive power of adversaries. By treating privacy-preserving system design as a controlled ill-posed inverse problem, one can seek architectures in which, under realistic attacker models and noise levels, even the best reconstruction algorithms fail to recover sensitive attributes with unacceptable accuracy. This perspective invites further collaboration between inverse-problems theory, information security, and machine learning in developing principled frameworks for understanding and controlling indirect leakage in future large-scale data-driven systems.

References

1. Dwork C (2006) Differential privacy, in International colloquium on automata, languages and programming (ICALP).
2. Wood A, Altman M, Bembenek A, Bun M, Gaboardi M, et al. (2020) Differential privacy: A primer for a NonTechnical audience. Vanderbilt Journal of Entertainment and Technology Law 21(1): 209-276.
3. Dinur I, Nissim K (2003) Revealing information while preserving privacy in Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems (PODS) New York: ACM.
4. Tikhonov AN, Arsenin VY (1979) Methods for Solving Ill-Posed Problems (in Russian). Moscow: Nauka.
5. Hansen PC (2010) Discrete inverse problems: Insight and algorithms. Philadelphia: SIAM.
6. Tikhonov AN (1965) On ill-posed problems of linear algebra and their stable solution" (in Russian), Doklady AN SSSR 163(3): 591-594.
7. Landweber L (1951) An iteration formula for fredholm integral equations of the first kind," American Journal of Mathematics 73(3): 615-624.
8. Lattès R, Lions J (1969) The method of quasi-reversibility: Applications to partial differential equations. New York: Elsevier.
9. Aronszajn N (1950) Theory of reproducing kernels. Transactions of the American Mathematical Society 68(3): 337-404.
10. Schölkopf B, Smola A (2002) Learning with kernels: Support vector machines, regularization, optimization, and beyond. Cambridge: MIT Press.
11. SaturnCloud (2024) Model inversion attacks.
12. Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. Science.
13. Jin Y, Fu W, Kang J, Guo J (2019) Bayesian symbolic regression. arXiv.
14. Ardizzone L et al. (2018) Invertible neural networks for inverse problems. arXiv.
15. Prs G (2020) Normalizing flows.
16. Kaipio J, Somersalo E (2005) Statistical and computational inverse problems. New York: Springer.
17. Tran VH (2018) Copula variational bayes inference via information geometry. arXiv.
18. Konyavskiy VA, Konyavskaya SV (2019) Trusted Information Technologies: From Architectures to Systems and Tools (in Russian). Moscow: URSS pp. 264.
19. Konyavskiy et al. VA (2024) 'Technology of 'blind' processing of attract-ed data in machine learning systems" (in Russian). Voprosy Zashchity Informatsii 2: 17-32.



This work is licensed under Creative Commons Attribution 4.0 License
DOI: [10.19080/RAEJ.2025.07.555701](https://doi.org/10.19080/RAEJ.2025.07.555701)

Your next submission with Juniper Publishers will reach you the below assets

- Quality Editorial service
- Swift Peer Review
- Reprints availability
- E-prints Service
- Manuscript Podcast for convenient understanding
- Global attainment for your research
- Manuscript accessibility in different formats
(Pdf, E-pub, Full Text, Audio)
- Unceasing customer service

Track the below URL for one-step submission

<https://juniperpublishers.com/online-submission.php>